Doctoral Thesis

Enabling Secure and Wireless Battery Management Systems

bak. elektr. Dipl.-Ing. Fikret Bašić

DOCTORAL THESIS

to achieve the university degree of
Doktor der technischen Wissenschaften

submitted to
Graz University of Technology
and conducted at the
Institute of Technical Informatics

Supervisor:
Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Eugen Brenner

Advisor:
Ass.-Prof. Dipl.-Ing. Dr.techn. Christian Steger

Graz, August 2023

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

Date

Signature

Acknowledgements

Ever since I can remember, I have always been curious and eager to explore new things. Eventually, this passion led me to pursue a dissertation. I spent many months reading, brainstorming, analyzing, researching, implementing, paper writing, teaching, and project leading, always having to give a little extra to push through to the next step. This hard road is finally at an end, but it would not have been possible without the support of my parents, family, friends, and colleagues. I am grateful to have met so many inspiring individuals who played a significant role in my journey. If I could name every single person, I would, but sadly that would require a thesis of its own.

First and foremost, I want to express my gratitude to my parents, Nafija and Ferid Bašić, who always supported me, stayed with me through difficult periods, and helped me to be here where I am today. Thank you very much from the bottom of my heart. - Draga mama, dragi tata, hvala vam iskreno na svoj vašoj podršci i što ste mi omogućili da stignem tamo gdje sam trenutno.

I want to thank the whole staff of the Institute of Technical Informatics here at Graz University of Technology. I especially would like to express my gratitude towards my advisor and mentor, Ass.-Prof. Christian Steger, for his unwavering support throughout our years of collaboration. Our journey began in the summer of 2017 when I first joined as a student assistant, but his guidance has been especially invaluable now during my PhD career. I really appreciate all your support and advice over these many years. Furthermore, I want to thank my supervisor Ao.Univ.-Prof. Eugen Brenner for accepting me as one of his students and giving me this opportunity. I also want to thank my external reviewer Assoc.-Prof. Andreas Riel for honoring me in accepting to be on my defense commission and for reviewing my dissertation. Additionally, I want to thank Prof. Kay Römer, Assoc.-Prof. Carlo Alberto Boano, and Dr. Georg Macher for all your research advices and support.

I cannot forget to give a big shoutout and “thank you” to my dear colleagues at the Institute, who went together with me through thick and thin. They are the ones that made this journey all the more enjoyable. Each one of you deserves your own spotlight. I was very lucky to share the most unique and culture-filled, not to mention the best, office at the Institute. Here, a big thanks to Dr. Michael Spörk who welcomed me with open arms, Dr. Alexander Rech for all the opportunities, Dr. Martin Erb, for lending an ear, Dr. Lukas Gressl, for giving me new insights, Theo Gasteiger, for your ‘just do it’ attitude, and Rainer Hofmann, for being always a good friend. Thank you also to older and younger members of the office, Felix Warmer, Dong Wang, Christian Seifert, and Christof Schützenhöfer. I know that the ‘Party Büro’ will stay in good hands.

I also want to thank Philip Stelzer and Benjamin Rohr, my first Institute colleagues, Markus Feldbacher, for always lending a hand in my newcomer Austrian challenges, Reinhard Enhuber, the most daring bicyclist, Dr. Thomas Ulz and Dr. Thomas Pieber, my first advisers. My sincere thanks also to Dr. Michael Krisper for his teaching insights and for providing me with his template. A very warm thank you also to Elisabeth Salomon, for your thoughtfulness, Hannah Brunner, for your determination, Romana Blažević, for being always delightful, Maximillian Schuh, for being a great pal and fast Bosnian learner, Markus Gallacher, for your positive outlook, Michael Stocker, the real multi-tasker, Dr. Markus Schuß, the technical sage with the best comeback jokes, Gernot Fiala, and all your amazing stories, Francesco Corti, for the style points, and Mohamed Hassaan M. Hydher, for being a courteous researcher. Thank you also to all the well-rounded academics from “down the hall”, Dr. Tobias Scheipel, Jürgen Dobaj, Meinhard Kissich, Kristof Kanics, Leandro Ribeiro, and Sonali Deo. I will always cherish our nights, hangouts, and even the short coffee chit-chats. Thank you all for being there.

I want to thank the whole NXP collective for their expertise and project collaboration, but especially Robert Kofler and Klemens Breitfuß for all your support over these years. A big thank you also to my students and colleagues who accompanied me throughout the projects, Martin Gärtner, Claudia Laube, Christian Seifert, Maida Mačak, Patrick Eder, Andreas Finsterbusch, and Valentin Bürgel. You all did an amazing job and I am very proud of your achievements.

I am grateful to all the teachers, professors, and school colleagues who have shared their own journeys with me, beginning at “Prva osnovna škola” and “Gimnazija” Velika Kladuša, to Bihać Faculty of Technology. I will always have fond memories of those times. This would not have been possible without you.

Finally, I want to take a moment to thank my family, friends, and colleagues back in my home country, who were always interested in and followed my PhD journey with great enthusiasm. I especially want to express my appreciation to the families Muminović, Muratović, Murić, Miljković, Anel Pajazetović and his parents, Amir Alispahić and his family, aunt Safija, and my cousins Benjamin and Belmin, as well as my entire extended Bašić family. To all my neighbours, but especially my first neighbour, Faris Muminović, for all the insightful coffee discussions, Hasib Torić, for showing me how to be persistent, Dinko Čejvanović, for all our long and eventful talks, and Klara Kristić, for all the Graz adventures. To my old ‘Fantastic Seven’ friends who have been an important part of my childhood, Azur, Mirza, Selver, Maki, Haro, and especially Almir Latić, for being who you are. I will also never forget all the colleagues that helped me the first time I came to Graz and with many whom I stayed in contact with up until today, Emina, Aid, Ajdin, Emir, Hanka, Adis, Miljana, Stefan, Philip, Christoph, Sara, Esmeralda, Adem, Kemal, Nenad, Siniša, Miloš and Aleksandar, Abdulmeki, and Samir.

Thank you all for supporting me and making my PhD journey more interesting!
Hvala vam svima na podršci i što ste učinili moj doktorski put mnogo zanimljivijim!

Graz, August 2023
Fikret Bašić

Abstract

The increase in environmental awareness has led to a growing interest in the new developments in the field of renewable energy, both from academia and industry. Today, electric vehicles play a significant role. One of the most important components in modern electric vehicles are the batteries needed to power them and the battery management systems (BMS) responsible for their control and safety. One challenge with modern BMS is the use of the traditional wired design for diagnostic data transfer. As the number of battery cells increases, so does the number of cables, which increases production costs and maintenance complexity. To address this problem, several wireless BMS designs have been proposed with varying degrees of success. However, little research has been conducted that considers both internal sensor communication and external BMS communication, as using multiple wireless technologies for different use cases can lead to unintended interference issues and complexity.

In light of the European Union's new initiatives on battery passports, which were scheduled to take effect as early as 2023, new solutions should be proposed that allow for flexible design of BMS data readout while being cost-effective, expandable, and secure. Security is particularly important in modern BMS, as tampering at any of the data transmission layers could compromise the functional operation of the BMS and the privacy of its users. However, applied BMS security is still a largely unexplored field. The current state of the art does not provide a clear answer with many challenges related to the design of BMS security still remaining open.

In this dissertation, I explore and propose two novel concepts:

- A design based on the use of RFID technology, in particular near-field communication (NFC), for internal and external BMS communication.
- A lightweight security architecture based on novel implicit certificates that takes into account all communication interfaces of the BMS distribution and enables secure data transmission from sensor sources through internal system networks, and finally, to end users.

The dissertation provides security design extensions at three architectural levels. First, at the intramodule BMS level with the proposed secure BMS data block design and device authentication for battery packs via NFC sensor communication. Second, for the internal local network, e.g., in-vehicle, with a security architecture for authentication and certificate derivation. Third, for cloud connectivity and lifecycle tracking with a system design proposal based on a centralized gateway and secure encoding of BMS data blocks. In addition, an important focus is placed on deriving a dynamic secure session design. To this end, a novel design is proposed based on the use of a station-to-station protocol with implicit certificates that provides perfect forward secrecy and can be extended to other vehicle controllers or similar embedded environments. To test the applicability of the proposed design and research ideas, a test suite was designed, implemented, and evaluated. The evaluation results showed the feasibility of the proposed solutions with the BMS architecture considering the performance and security requirements when used in real applications. The proposed design solutions are fully compatible with various derivatives of modern BMS topologies and can be easily applied and extended.

Kurzfassung

Das gestiegene Umweltbewusstsein hat zu einem wachsenden Interesse an neuen Entwicklungen im Bereich erneuerbarer Energien sowohl in der Wissenschaft als auch in der Industrie geführt. Heutzutage spielen Elektrofahrzeuge eine bedeutende Rolle. Eine der wichtigsten Komponenten moderner Elektrofahrzeuge sind die zu ihrem Antrieb benötigten Batterien und die "Battery Management Systems" (BMS), die für deren Steuerung und Sicherheit zuständig sind. Eine Herausforderung bei modernen BMS ist die Verwendung des traditionellen kabelgebundenen Designs. Mit zunehmender Anzahl der Batteriezellen steigt auch die Anzahl der Kabel, wodurch sich die Produktionskosten und der Wartungsaufwand erhöhen. Um dieses Problem anzugehen, wurden verschiedene Designs für drahtlose BMS mit unterschiedlichem Erfolg vorgeschlagen. Allerdings gibt es nur wenige Untersuchungen, die sowohl die interne Sensorkommunikation als auch die externe BMS-Kommunikation berücksichtigen, da die Verwendung mehrerer drahtloser Technologien für verschiedene Anwendungsfälle zu unbeabsichtigten Interferenzproblemen und Komplexität führen kann.

Angesichts der neuen Initiativen der Europäischen Union zu Batteriepässen, bzw., "Battery passports", die bereits 2023 in Kraft treten sollten, sollten neue Lösungen vorgeschlagen werden, die eine flexible Gestaltung der BMS-Datenauslesung ermöglichen und gleichzeitig kostengünstig, erweiterbar und sicher sind. Sicherheit ist in modernen BMS besonders wichtig, da Manipulationen an einer der Datenübertragungsebenen den funktionalen Betrieb des BMS und die Privatsphäre seiner Benutzer gefährden könnten. Allerdings ist die angewandte BMS-Sicherheit noch ein weitgehend unerforschtes Gebiet. Der aktuelle Stand der Technik liefert keine eindeutige Antwort, da viele Herausforderungen im Zusammenhang mit der Gestaltung der BMS-Sicherheit noch offen sind.

In dieser Dissertation untersuche und schlage ich zwei neuartige Konzepte vor:

- Ein Design, das auf der Verwendung von RFID-Technologie, bzw., "Near-field Communication" (NFC), für die interne und externe BMS-Kommunikation basiert.
- Eine optimierte Sicherheitsarchitektur. Dies basiert auf neuartigen impliziten Zertifikaten, die alle Kommunikationsschnittstellen der BMS-Distribution berücksichtigen und eine sichere Datenübertragung von Sensorquellen über interne Systemnetzwerke ermöglichen.

Diese Dissertation bietet Erweiterungen des Sicherheitsdesigns auf drei Architekturebenen. Erstens auf der modulinternen BMS-Ebene mit dem vorgeschlagenen sicheren BMS-Datenblockdesign und der Geräteauthentifizierung für Batteriepacks über NFC-Sensorkommunikation. Zweitens für das interne lokale Netzwerk, z. B. im Fahrzeug, mit einer Sicherheitsarchitektur zur Authentifizierung und Zertifikatsableitung. Drittens für Cloud-Konnektivität und Lebenszyklusverfolgung mit einem Systemdesignvorschlag, der auf einem zentralen Gateway und sicherer Verschlüsselung von BMS-Datenblöcken basiert. Darüber hinaus wird ein wichtiger Schwerpunkt auf die Ableitung eines dynamischen sicheren Session-Designs gelegt. Zu diesem Zweck wird ein neuartiges Design vorgeschlagen, das auf der Verwendung eines Station-to-Station-Protokolls mit impliziten Zertifikaten basiert, das "perfect forward secrecy" bietet und auf andere Steuergeräte oder ähnliche eingebettete Umgebungen erweitert werden kann. Um die Anwendbarkeit der vorgeschlagenen Design- und Forschungsideen zu testen, wurde eine

Testsuite entworfen, implementiert und evaluiert. Die Evaluierungsergebnisse zeigten die Machbarkeit der vorgeschlagenen Lösungen mit der BMS-Architektur unter Berücksichtigung der Leistungs- und Sicherheitsanforderungen beim Einsatz in realen Anwendungen. Die vorgeschlagenen Designlösungen sind vollständig kompatibel mit verschiedenen Derivaten moderner BMS-Topologien und können einfach angewendet und erweitert werden.

Extended Abstract

Research into battery management systems (BMS) has gained popularity in recent years. This is due to the necessity to improve current systems and better respond to the modern requirements found with various electrical systems, especially electric vehicles, and smart power grids. Both industry and academia are focusing on developing new solutions to support emerging research questions. This is important, as BMS fill the role of essential control devices for modern electrical systems, supporting the concept of renewable energy. As the number and complexity of electrical systems used increases, so does the number of battery cells required. In fact, we observe an exponential increase in the production of lithium-ion battery cells today, with the numbers expected to continue to grow over the next decade. A large number of battery cells generates more waste, a problem that is not easily solved and certainly not by simply relying on recycling. Battery cell recycling is both complex and costly, but more importantly, it has a negative environmental impact. An effort is made to allow the maximum utilization of battery cells before their recycling phase. In support of this notion, the European Union has announced the introduction of battery passports. These battery passports would keep track of the necessary battery data to support the concept of "*battery second life*". However, even with the current specifications, many questions are still left unanswered. These relate in particular to data management, the new system design, and - most importantly for us - system security.

Tracking of the BMS and battery packs is aimed to be done on each battery cell deployment change, i.e., whenever there is a change in the system use case. Battery packs are expected to be stored in warehouses where a reliable, fast, and secure mechanism is expected to be used to enable battery cell status readout and verification. With respect to this challenge, modern BMS are moving toward wireless rather than wired communications, on which numerous publications have appeared in the last decade. Replacing wireless with wired communication for BMS reduces their weight, cost, maintenance, and complexity. However, it is not possible to simply rely on the current state-of-the-art (SotA) solutions because most research on BMS focuses on intra-module communication, i.e., communication between the main BMS controller and the operating BMS controllers, leaving open the design issues related to the intra-sensor and external diagnostic readouts. Another issue that arises with wireless communication is the expansion of potential attack vectors as the attack surface also expands. Current SotA in this area is relatively limited. The focus of the research is kept on BMS security from a theoretical and rather abstract perspective, ignoring functional, protocol, and system design aspects.

A BMS communicate their data not only to its internal network but also to an external network. In this context, a BMS can send its data through an in-vehicle network, to the main gateway controller, and finally to the end system device through an OEM that relies on the cloud or similar infrastructure. To support the notion of battery passports and modern functional use of BMS data on external platforms, an appropriate security architecture must be considered to enable efficient and secure data transmission from the source of the battery cell sensors, through the modulated BMS components and the main BMS controller, to the central gateway unit, and finally to the end-user system. Currently, no such unified design exists, with designers and implementers having to rely on solutions found in similar automotive environments, but not necessarily proven for BMS requirements.

Research questions. After evaluating the challenges and requirements of modern BMS system design in relation to security and wireless connectivity, we identified the following research questions that were the main focus of this dissertation's investigation:

1. How to implement a unified and secure wireless battery management system design for the internal sensor and external diagnostic communication?
2. How to design a lightweight security architecture for battery management systems for local network communication?
3. How to realize an efficient and secure design for battery management system data acquisition and propagation to external end systems and services?

Contributions. Our research aimed to create a BMS security system that is lightweight, flexible, and independent of BMS topologies at every layer of data propagation. We developed three research questions to guide us: the first concerning internal communication of the BMS sub-system, the second focused on the BMS local area network, and the third on BMS data processing and remote transmission. Our goal was to ensure security across all these layers. Each layer considers the necessary security solutions based on the derived security requirements from the analysis of the current BMS SotA, vehicle, embedded systems, and other security-related publications and sources. In the end, we aimed to provide an answer for the current and future BMS in terms of complete data propagation from the cell source to the external and remote end system. These solutions must also meet the requirements imposed by the modern battery passport and second-life initiatives.

BMS secure wireless readout. The first part of the research focused on the use of wireless communication technology for internal sensors and external status readouts considering device and data security. Our novel contribution explores the use of near-field communication (NFC) as a wireless technology for the use with BMS, being the first such solution in the field. The solution for the system design is split into two categories: internal and external readouts, with three use cases. The internal readout concentrates on the battery pack sensor, which entails reading battery cell sensor data using the intermediary battery pack controller (BPC). The external readout concentrates on active and idle diagnostics readout, which focuses on using a mobile device for second-life applications. The system design has been investigated with solutions proposed on the matter of devices in use, communication interfaces, wake-up procedure optimization, and data structuring. In addition to these solutions, we have also taken into account the security aspect of the system. Our primary concern is to authenticate the battery packs to prevent any malicious or counterfeit devices from being used. To ensure external status readout security, we have designed a lightweight security protocol based entirely on symmetric cryptography. We have also proposed a dedicated secure data structure called SNDEF, which extends on the NDEF layer structure. To validate the effectiveness of our proposed solutions, we have implemented and evaluated them experimentally to show their potential use with real BMS deployments.

BMS secure network design. Modern BMS transmit important safety-related information through their internal network to various devices, including the communication gateway, display unit, main engine unit, and electric vehicle charging controller (EVCC). It is important to ensure that both the devices and data on the network are authenticated and their data confidentiality and integrity are maintained. We are the first to develop and implement a complete lightweight security architecture at the protocol and system design level for BMS local area network communication. We do this by employing implicit certificates, specifically the Elliptic Curve Qu-Vanstone (ECQV) scheme. The design is based on centralized network control with a dedicated device gateway responsible for initial device authentication and certificate derivation and exchange. To this end, we also develop and propose a device authentication protocol based on symmetric cryptography, while implicit certificates are used for subsequent communication authentication between BMS and other network devices, i.e., electronic control units

(ECU). To complement a secure session communication between BMS and other devices, we realize a lightweight key derivation protocol based on the traditional Station-to-Station (STS) protocol considering the ECQV scheme. By using this key derivation protocol, we are able to achieve dynamic properties, specifically the perfect forward secrecy. The aforementioned protocols have been designed under a separate layer and data structure, to allow easier integration into the Controller Area Network (CAN), but also other communication technologies. The protocols were analyzed within the implemented test suite by implementing them directly on the communication lines, i.e., relying on the serial and CAN protocols. The proposed STS protocol was also optimized and tested in comparison to other SotA protocols for session key derivations using implicit certificates. It has also been tested on different embedded devices based on their performance to account for different variants of BMS controllers.

BMS secure design for external systems. BMS generate a large amount of data that can be used as input to other services and processes, such as machine learning predictions, vehicle network profiles, etc. The data is expected to be also used as an input for the dynamic entries of “battery passports”. However, at the moment, there is an open question regarding the necessary security coverage for this process as no standards exist. In order to address this issue, we have developed a two-fold solution. Firstly, we have created a secure BMS data block structure based on the chained hierarchical principle which can be used for data logging processes regardless of the targeted BMS topology architecture. Secondly, we have extended the BMS security architecture to include on-premise and cloud data management. This is the first contribution in the field for a secure BMS architecture that takes into account external services. Our design model is a hybrid BMS divided into layers from the BMS data access point of view, with each layer serving as an individual security platform and point of protection. In addition, we discovered and applied two system architectural design patterns, Embedded Platform to Memory (EP2M) & Secure Embedded Logging (SEL), for a secure and efficient logging design.

Outcome. To complete the research and present the usability of the proposed design and architectural solutions from both system design and security perspectives, we have designed and implemented a complete test suite system consisting of real BMS hardware components. We implemented each layer separately and observed its behavior under interacting, as well as independent process segments. The test suite was used for experimental evaluation and verification of the proposed design solutions, with additional separate analyses conducted on the proposed security protocols and methods. We conclude that our presented BMS secure and wireless system architecture answers the posed research questions and meets the necessary design requirements when deployed in real-world systems. The research output and contributions are summarized in Table A.

Table A: Summarization of the dissertation's contributed solutions to BMS security and system design.

	Targeted system elements	**System environment layer**	**Proposed security or system design solution**	**Part of Publication**
1.	*BMS & BPC interfaces*	BMS sub-system	BMS wireless system design utilizing NFC for internal and external readouts	B, C, D, G
2.	*BMS & BPC NFC link*	BMS sub-system	Two NFC-based wake-up methods optimized for BMS use cases	G
3.	*BPC & battery cell pack*	(NFC internal set) BMS sub-system	Efficient two-factor battery pack authentication method	B, C, G
4.	*BMS, BPC & ext. devices*	(NFC external set) BMS sub-system	SNDEF - Secure data layer extended on the NDEF format	D
5.	*BMS, BPC & ext. devices*	(NFC external set) BMS sub-system	Lightweight authentication and secure session derivation protocol for NFC-based BMS external readout	D, G
6.	*Int. channel & devices*	Internal local area network	Security architecture for BMS authentication and certificate derivation based on ECQV implicit certificate scheme	E
7.	*BMS & local devices*	Internal local area network	Secure static key derivation protocol for BMS that rely on the ECQV scheme	E
8.	*BMS & local devices*	Internal local area network	Secure STS-ECQV dynamic key derivation protocol offering perfect forward secrecy	F
9.	*Int. channel*	Internal local area network	Efficient secure data formats for serial network communication protocols, e.g., CAN	F, H
10.	*BMS & log components*	BMS sub-system	Two design patterns for efficient and secure on-premise data logging applicable to BMS	A
11.	*BMS, c. gateway, cloud*	All layers	Secure BMS block data structure based on a chain hierarchical principle	H
12.	*BMS, c. gateway, cloud*	Cloud & end system backend	Layered design concept for secure data propagation from BMS to external cloud systems	H

List of Figures

List of Tables

List of Abbreviations

AEAD	Authenticated encryption with associated data
AES	Advanced Encryption Standard
API	application programming interface
ASIC	application-specific integrated circuit
ASK	amplitude-shift keying
AWS	Amazon Web Service
BAN	Burrows–Abadi–Needham
BCC	Battery Cell Controller
BESS	battery energy storage systems
BLE	Bluetooth Low Energy
BMS	Battery Management System
BPC	Battery Pack Controller
CA	Certificate Authority
CAN	Controller Area Network
CAN-FD	Controller Area Network Flexible Data-Rate
CC	Common Criteria
CCM	Counter with CBC-MAC Mode
CCU	Cell Control Unit
CMAC	Cryptographic-MAC
CoAP	Constrained Application Protocol
CRC	Cyclic redundancy check
CSEc	Cryptographic Service Engine compressed
CVSS	Common Vulnerability Scoring System

DFD	data flow diagram
DKD	dynamic key derivation
DoD	depth of discharge
DoS	denial-of-service
DTLS	Datagram Transport Layer Security
EAL	Evaluation Assurance Level
EC	Elliptic Curve
ECC	Elliptic Curve Computation
ECDH	Elliptic Curve Diffie-Helmann
ECDSA	Elliptic Curve Digital Signature Algorithm
ECQV	Elliptic Curve Qu-Vanstone
ECU	Electronic Control Unit
EDR	event data recorder
EDW	event detection wake-up
EEPROM	electrically erasable programmable read-only memory
EHW	energy harvesting wake-up
EP2M	Embedded Platform to Memory
EU	European Union
EV	electric vehicle
EVCC	eletric vehicle charging controller
FSM	finite state machine
GCM	Galois/Counter Mode
GPIO	general-purpose input/output
GSN	goal structuring notation
HAL	hardware abstraction layer
HMAC	Hash-MAC
HSM	Hardware Security Module
HTTPS	HyperText Transfer Protocol Secure
I2C	Inter-Integrated Circuit

IFAL	Issue First Activate Later
IoT	Internet of Things
ISO	International Organization for Standardization
IV	Initialization vector
KDF	key derivation function
LAN	local area network
LIB	Lithium-ion batteries
LPUART	low power UART
MAC	Message Authentication Code
MCU	microcontroller unit
MitM	Man-in-the-Middle
MQTT	Message Queuing Telemetry Transport
NDEF	NFC Data Exchange Format
NFC	near-field communication
NIST	National Institute of Standards and Technology
OBD	on-board diagnostic
OEM	original equipment manufacturer
OMAC	One-Key CBC MAC
OTP	one-time password
PCB	printed circuit board
PKI	Public Key Infrastructure
PP	Protection Profile
PUF	physical unclonable function
QR	Quick Response
RF	radio frequency
RFID	radio-frequency identification
RNG	random number generator
RSA	Rivest–Shamir–Adleman
RTD	Record Type Definition

SAE	Society of Automotive Engineers
SE	Secure Element
SEL	Secure Embedded Logging
SHE	Secure Hardware Extension
SKD	static key derivation
SNDEF	Secure NFC Data Exchange Format
SoC	state of charge
SoH	state of health
SotA	State-of-the-Art
SPI	Serial Peripheral Interface
SRAM	static random access memory
ST	Security Target
STS	Station-to-Station
TLS	Transport Layer Security
ToE	Target of Evaluation
TPL	transformer physical layer
TPM	Trusted Platform Module
TSCH	Time Slotted Channel Hopping
UART	universal asynchronous receiver-transmitter
UID	unique identifier
V2I	vehicle-to-infrastructure
V2V	vehicle-to-vehicle
VLPS	very low power state

Contents

CHAPTER 1

Introduction

> *"We can only see a short distance ahead, but we can see plenty there that needs to be done."*
>
> \- Alan Turing

Summary: *As every journey has a beginning, I will start this scientific journey with an introduction. In the first part, I will motivate and explain why security and wireless connectivity are such important concepts with modern Battery Management System (BMS) today. After introducing and motivating the topic, I will dive deeper into the problem statement and challenges and also outline the solutions and contributions proposed in this dissertation. And finally, at the end of this chapter, I will give a brief overview of the content of the remaining chapters.*

◇◇◇

1.1 Motivation

In recent decades, the detrimental changes in the environment, climate, and our own lives have given rise to the idea of energy sustainability. Awareness of energy consumption has sparked community interest in becoming more involved with the concept of energy sustainability by replacing traditional fossil fuels with renewable energy sources. An old concept of electric vehicle (EV) has been revived. The reduction in the use of products that rely on fossil fuels has led to an increase in the production and sales of EV [1]. This has led to a further increase in the battery market, with the side effect of increasing the problem of regulating this huge amount of batteries at the end of their life [2]. Compared to previous years, there is currently a rapid growth in battery consumption, with the numbers expected to continue to grow exponentially in the coming years as well [3, 4, 5]. Based on the BloombergNEF's seventh annual long-term Electric Vehicle Outlook (EVO) analysis, which was the latest at the time of writing, electric vehicle market share in Europe and China is expected to rise to 39 % of total vehicle share by 2025 [6]. Battery sales alone accounted for 94 % increase in 2021 compared to 2020. Recycling a large number of batteries could be both energy inefficient and hazardous to the environment, generating a large amount of waste [7, 8]. An alternative approach to recycling would be to use batteries for other, less functional and safety-demanding applications [9, 10, 11]. This process is often referred to as giving the "*second life*" to batteries [12]. In the modern world, rechargeable batteries are not only used with EV but also in other applications such as power tools, electric buses, scooters and bicycles, industrial machinery and electronics, future smart power grid systems such as backup power for data centers, telecommunication base stations and postal services [3, 9, 13]. However, one problem that persists to this day is that regulations vary across countries and even across original equipment manufacturer (OEM). Many automakers have not yet embraced the idea of the second-life use of batteries. The main argument comes from the fact that the average useful life of batteries in EV is expected to be around 10 years, at which point the batteries might not be any more market competitive or usable due to the technological differences [3]. Nevertheless, many initiatives have already been made across both the European Union (EU) and Asia to support this idea of the second use of batteries, with the proposal to use them alongside recycling based on the present battery health [9, 14, 15, 16].

To address the challenges of the current battery market, alongside the battery second life push, the EU has proposed an initiative of creating "*battery passports*" [15, 16]. A battery passport is a digital representation of a battery that stores data from the battery's birth, i.e., raw material extraction and manufacturing, through installation and secondary use, to recycling. While the initiative was originally aimed only at supporting the ethical mining of key battery resources, such as lithium ores, it has since been expanded to also include the verification of batteries for counterfeit variants and the on-premise or online digital storage of battery lifecycle data to support the concept of a battery's second life [17, 2]. In the coming years, we should witness the adoption and use of battery passports, but there are still many challenges that need to be addressed, especially related to battery data sharing and management.

The role of BMS. Battery packs are, in themselves, very simple devices with limited ability to operate independently and communicate with the outside world. Their main purpose is to serve as a power source and provide sensor data. Based on this sensor data, they are regulated and controlled by a central device called BMS [18, 19]. This device is pivotal in our discussion because it provides a central connection point between the outside world and the internal environment of the battery cells. The use of BMS is becoming increasingly important as they act as a bridge between the battery cells and the end users. These users, either OEM or private EV users, can benefit from the data obtained to support the concept of efficient battery use and second life. A BMS would sample the data from the battery pack and afterwards share it with outside systems, but the current models lack the proper design to execute this activity properly [20]. The reliance on traditional wired connectivity between BMS modules limits the possible use cases for the internal and external sensor readouts, with new solutions being necessary [21, 22]. Many industries today, hence, try to switch to wireless use with BMS as an answer to the ever-growing market demands [23, 24]. A similar interest can also be seen regarding the general BMS research field. As an attractive modern topic, there has been a rapid growth of publications in the BMS research field in the context of EV, with a linear increase of total published papers observed over the last ten years [25]. However, there has been a significant lack of papers concerning the security of BMS. One of the reasons is that the majority of published papers focus on conventional BMS concepts concerning chemical, mechanical, and electrical features, rather than concepts dealing with digitalization and connectivity expansion [25, 26, 27]. Furthermore, the expansion of services and connections coming from in and out the BMS is directly related to an increase of potential vulnerabilities and threats [28, 29, 30]. Similar to the initial security observations regarding smart power grids [31, 32], the security with BMS is often neglected due to its perceived complexity and cost requirements. Nevertheless, threats exist, and they will certainly become more prominent with the future realization of the aforementioned battery pack and BMS use cases. For one, a potential attacker would try to pry for vulnerable access to the system for the purpose of capturing a module, manipulating the system data, or compromising the user's privacy. The motivational aspect of these attacks could range from industrial espionage, extortion, or simply vandalism. Thus, for a modern BMS design, it is necessary to provide the authenticity of each internal and external module of communication and guarantee a safe and secure transfer of critical system and sensor data from their source to the end systems.

1.2 Context and Domain

This dissertation was carried out at the Institute of Technical Informatics at Graz University of Technology, in collaboration with NXP Semiconductors Austria GmbH Co & KG as part of the project FFG SEAMAL BMS [1] for the first period between August 2020 and October 2022, and EU KDT OPEVA [2] from January 2023. NXP Semiconductors is one of the leading firms in developing solutions for modern vehicle BMS. They have supported the project by providing industry insights, real-world use cases, and state-of-the-art BMS devices and modules for the purpose of emulating and testing the developed solutions. The publications and scientific work within this dissertation have advanced the idea of secure and wireless BMS system design and influenced further investigations, research, and product investments.

1.3 Problem Statement

Traditionally, BMS have been simplistic controllers that dealt mainly with analog data and local safety and functional control of battery cells, which involved charging, discharging, cell balancing, monitoring sensor thresholds, and diagnostics [33]. Over time, more functions were added, such as constant cell monitoring, thermal management, and battery parameters estimations such as state of charge (SoC) and state of health (SoH) [34, 26, 35, 36, 33]. The inclusion of many modern functions has opened the door to several new challenges that have not yet been considered when looking at current BMS model derivations. BMS are limited in computational power and memory space, as well as extension capabilities [20]. Therefore, it is of utmost importance to carefully disseminate the relevant challenges when dealing with new BMS functions and applications.

In relation to modern BMS challenges, we are interested in the security aspects of BMS. The security of BMS is still a largely unexplored field. Despite several research papers focusing on surveys and theoretical analysis of security threats and defense requirements, no work has yet been published that considers the security architecture for smart and extended BMS in terms of their system design perspective and real-world implementation use cases. As BMS become more complex, they also interact with other systems and become a part of larger networks. In this context, we can no longer view BMS simply as an individual system, but rather as part of a larger technological ecosystem. We observe this concept through the propagation of data entering and leaving a BMS. At a source point, the BMS contains several modules that are enclosed and propagate its data from the battery pack sensors to the external world. At the next level, we find the internal network, e.g., a vehicle network, and finally at the top layer we can find remote end systems and services.

In the following three sub-sections, we focus on explaining different challenges and established research goals by examining each layer of secure BMS data propagation, while also deriving the research questions. The summary of this outlook is shown in Figure 1.1. It represents an inverted triangle where the size of each layer correlates with the amount of processed BMS data. The first research objective is to investigate security considerations for only singular BMS. The next layer considers local networks through which multiple entities and BMS may communicate. Finally, at the third and largest layer, we consider communication between multiple systems, such as vehicles, considering one or more BMS.

[1] This work is supported by and received funding from “EFREtop: Securely Applied Machine Learning - Battery Management Systems” (Acronym “SEAMAL BMS”, FFG Nr. 880564).

[2] This work is supported by the OPEVA project that has received funding within the Key Digital Technologies Joint Undertaking (KDT JU) from the European Union’s Horizon Europe Programme and the National Authorities (France, Belgium, Czechia, Italy, Portugal, Turkey, Switzerland), under grant agreement 101097267. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or KDT JU. Neither the European Union nor the granting authority can be held responsible for them.

Research questions. Based on the problem statements, we derive three research questions (RQ):

RQ1 How to implement a unified and secure wireless battery management system design for the internal sensor and external diagnostic communication?
RQ2 How to design a lightweight security architecture for battery management systems for local network communication?
RQ3 How to realize an efficient and secure design for battery management system data acquisition and propagation to external end systems and services?

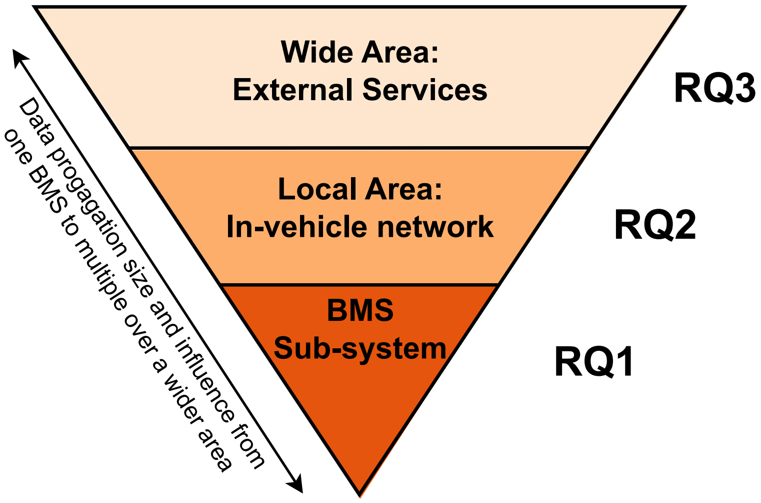

Wide Area:
External Services
Local Area:
In-vehicle network
BMS
Sub-system
RQ3
RQ2
RQ1
Data propagation size and influence from one BMS to multiple over a wider area

Figure 1.1: Reverse triangle representation of the BMS data propagation impact from source to the remote services. Each layer is observed as a separate research question in terms of the system design security: RQ1 - concerns one BMS sub-system with battery cells and sensors, RQ2 - local, e.g., vehicle network that can contain multiple BMS or related control units, RQ3 - wide area with remote and cloud services, covers multiple BMS and host systems.

1.3.1 Secure wireless internal and external BMS readout design

When discussing the security aspects on the BMS sub-system layer, one needs to study data propagation in relation to the module deployment and also account for the communication design. Traditionally, BMS rely on the wired and segmented communication between modules starting with battery cell sensors. However, there are several limitations when using the traditional wired communication design with BMS [22, 37, 38, 39, 40]:

- *Assembly cost.* The current wired BMS solutions are costly as they require additional manufacturing steps when installing the small and specialized printed circuit board (PCB) both inside and outside battery cells. The need to wire every individual sensor with the communication interface is also design-demanding and can often result in a non-optimal placement of the sensors in the battery cells' housing. This increases the overall cost of the design as special adjustments need to be made on the PCB-level during the automated manufacturing process [38].
- *Scalability.* With the realization of new BMS functionalities and use cases discussed earlier, BMS are becoming larger and more complex. The current design and topologies suffer from increased maintenance complexity [41]. They can also result in several other drawbacks such as electromagnetic interference and physical connection failures [42].

- *Placement and area.* Wires require more space and add additional weight. The placement of modules, and sensors, in particular, adds an additional layer of complexity. In certain circumstances, the use of physical wires may also interfere with the vital placement of battery sensors, reducing the overall precision of the BMS diagnostic functions.

Table 1.1: Wireless technology challenges that need to be addressed when considering system design for BMS.

Challenge	**Deployment**	**Description**
Restricted throughput	Internal	Data throughput rate from the sensors to the BMS should meet the required functional requirement transfer rates [22, 40].
Interference	Internal & External	Intereference can happen between technologies that use the same frequency band, e.g., 2.4 GHz, which is used by LR-WPAN, Bluetooth, ZigBee, and WiFi [43, 44, 45, 46].
Multipath propagation	Internal	The communication needs to be reliable even under the enclosed obstructive environment [37].
Widespread availability	External	The system design and protocols need to be supported across a large plethora of portable devices.
Wake-up	Internal & External	To reduce the connections with battery cells, and support sleep slates, the design should provide a reliable wake-up feature.
Security concerns	Internal & External	Wireless networks are vulnerable to eavesdropping, node capturing, remote attacks, and other malicious incursions [47, 48, 49].

In order to alleviate the limitations imposed by a wired design, it is possible to replace it with a wireless one. However, choosing a wireless technology is not an easy task, as there are several challenges to be taken into account when addressing our research goal in relation to the intended use cases. The goal is to find a solution that replaces the current costly and high-maintenance wired design with a wireless design and also works for both communication aspects related to BMS, i.e., internally, between the sensors and the battery pack controllers, and externally, with diagnostic readout devices. For both internal and external communication, there are two main use cases to consider. One, which is concerned with the intra-module battery cell sensor readout, traditionally relying only on the wired analogue connections, and two, external diagnostic readout for battery health and status analysis. The external diagnostic readout use case is becoming more and more important as it is one of the desired applications that complement primarily the battery passports, but also battery second life, battery swapping, and charging use cases. Both the internal and external wireless challenges are listed in Table 1.1. Based on these aspects, the first research question is formulated as follows.

Research Question 1

How to implement a unified and secure wireless battery management system design for the internal sensor and external diagnostic communication?

Under "unified", we consider a centralised system design solution that is able to meet the challenges and requirements of both data readout approaches by relying on a cost-effective and flexible wireless design. A unified design provides a better balance of production costs and helps avoid interference that can occur with different wireless technologies. This also means relying only on one wireless technology, as opposed to multiple ones, under the same communication environment.

As an alternative wireless approach and potential solution, we consider radio-frequency identification (RFID) technologies. RFID offers a short-distance communication that suffices for BMS use cases. Specifically for our targeted application, near-field communication (NFC) can be used as it offers a relatively short, but secure and robust connection interface. Currently, open questions remain regarding the necessary design considerations, performance overview, implementation requirements, and design requirements tied to the security aspects and newly open threats that arise from using NFC within the BMS environment. A research investigation should be carried out aimed at answering system design questions when considering NFC and similar RFID technologies.

To provide a complete design, a thorough security threat research analysis must be conducted that takes into account the use of modern BMS with the NFC technology. The expansion of BMS functionalities also expands the possibilities of potential attack vectors. An attacker could manipulate the temperature sensor values, leading to false detection of thermal runaway, or completely mask real damages [50]. A security model solution must be provided to mitigate security threats. Since these devices are produced in large quantities and typically have one communication interface point per one or more battery cells, research should consider a lightweight security architecture that provides security at the device authentication level and the data security level while being performance- and cost-efficient.

1.3.2 Security architecture for BMS network communication

The traditional deployment of BMS limited its interaction with external devices. However, with the increase in complexity, BMS are found communicating with multiple devices as part of a local area network. For vehicles, this could be an internal bus network connecting several or all Electronic Control Unit (ECU) components. From our perspective, the main communication from BMS would come with a central gateway unit, eletric vehicle charging controller (EVCC), the main engine ECU, the dashboard controller, and any external device for the purpose of diagnostic analysis or updates. By opening up, BMS have become vulnerable, and adequate protection against malicious attacks must be ensured [30]. Based on the current State-of-the-Art (SotA) on BMS and similar control systems, we identify the following main threats when considering a local network that can be initiated either remotely or locally with previously obtained unauthorized access [51, 29, 49, 52, 53, 54]: (i) attack on data integrity during configuration and updates, (ii) espionage, i.e., eavesdropping, (iii) compromise of security materials or exchanged data via Man-in-the-Middle (MitM) attacks, (iv) physical device compromise.

Simply relying on traditional security principles and architectures may be tempting, but relying on a purely asymmetric cryptographic security suite, such as a complex Public Key Infrastructure (PKI), can lead to many pitfalls and problems down the road. Maintaining certificates and other cryptographic material can be very difficult, inaccurate, and detrimental to the user experience, especially in vehicles where starting and driving the vehicle depends on security verification. In addition, many traditional PKI are resource and performance intensive and might not be usable under standard vehicle and BMS environments. This does not mean that asymmetric cryptography should be ignored, but rather adopted with the use of symmetric cryptography approaches for simpler, but still secure, design models. Based on these challenges, the second research question is derived.

Research Question 2

How to design a lightweight security architecture for battery management systems for local network communication?

From the research question formulation, under local network communication, we consider the communication architecture with in-vehicle local networks, e.g., bus systems, as the most important use case of complex BMS applications. However, the presented design should also be applicable to other local networks, e.g., for smart power grids, as long as they meet the targeted system design requirements. For this study, the following research requirements were derived from the posed research question.

Research requirement 1. The research investigation should be focused on "lightweight" aspects, i.e., it is intended to observe BMS as a constrained embedded device. A few BMS today have the performance capabilities in running high-end security protocols and hosting demanding security architectures, with most of them still being hosted by limited microcontroller unit (MCU), similar to the ones found in other ECU. The targeted security mechanisms need to consider these aspects and allow for the functional execution of standard BMS system monitoring and diagnostic service controls under the introduced security load.

Research requirement 2. The security architecture should not only be lightweight, but also provide full protection against established BMS security threats by protecting the devices and the data in terms of confidentiality, integrity, availability, and authenticity.

Research requirement 3. The design should be flexible in terms of updating and providing new extensions, both from the security configuration side, but also from the general system and network side. This could be seen by changing certain control units, communication protocols, or even changing the network or BMS topology.

1.3.3 Secure BMS data acquisition and propagation

As we have discussed, the production and use of battery cells for EV and other applications has increased dramatically in recent years, with no sign of decline [5, 3]. Accordingly, solutions are being analyzed that would enable easier and more efficient replacement of battery packs at the end of cell life for the purpose of reducing global waste [11, 13]. This process requires easier tracking and monitoring of batteries through their linked BMS controllers.

However, battery lifecycle tracking for second-life use presents one of the major BMS-associated challenges today. Battery lifecycle tracking needs to account for functional accuracy, lightweight and pervasive use, not interfering with safety BMS functions, adequate data storage and propagation, flexible and adaptive readout communication technology, and above all, security design for both the device and data reliance. Currently, there is no standard that addresses BMS for large-scale applications and extended functions [26]. And thus, we come to our third and final research question.

Research Question 3

How to realize an efficient and secure design for battery management system data acquisition and propagation to external end systems and services?

We see two major groups of challenges associated with the third research question that need to be addressed when considering external services and data propagation with BMS:

Challenges with "BMS data reliance". Modern BMS generate large amounts of data due to their monitoring and diagnostic processes. A key factor in system design is to provide efficient and accessible logging of BMS data [55, 25]. Relying only on local storage modules would prove insufficient, as they most likely would not possess enough storage memory to support a battery pack during its entire system lifetime or be able to meet the requirements of the battery passport initiative [15]. Therefore, to support full logging services, we might want to also rely on modern cloud connectivity, a concept that is gaining ever-more prevalence also in other fields of BMS research [56, 57, 58]. However, relying only on pure cloud connectivity for BMS data propagation has three major disadvantages [55, 59]:

1. Cloud services require a continuous internet connection, which might not always be available, e.g., in case of an incident in a tunnel, in which important event data would be lost.
2. More transmission layers also mean higher delays and interruptions with other processes.
3. Cloud bases are tied to particular owners, where in case of changes in data legislation or business model, BMS-related data might be impacted and made temporarily inaccessible.

Challenges with "BMS security". As with the previous research questions, the proposed system design needs to be able to provide both adequate and lightweight security. Most research on BMS cloud connectivity does not consider security leaving this question open. Security for BMS data transfers must consider all layers of data transmission, with each layer providing adequate authentication and secure data transfer [55, 49]. Data confidentiality must be ensured in the external services, as the devised BMS data blocks must only be able to be read by authorized parties, which is ensured by the appropriate secure BMS data block design.

Based on the set challenges, we want to investigate and propose a design solution for the implementation of secure data acquisition that considers both on-premise and cloud data transfer [60].

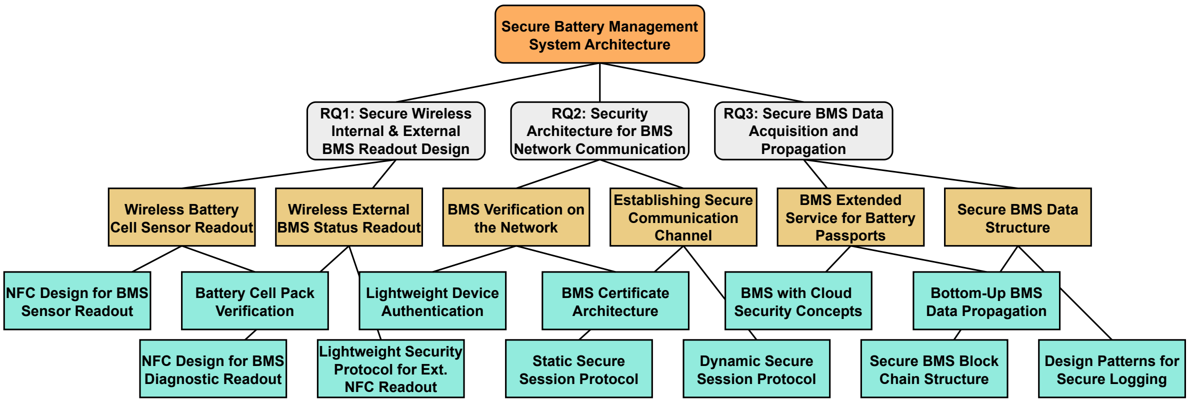

Secure Battery Management System Architecture
RQ1: Secure Wireless Internal & External BMS Readout Design
RQ2: Security Architecture for BMS Network Communication
RQ3: Secure BMS Data Acquisition and Propagation
Wireless Battery Cell Sensor Readout
Wireless External BMS Status Readout
BMS Verification on the Network
Establishing Secure Communication Channel
BMS Extended Service for Battery Passports
Secure BMS Data Structure
NFC Design for BMS Sensor Readout
Battery Cell Pack Verification
Lightweight Device Authentication
BMS Certificate Architecture
BMS with Cloud Security Concepts
Bottom-Up BMS Data Propagation
NFC Design for BMS Diagnostic Readout
Lightweight Security Protocol for Ext. NFC Readout
Static Secure Session Protocol
Dynamic Secure Session Protocol
Secure BMS Block Chain Structure
Design Patterns for Secure Logging

Figure 1.2: Graph presenting the core dissertation hypothesis subdivided into research questions, research sub-questions, i.e., research goals, and overview of the resulted contributions, in that hierarchical order.

1.4 Contributions

To answer the previously stated problem statements and research questions, several contributions were made during the scope of the long research process. They are contained under the research sub-questions, which are considered as "research goals". Figure 1.2 provides an overview of the related contributions for each set of research goals, where each research goal is derived from specific research questions centered around the research field of *Secure BMS Architecture*. The research goals and their contributions are further listed and described in more detail.

During the work on the dissertation, a goal was set to allow for a reusable design between different derivations of the BMS topology and to meet not only current but also future system requirements.

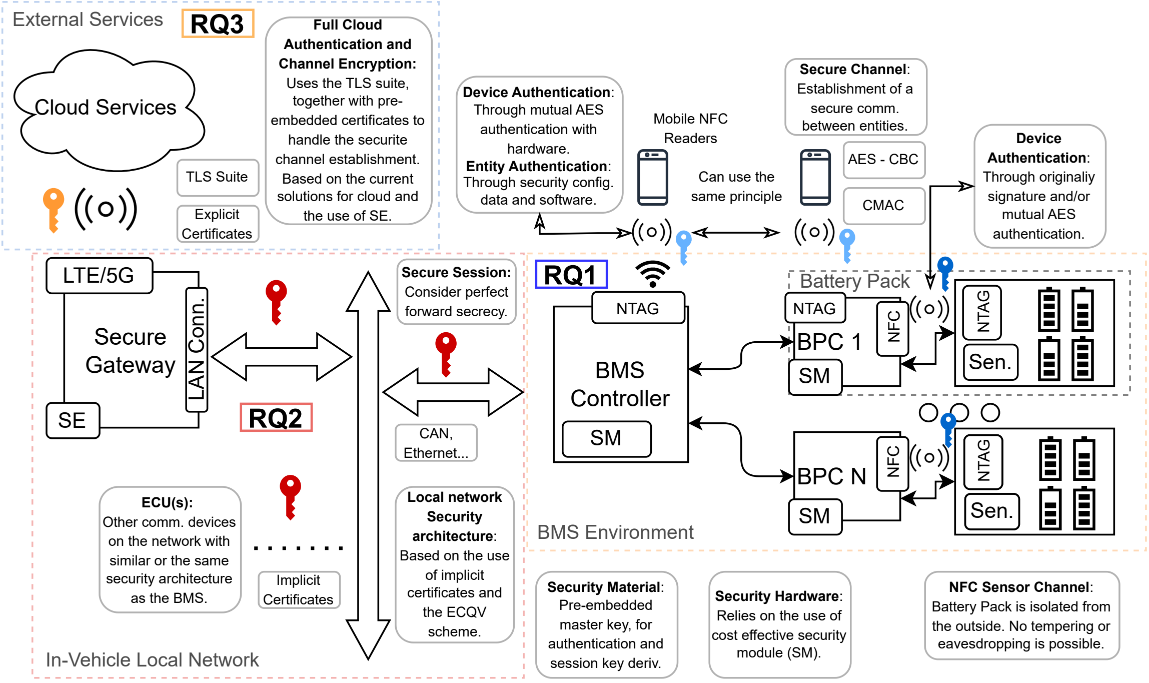

External Services
RQ3
Cloud Services
TLS Suite
Explicit Certificates
Full Cloud Authentication and Channel Encryption: Uses the TLS suite, together with pre-embedded certificates to handle the securite channel establishment. Based on the current solutions for cloud and the use of SE.
Device Authentication: Through mutual AES authentication with hardware.
Entity Authentication: Through security config. data and software.
Mobile NFC Readers
Can use the same principle
Secure Channel: Establishment of a secure comm. between entities.
AES - CBC
CMAC
Device Authentication: Through originaliy signature and/or mutual AES authentication.
LTE/5G
Secure Gateway
LAN Conn.
SE
RQ2
Secure Session: Consider perfect forward secrecy.
CAN, Ethernet...
ECU(s): Other comm. devices on the network with similar or the same security architecture as the BMS.
Implicit Certificates
Local network Security architecture: Based on the use of implicit certificates and the ECQV scheme.
In-Vehicle Local Network
RQ1
NTAG
BMS Controller
SM
Battery Pack
NTAG
BPC 1
NFC
SM
NTAG
Sen.
BPC N
NFC
SM
NTAG
Sen.
BMS Environment
Security Material: Pre-embedded master key, for authentication and session key deriv.
Security Hardware: Relies on the use of cost effective security module (SM).
NFC Sensor Channel: Battery Pack is isolated from the outside. No tempering or eavesdropping is possible.

Figure 1.3: Overview of the research contributions on proposing system design guidelines for the secure BMS and its extended network. It consists of a BMS environment, an in-vehicle network, and external cloud services.

Effectively, any engineer wishing to build a secure and robust BMS with extended service connectivity to the outside world and wireless connectivity for internal and external readouts can be guided by the insights gained in this dissertation. To this end, we have created a list of key system components and security propositions embedded in the conventional BMS, vehicle, and cloud design, as shown in Figure 1.3. The Figure contains the main three domains of interest previously discussed, starting with an BMS sub-system environment encompassing contributions to *Research Question 1* (RQ1), the In-vehicle Local Network domain with contributions to *Research Question 2* (RQ2), and finally, External Services with a focus on cloud extensibility with contributions to *Research Question 3* (RQ3).

1.4.1 Wireless battery cell sensor readout

The use of wireless technology with BMS is becoming an attractive topic within the research community, with several concepts and designs published by both academia and industry. However, the use cases considered are limited to intra-module communication only. Most system design solutions still rely on wired communication between battery cell sensors and module controllers, but as mentioned in the problem discussion above, this imposes many limitations on the design of modern BMS. In addition, the solutions offered for sensor readout systems do not take into account the security requirements of modern battery packs. Here, the most important aspect is the validity of the battery packs, which should be checked via the adjacent interfaces. Therefore, in the presented research, we focus on providing an adequate system design solution for wireless sensor readout by also considering battery cell pack module verification. In response to the presented limitations for wired and wireless sensors, we present a system architecture that uses RFID, specifically NFC. Wireless NFC technology offers relatively low-cost system design solutions based on active/passive readout with the potential use of energy harvesting, decoupling them from main power sources and allowing flexible sensor placement within batteries. It also offers better security features compared to most other wireless technologies thanks to its short communication range and frequency band.

Novel concepts. We are the first to deploy NFC as a wireless communication technology for BMS for the targeted use cases. As mentioned in a recent study on wireless BMS [22], the use of RFID, especially NFC, with BMS is a relatively new area of research with very little previous work. To complement this, we contribute with a NFC-based system design solution by focusing on the use of design applicability and exchange protocol, as well as an authentication model to verify battery pack validity. We experimentally demonstrate the applicability of the presented solutions as part of the BMS test prototype and evaluate it in terms of its system performance and response to security threats.

1.4.2 Wireless external BMS status readout

In the future, rechargeable batteries will be used as portable devices that can be quickly replaced, either between charging stations, or after the capacity has degraded, to be stored for an extended period of time before being reused for second-life use in other application systems. To extend the wireless readout of the battery cell sensors and complement the unified design, which is one of the focus points of the first research question, we rely on NFC technology. Using NFC for external readout provides flexibility that can be easily achieved by using many NFC-enabled devices, such as mobile phones, for quick and easy readout. However, because the communication would be done with an external device, a full security suite including data authentication and provision of a secure channel must also be considered.

Novel concepts. We are, to the best of our knowledge, the first to introduce a secure and wireless system design for external readout of BMS and battery pack modules. We also contribute with a novel security layer with a dedicated data structure on top of the NFC link layer and a lightweight protocol design for secure communication. We demonstrate and analyse the communication experimentally using real hardware. The communication is also analysed in terms of wake-up procedure to consider sleep states with two proposed models evaluated based on measured wake-up cycles and energy draw.

1.4.3 BMS verification on the network

Communication in a closed local area network relies on verification and knowledge that each device participating in it is valid. To achieve this, we rely on the use of novel implicit certificates, in particular, the Elliptic Curve Qu-Vanstone (ECQV) scheme. Implicit certificates have an advantage here in that they are smaller and more efficient than explicit certificates and their associated schemes. While the use of ECQV in vehicular networks has already been proposed, no work has yet been done that considers the use of these types of certificates and their security schemes with BMS and related services. In addition, we consider an efficient and lightweight device authentication protocol that precedes certificate generation and exchange. Once authenticated, devices receive implicit certificates from a secure gateway Certificate Authority (CA) that allow them to derive their public/private key pair, which can then be used for in-network authentication and session key derivation.

Novel concepts. We propose and demonstrate the realization of a complete lightweight security suite with device authentication and certificate derivation based on the ECQV scheme for internal system networks that consider the BMS as an integral unit of the network. To the best of our knowledge, we are the first to present a complete security suite for BMS with an implicit certificate architecture.

1.4.4 Establishing secure communication channel

After devices on a network have been authenticated, they can communicate with each other. A BMS may want to communicate with a central gateway for the purpose of sending valuable diagnostic data for predictive maintenance in cloud services or communicate with a EVCC during the charging process. Device communication requires establishing a secure session with a dedicated session key. We focus on exploring and deriving two future-proof approaches for this activity: a (i) static key derivation (SKD) method and a (ii) dynamic key derivation (DKD) method that also provides perfect forward secrecy, albeit with a slightly increased overhead. Both methods are based on the previously established implicit certificates and the ECQV scheme. The SKD method is based on the Diffie-Hellmann method with Elliptic Curve Digital Signature Algorithm (ECDSA) authentication while using implicit certificates. The DKD method is a novel method, not presented before, based on the proven Station-to-Station (STS) protocol for forward secrecy. Since the time execution is of paramount importance, we also focus on deriving and providing additional optimization steps for the DKD method.

Novel concepts. We are the first to present the secure session establishment for BMS with implicit certificates. We are also the first to present the DKD protocol for the ECQV scheme using the STS protocol. No previous work has combined these approaches. In addition, we also demonstrate and show the performance and evaluation of the aforementioned methods in both automotive implementations using the Controller Area Network (CAN) protocol between BMS and a hypothetical EVCC, as well as in other embedded devices to accommodate different controllers with varying performance levels.

1.4.5 BMS extended service for battery passports

BMS generates a large amount of data during the lifetime of each battery pack. This data would be processed, stored, and then processed again both on the local on-premise site and on external cloud and end-user systems. With the introduction of battery passports and similar initiatives, the use cases of BMS have expanded and must now accommodate the new security requirements. The work in this dissertation serves as an extension of many other published research contributions in the area of BMS cloud services that focus on data-driven models and cloud-enhanced algorithms. What we present is a hybrid layered model for a secure BMS cloud architecture. Security is analyzed at different layers, considering both the BMS sub-system, secure local gateway, cloud service, and users' end systems. The integration of the logging function with the BMS would need to be done in the manufacturing phase to address the system design specifications. Here, we analyze and discover two design patterns that can be used at the system architecture level: (i) Embedded Platform to Memory (EP2M), a pattern for efficiently designing an on-premise data logging system, and (ii) Secure Embedded Logging (SEL), which provides design guidance for establishing a secure pipeline between main controllers and storage units.

Novel concepts. To the best of our knowledge, this is the first contribution in the field of cloud and external data processing for BMS that proposes a system design for secure data logging and propagation from battery cell sources to end systems. To support the design for secure on-premise data logging, we also present two design patterns and show their applicability with the BMS.

1.4.6 Secure BMS data structure

At the time of writing, there is not a clear solution for a secure data structure that addresses the handling and storage of BMS diagnostic data across multiple systems and different BMS topology derivations. A common data structure would be a necessity for the upcoming battery passport regulation, as it would allow for easier transfer and encoding between different OEM and users when transferring battery packs between different use cases and systems. Our goal is to propose a general BMS data structure that is independent of any topology or use case and that can be used to handle BMS monitoring and diagnostic data, while also taking into account the necessary data security requirements, i.e., user privacy, data integrity, and authenticity. The BMS diagnostic data stored with our proposed secure BMS data blocks can be used as intermediate storage for higher safety regulation and further distributed to external systems, e.g., an external reader, or a cloud system. The data structure is hierarchical and considers the connection between log data blocks, BMS blocks and system blocks, considering both the extension of the necessary static metadata of the battery passport and dynamic diagnostic data.

Novel concepts. The work in this dissertation is the first to propose and demonstrate a secure BMS data structure design based on generated blocks and inter-connected chain principle. The data blocks consider a hierarchical structure and data exchange with on-premise, intermediate, and cloud data propagation layers, and are designed to be adaptable to different topologies and use cases.

1.5 Outline of the Dissertation

The remainder of this dissertation is structured as follows:

- **Chapter 2** describes the **fundamental** knowledge for this thesis. This concerns background knowledge on BMS, its functionality and topologies, wireless technology, especially NFC, and security models and cryptography.

- **Chapter 3** includes the relevant **related work** for this dissertation. Primarily, it deals with wireless BMS, security investigations for BMS and vehicle systems, NFC and its security models, and implicit certificates as one of the core security topics behind the realized architecture.

- **Chapter 4** discusses the first part of the **design**. It explains the security and wireless connectivity using NFC with BMS, its system design, data structures, and security models, and deals mainly with the contributions associated with the first research question.

- **Chapter 5** deals with the second part of the **design** and presents design contributions associated with the second and third research questions. It presents how the security architecture has been developed for the BMS that includes the realization of security requirements, central security architecture for authentication and secure session derivation, secure data structure, and system design for external and secure cloud data propagation.

- **Chapter 6** explains the practical contributions behind the dissertation by discussing the realized and used test suite **implementation** that emulates the use of the presented design solutions in a real-world environment. It presents different layers of development, important hardware and software building blocks, code realization, and developed protocols and data structures.

- **Chapter 7** shows the experimental research analysis by a conducted **evaluation** relying on both practical and analytical tools. The evaluation considers all the major building blocks of the system design solution and it is divided into performance and security evaluation sections.

- Finally, **Chapter 8** presents **conclusion** and **future work** that summarize the research investigation, solutions, and results realized in this dissertation, as well as the recommendation on current and future open research points.

- In the **Appendix**, the full texts of eight published papers are attached that have been written during the duration of this dissertation, together with the respective summaries and descriptions of personal contributions.

CHAPTER 2 Background

Summary: *This chapter provides a summary of the theoretical fundamentals behind the concepts discussed in this dissertation. It begins with an overview of what BMS are and the important aspects considered in the research, including the battery passport and second-life use cases, before addressing the security and NFC aspects as one of the main building blocks of the proposed design. This chapter serves as a reference point for the topics mentioned in the rest of the dissertation.*

◇◇◇

2.1 Battery Management Systems

BMS are control devices responsible for monitoring and safety regulation of battery cell use [18, 19, 61]. Battery cells are connected either in series or in parallel and are part of a battery pack. A battery pack can contain multiple battery cells and is monitored by intermediate modules or the BMS itself [33, 61]. The primary function of a BMS is to monitor and control the power supply to the battery cells for the system that uses them. However, it can also control various safety control mechanisms, such as breakers, if potential hazards are detected. In this way, a BMS is able to shut down modules that exhibit abnormal behaviors [26]. They can also activate or affect other safety control systems, such as cooling, to control the temperature inside the battery packs. Table 2.2 shows the three main monitoring BMS groups and the associated protection mechanisms.

A BMS ultimately monitors important parameters of the cells, including their charge, voltage, capacity, temperature, pressure, and derived parameters such as SoC, SoH, depth of discharge (DoD), charging period, etc. The purpose is to ensure that the higher-level system, such as an electric vehicle, maintains its optimal use of the batteries [62]. They help prevent overcharging and over-discharging, which can lead to various problems, such as the degradation of the batteries' health [63] or, in the worst case, the thermal runaway caused by the rapid rise in battery cell temperature [26, 50]. They are also responsible for controlling charge equalisation, i.e., ensuring that the battery charging process is uniform across all cells. Table 2.1 lists the main BMS functions regardless of their topology or use case. Here we can see that the most important protection points are in the battery itself. An BMS works by protecting the connection points. If a less significant issue has been detected, a warning may be issued. If a more serious problem exists, the BMS has the ability to control the power deliverance, by either shutting down the system with a breaker or using a more controlled mechanism. BMS are irreplaceable controllers for Lithium-ion batteries (LIB) and other battery types used in a wide variety of systems today, but mainly among EV, smart power grid subsystems, home grid systems, and battery energy storage systems (BESS) [62, 58, 64, 61, 33].

In recent years, digitization has become more and more prevalent in traditional analog systems. BMS are no exception. Here we see the expanded use of external systems, such as the cloud, to provide advanced services:

- Profile tracking: tracking of individual user profiles based on their interaction with the targeted use case, e.g., vehicle battery usage for individual drivers. The established profiles can be used for more accurate predictions in the future or as input to diagnostic synchronization algorithms intended for vehicle fleets [41, 27, 58].
- Predictive maintenance: using the cloud as a collection of more powerful devices to run and predict key BMS diagnostic parameters such as SoC or SoH. These can be run as independent processes or in parallel in the form of digital twins [56, 65, 57].

In this dissertation, the use of the aforementioned advanced BMS functions is extended through design and implementation extensions with wireless readout, secure data processing, and a BMS data structure based on the chain principle, as seen in Figure 2.1. The use of secure data processing allows the received and derived observed BMS and battery data to be processed and transferred into secure blocks, via secure interactions with both internal and external components. The external readout relies on wireless technology or processing via a gateway, i.e., a cloud service, to extend the tracking of important battery pack lifecycle data.

Table 2.1: Main BMS functionalities in a system responsible for managing a set of battery packs.

Function	Parameters	Description
Monitoring	Raw values such as cell voltage and current, temperature, pressure, etc.	The tracking of the main battery cell parameters. Here, the prediction and analysis are done only on a rudimentary level.
Diagnostics	SoC, SoH, DoD, charging time, inner cell impedance, energy usage rate, etc.	Battery cells' status estimation based on the calculation of predictive values.
Cell balancing	Different parameters and indicators, primarily the SoC	Control of the cell charging and discharging process to ensure maximal battery cells' lifetime.

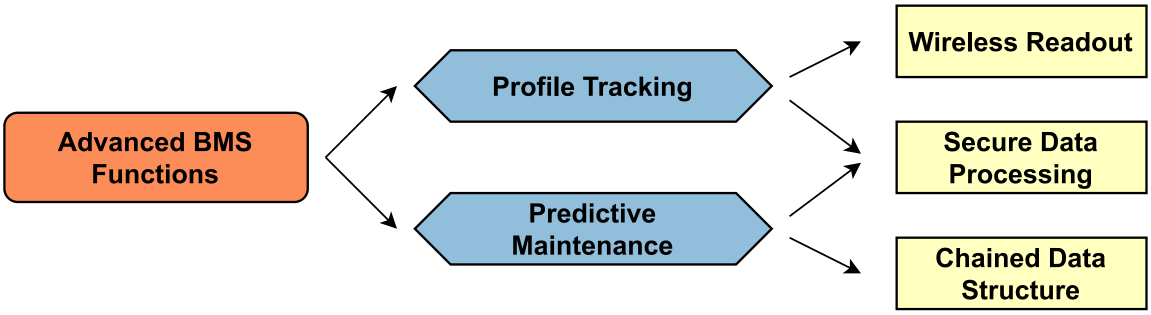


Figure 2.1: Advanced BMS functions derived from the ever-increasing digitalization of the BMS and their extensions. We observe two main use groups, one for battery profile tracking, and the other for predictive maintenance.

Table 2.2: BMS active monitoring parameters.

Target	Monitored parameters	Protection
Battery power	Under- & over-voltage, under- & over-current	Breakers and relays that can close the main connection between the battery packs and the remainder of the system.
Battery sensors	Low & high temperature, pressure changes	Breakers for immediate safety control, observed control of temperature cooling.
Operational	Grounding issues, current leakage, charging time, etc.	Different charging and balancing policies, warning status observation.

BMS can be deployed in a variety of network topologies. Each topology has different advantages and disadvantages and is characterized by its communication flexibility, size, cost, extensibility, and reliability. There is no clear answer to the question of which topology is better, but its use depends on the intended use case. In the literature, there are some differences in the description and naming of each topology. In this dissertation, the four most commonly used BMS topologies today are considered, namely [26, 22, 66, 67, 68]:

a) Centralized - Considers a single BMS controller that communicates with and manages all battery cells. Since all functions are concentrated in one main BMS controller, this type of topology is not suitable for large systems because it would require a more powerful controller and would also radiate more heat due to its multi-task configuration. It also requires many communication channels, one for each cell and sensor, and therefore, does not scale well with the increase in the number of battery cells. The advantage of this topology is that it is a cost-effective solution suitable for less demanding systems.

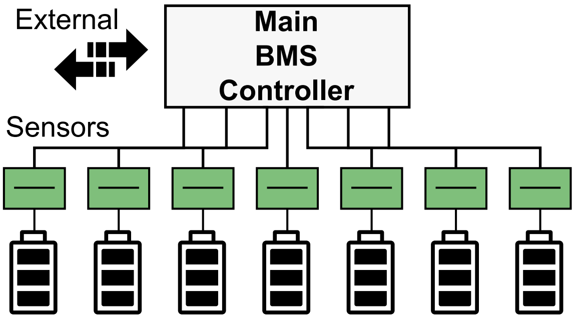


Figure 2.2: BMS centralized topology.

b) Modulated - There are several different specifications of how a modulated topology is described. In this work, it is considered a topology in which several equivalent control units are used instead of one central unit. Among them, one *leading* controller is selected, while the rest are *followers*. Each controller is connected to the next one. There is also an alternative modulated star topology where each follower is connected to the leading controller. This topology provides a good balance between price and performance.

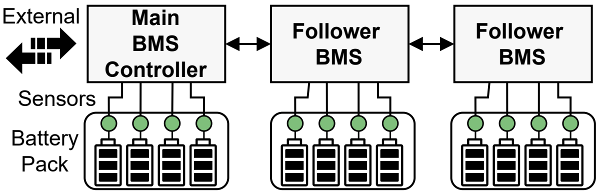


Figure 2.3: BMS modulated topology.

c) Distributed - This topology is very similar to the modulated, but rather than having several modulated BMS control units, smaller battery cell control units are used that communicate only with adjacent control units [68]. We call them Battery Pack Controller (BPC) [1]. Each BPC observes one battery pack that consists of several battery cells and sensors. They are often connected in a daisy chain, meaning that the main BMS controller only communicates directly with the first and last BPC. This allows for better fault tolerance, but they are generally more costly and envisioned for more complex systems, e.g., vehicles. This is also our main reference topology.

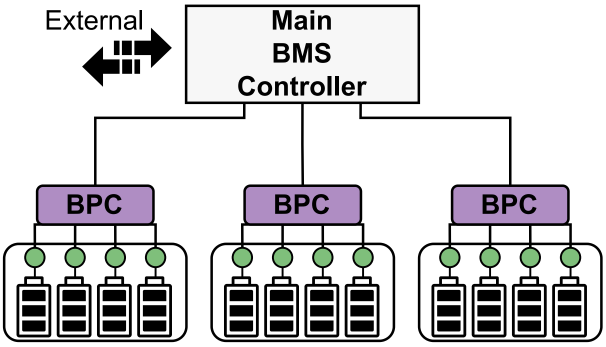


Figure 2.4: BMS distributed topology.

d) Decentralized - Rather than relying on only one main BMS controller, the decentralized topology allows the use of several BMS controllers, either independent or dependent on each other, as part of one overarching system. A decentralized topology can consist of multiple centralized, modulated, or distributed topologies. The advantage is allowing better distribution of battery cells in an environment, but this approach adds extra complexity and potentially cost.

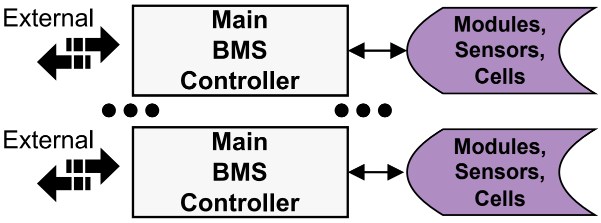


Figure 2.5: BMS decentralized topology.

Standardizations Overview

Several standards have been published associated with electric vehicles and batteries in general [12, 26]. We will list only some of the relevant ones here. Standards related to safety, security, and NFC wireless connectivity are mentioned in their respective chapters. *International Organization for Standardization (ISO) 6469-1* [69] specifies requirements for in-vehicle rechargeable energy storage systems, including consideration of electrical, functional, and simulated accident requirements. Similarly, *ISO 6469-3* [70] expands the specification by focusing on electrical safety requirements, e.g., safety requirements against electrical and thermal incidents, which primarily apply to EV. *Society of Automotive Engineers (SAE) J2288_200806* [71] presents a set of methods for determining the lifecycle of a EV battery. *IEEE 1679.1* [72] refers to the use of lithium batteries in stationary applications and provides their functionalities. In this standard, BMS is also referred to as an active management system.

2.1.1 BMS Data Structure

The vehicle ECU may periodically store relevant driver parameters in event data recorder (EDR) units [73]. These may or may not be connected to black boxes, devices that contain information from event registers, and can be used to assess the driver's condition after an accident [74]. While BMS are a part of the vehicle and their diagnostic data may or may not be collected by the assigned EDR, there are also

[1] Throughout this work, I will be referring to BPC as the intermediate control module devices responsible for charge balancing control and sensor data gathering of a group of battery cells. In literature, this module can be found under different names, such as Battery Cell Controller (BCC) or Cell Control Unit (CCU).

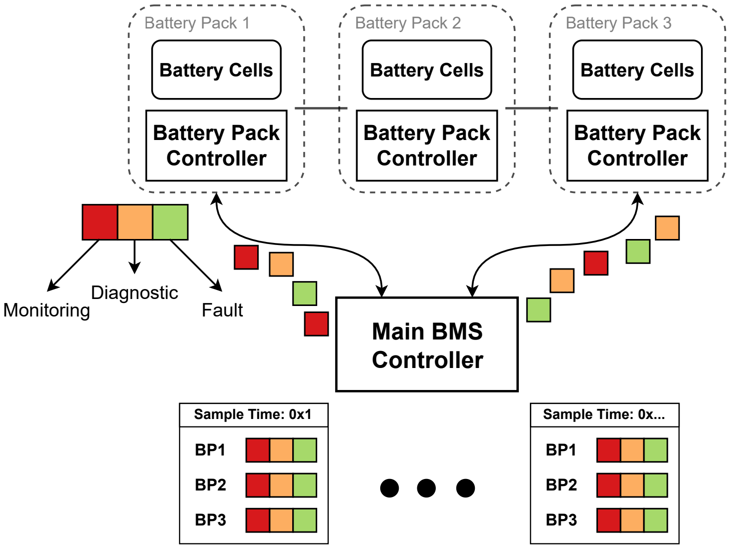


Figure 2.6: BMS data sampling model for distributed topology: The main BMS controller gathers the tracking data from individual BPC and stores them from each sampling cycle. It is responsible for its storage and processing.

other use cases where the logging of BMS data can be of great benefit. For a BMS, as for any other safety-critical system, it is extremely important to create log files that keep track of the system events. The log files must meet security requirements for authenticity and integrity [75, 76]. When log data is recorded, it must be ensured that it has not been altered since its original creation (integrity), and it must be guaranteed that it originates from a valid entity, i.e., that it is not a forged or counterfeit entry. Standard cryptographic schemes can be used for this purpose. However, this is not sufficient, as other decisions must also be made that take into account the method of data storage, software and hardware specifications, and the management of appending, tracking, and truncating cryptographically secured log data. These design requirements will be considered in realizing the design of the BMS data structure proposed later in this dissertation.

Independent of the topology and targeted use case, three main BMS data groups are observed:

a) Monitoring data - raw measured battery cell values, e.g., voltage and temperature.
b) Diagnostic data - derived results based on the internal analysis, e.g., detected over-voltage, under-voltage, high temperature, battery capacity imbalance, etc.
c) Fault data - raw date detecting a specific fault; generally tied with respective fault registers.

Figure 2.6 graphically shows the data collection process of a distributed BMS. As can be inferred, a BMS is responsible for managing a large amount of data. Each sampling time, which varies by the system but is typically in the range of 100 ms up to 1 s, can generate as much as hundreds of bytes of data per battery pack module. This data must be properly processed to accommodate both local storage and remote transmission functionality.

2.1.2 External cloud connectivity

Cloud computing has evolved in recent years to include a large list of services. This refers to any service that runs on external servers in data centers, hidden behind a wrapper from end users. They usually involve collecting and processing large amounts of data, from Internet of Things (IoT) to large industrial systems [77]. Recently, many papers have been published focused on combining the BMS with cloud concepts. The use of cloud service came out of necessity to extend the functionality of BMS, with current on-premise solutions lacking computational power, storage capacity for logging activities, and information distribution for easier user access [58]. Thus, the use of cloud systems with BMS enables the collection and processing of a large amount of BMS data to enable services such as remote monitoring and predictive maintenance, i.e., the advanced BMS functions mentioned in Figure 2.1 [78, 59].

Specifically, the use of cloud services with BMS enables [79, 58]:

- Using artificial intelligence models or digital twins for supporting the concept of battery cells' lifecycle monitoring and predictive support [56, 78, 80, 57].
- Providing increased computational and processing power for calculating the targeted BMS diagnostic parameters, e.g., SoH or SoC [59, 80].
- Enabling additional monitor for fault detections and improving battery age via a more accurate control of charging and discharging cycles [25, 57].
- Using the concept of "swarming" for data collection that can further be used for predictive maintenance or profile tracking when considering multiple systems, such as vehicle fleets [56, 25].

The importance of BMS cloud services has already led to several market solutions, such as the EV Logger from CSS Electronics [81] that offers offline and online logging of dynamic EV parameter data, with the BMS being one of the main units of interest. However, these only offer solutions per user and vehicle and are tied to the product. A cloud-only solution is offered by Bosch as "Battery in the Cloud", which is extensible to multiple stakeholders [82], with NXP Semiconductors also tapping into the market with digital twin and artificial intelligence services using BMS cloud connectivity [83]. It is believed that in the coming years, more OEM and other industry and academic partners will invest and collaborate on bringing cloud connectivity for BMS and related ECU as part of future EV development.

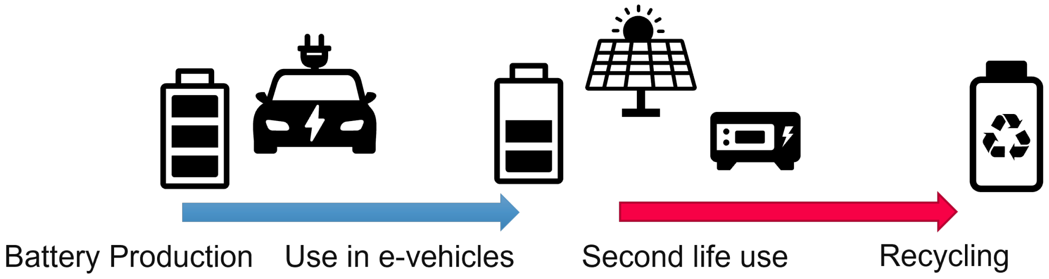


Figure 2.7: Batteries' second life lifecycle: After failing below 80%, battery cells used in e-vehicles can be repurposed to be used in less demanding applications, effectively increasing their lifetime before eventual recycling.

2.2 Battery Passport and Second Life

The second use of batteries is a concept that suggests the use of rechargeable batteries even when they are deemed inoperable for the current system in use [84, 14, 12]. An example would be EV, where batteries can be repurposed for other applications such as home appliances or less-demanding smart power grids before they are eventually recycled, as illustrated in Figure 2.7 [13, 85, 86].

Even when the same BMS and battery packs are present in two different EV, the rate of battery cell degradation can vary significantly. Several factors influence battery cell degradation, the most important being driving behavior, battery cell mechanical and chemical properties, and climate and weather factors, followed closely by BMS control characteristics and infrastructure, among others [87, 88, 89, 90]. These battery specifications help determine the classification of battery use cases, specifically under which context they can be used with a "primary" owner and under which for the "secondary" owner (for second life use case) [60]. Therefore, it is of utmost importance to be able to successfully monitor and record the changes of battery cells during their lifecycle when they are used as part of different utility systems.

The new European Union initiative proposes the concept of "*battery passports*" [15, 16, 91]. It entails that all individual battery cells should be traceable by a uniquely assigned identification, preferably readable by Quick Response (QR) codes. This is done to support the ethical concerns related to the mining of minerals and elements needed for battery cell production. In addition, the concept of a battery passport is also seen as a way to better regulate the use of battery cells, followed by second-life use and finally recycling for the purpose of reducing environmental waste. In this context, the battery passport is essentially a digital representation of a single battery cell that conveys important and relevant product information [15, 17, 2]. Currently, consideration is also being given to using the battery passports in conjunction with cloud systems, which could provide dynamically derived information in addition to static data to better support the functions listed in Section 2.1.2.

This dissertation contributes to support the idea of the second life of the battery and presents solutions that can help in the realization of applications for safer, and more secure tracking, storage, and processing of battery-related lifecycle information. The solutions presented in this dissertation, especially those focused on external secure wireless readout, can be considered for future battery passports.

2.3 Security Concepts

Security is a set of concepts and methods that define how to analyze potential threats, assets that need protection, and mechanisms of defence [92]. Under assets, we can consider any organization, user, or device that needs to be protected. Threats are any form of compromise that can be misused by a malicious party to attack assets. Under embedded devices, we often also see the term *cyber-security*, which is used interchangeably.

Each security protection can be considered under a different attribute. For information security, these can be grouped under the triangle of confidentiality, integrity, and availability (CIA). Furthermore, in the context of this dissertation, it is important to also consider authentication, authorization, and privacy [93, 94].

Confidentiality. Refers to an attribute of information protection that guarantees that no unauthorized party is able to read the information, i.e., the information can only be read by valid and intended parties. Information that is sent over an unsecured channel is vulnerable and prone to passive attacks, e.g., eavesdropping, and therefore must be protected, e.g., by encryption.

Integrity. The data and the information contained therein must not be manipulated or altered by unauthorized entities. Any alteration that should take place should be detectable. Maintaining integrity means that changes can be detected, but does not consider knowing its source.

Availability. Information should be accessible to valid parties at any foreseen time. Limiting or completely obstructing this information from being accessed is considered an attack on this attribute.

Authentication. Only valid entities should be able to communicate and send information to each other. The source sending the information must be valid. Even if the information is otherwise complete and true, it is considered a violation of this attribute if it was not sent by a valid source. We often consider two separate observations here: authentication of the source, i.e., the entity, and authentication of the information itself.

Authorization. The information can only be accessed from a specific access level. This level determines which information can be accessed from which valid entity. If the information is accessed by an unintended entity, both valid and malicious, it is considered a violation of the attributes.

Privacy. Data, and therefore information, exchanged between two valid parties should be able to convey the intended information without disclosing other undesirable information that reveals the personal preferences of the parties. It takes into account the retention, storage and handling of data.

For a system to be made secure, it must be augmented with an appropriate security model or architecture. Security engineering is a discipline that focuses primarily on tools, methods, processes, implementations, and testing of security concepts when building a system [95]. Any secure system development adheres to the following steps: (i) requirements definition, (ii) model specification, (iii) secure system design, (iv) secure system implementation, and (v) secure testing.

2.3.1 Cryptographic primitives

To protect against security threats, appropriate countermeasures must be considered. Security mechanisms are based on the implementation of different security models and architectures at higher design levels and security functions and primitives at lower levels [96, 95]. Several different security concepts have been considered throughout this dissertation, the most important of which are mentioned in this section.

Under security, we generally consider two cryptography principles:

- **Symmetric cryptography**: only one cryptographic key is used for security operations for both communicating sides, e.g. for encryption. Relying on this one shared secret reduces complexity and is often more performance-advantageous, but it also places the weight of protection on that secret. Symmetric cryptography, while it can be used to protect all security attributes, is today mainly employed after the initial authentication has taken place, for communication in secure channels and sessions.
- **Asymmetric cryptography**: two different cryptographic keys are used for security operations. One is a *private key* and the other is a *public key*. The private key always remains hidden on the entities side, while the public key is accessible to everyone, including malicious parties. The private key is used for signature generation and decryption, while the public key can be used for encryption and signature verification. Rivest–Shamir–Adleman (RSA) and Elliptic Curve (EC) algorithms are prominent examples used under this cryptography [92]. Asymmetric primitives are rarely used on their own and often employ PKI and "certificates" for the purpose of authenticating public keys.

In this dissertation, both symmetric and asymmetric cryptography are considered. Symmetric cryptography is used for parts of the BMS model that do not benefit from larger PKI architectures and where performance plays an important role. Asymmetric cryptography is typically considered when employing the implicit certificate architecture discussed in Section 2.3.3 and for communication with cloud and end-user systems.

Channel encryption

To protect the confidentiality of information, channel encryption can be used. Both symmetric and asymmetric cryptography provide different encryption primitives and algorithms. Encryption can be based on *Stream* or *Block Cyphers*. In this dissertation, we will rely only on block cyphers, in particular, the algorithm Advanced Encryption Standard (AES) and its modes of operation [97]. AES is the best-known and most widely used block cypher today. Its advantage, besides its proven security, is its performance and the fact that it can be easily implemented in hardware [95, 98]. However, conventional block cyphers only protect the confidentiality of data, while for integrity and authenticity, additional operations must be added to extend the functions, e.g., with Message Authentication Code (MAC), as presented in Section 2.3.1. Alternatively, it is possible to rely on the Authenticated encryption with associated data (AEAD) algorithm modes, such as Galois/Counter Mode (GCM) or Counter with CBC-MAC Mode (CCM), to achieve protection of all attributes under one function [99]. One mode is not necessarily more secure than the other, as it depends heavily on how it is implemented and handled. The main security weakness of block cyphers stems from their modes of operation and any information that they may leak when processing the payload data up to different blocks. Therefore, it is necessary to carefully consider what requirements each AES mode of operation must meet in our own design.

Message Authentication Code (MAC)

For integrity protection, it is often necessary to send additional data that is used during the verification process on the receiving end. One of the most common security primitives used for this process is the MAC. MAC is based on the symmetric cryptographic principle, meaning that it uses a shared secret to guarantee that the process of generating the additional authentication tag has been done from a valid source [95]. This key is often derived during the same key derivation function (KDF) process along with the encryption key. Therefore, a MAC guarantees not only the integrity of the data, but also its authenticity. The authentication key always has a fixed length, regardless of the size of the input data. The sizes are key-dependent and vary from 128 to 512 bits. MAC rely on the use of other cryptographic primitives as the underlying function, such as hash functions or cryptographic algorithms [100, 101]. For the work done in this dissertation, we rely mainly on the use of the algorithms Cryptographic-MAC (CMAC), also known as One-Key CBC MAC (OMAC) [102], and Hash-MAC (HMAC) [101].

Since a MAC is applied to data to compute its authentication identifiers, the question arises as to when such a process should take place. It can take place either before, during, or after the encryption of the data. Thus, there are three main modes for applying a MAC alongside encryption:

- *MAC-then-Encrypt*: MAC is first used to calculate the authentication tag using plain data as the input, with both the plain data and the tag being afterward encrypted.
- *Encrypt-and-MAC*: similar with the MAC-then-Encrypt, in that the data the MAC is first performed on the plain data, however, only the plain data is afterward encrypted.
- *Encrypt-then-MAC*: first the encryption takes place on the input data, followed by the MAC operation on the generated encrypted data.

For the purpose of this work, we will be only relying on the Encrypt-then-MAC mode, which is considered a more secure approach that does not require additional handling [103, 104].

Key exchange and derivation

Key management presents one of the most important security measures. The disclosure of a secret key could partially or even completely compromise the security architecture on which a system is built. Therefore, it is of utmost importance to ensure that keys are correctly derived, regularly updated, and securely exchanged between parties.

In our design, we distinguish between keys used for authentication (*private & public keys*) and for secure session communication (*shared secret key*). In addition, each device has a *master key* that is used for initial device authentication and as input for session key derivation. The secure session key can be derived either statically or dynamically. By SKD we refer to keys that are directly bound to their certificates, i.e., their derived private & public keys. They usually rely on traditional methods of key derivation that do not provide additional security benefits, such as Diffie-Hellman key derivation [105], i.e., where keys are derived from a multiplication between the private key of one entity with the public key of another entity and vice versa: $S_K = PrvK_A * PubK_B = PrvK_B * PubK_A$. Under SKD, as long as their private and public keys are not updated [2], so will their session key not get updated. On the other hand, the DKD considers the *perfect forward secracy* as an additional feature. Here, a new key is derived for each communication session that is independent of previous or future session keys, i.e., it has a sufficiently high entropy and is not bound to their derivation. This also makes it independent of the current certificate and its private & public keys, but this usually comes with a performance cost, which can sometimes be high on constrained embedded devices. In the literature, this dynamically derived key is often referred to as an ephemeral secret key.

The actual calculation of the key is done with a KDF. These are functions that generate a shared secret based on a given input. This input usually consists of a *pre-shared* secret, e.g., a randomly generated nonce, with possibly additional data, such as salt. Often the previous key is also used as input. In the design presented in this dissertation, the master key is also used as input in the first iteration. A KDF ensures that even if the session key is compromised, it does not reveal any information about the master key. The generated output is then processed as the key. The output should be at least as long as the key size, but if it is larger, it can always be truncated. Various KDF are used today, many of which are based on the use of strong cryptographic hash functions [106].

Secure hardware

While security primitives and functions can be implemented via software on any system that supports their processing requirements, this is often not as advisable. Simple software can be vulnerable to vulnerabilities due to unprotected side channels, buffer overflows, or memory leaks [107]. Security in modern devices is usually executed in a special trusted hardware segment, e.g., *TrustZone* [108]. Communication between the normal and the trusted zone must also be secured to prevent external probings [109].

There are several different security hardware extensions, with common ones being [95, 110]:

- *Security co-processor*: sits right alongside the main processor and is enclosed as one entity, but is usually limited in resources and offered functionality.
- *Hardware Security Module (HSM)*: bigger and more powerful full security-dedicated devices, meant to be used in larger systems and environments that can afford them.
- *Secure Element (SE) & Trusted Platform Module (TPM)*: smaller cost-effective chips that are added as isolated external components that communicate with the main processor, offering a large array of security functions and secure memory storage.

These hardware components also provide hardware-accelerated execution, i.e., their architecture is optimized so that dedicated security functions and primitives can be run much faster in comparison to when implemented directly on standard processing hardware [111]. This is especially important in constrained embedded devices, e.g., BMS controllers, where such acceleration can be highly beneficial.

[2] Usually occurs with the update of the certificate.

2.3.2 Security assessment

Evaluating a security model can be a very difficult process. The proposed design should provide a response to each plausible threat scenario. In order to accurately list and verify each potential threat, a detailed analysis must be performed by creating a *"threat model"* [112, 113]. There are many different threat analysis solutions available, both from academia and industry. Certain threat models, such as Microsoft's STRIDE model [114], can be used for general use cases, while others are intended for more specialized cases, such as for the automotive industry [115, 116], for models that combine safety with security assessment [117], network-based security threat models [118], etc. They usually follow the main principle of dividing the assessment into two phases:

- *Pre-design threat analysis*: listing of requirements and necessary threats based on previous models and experience done in the earlier phases of the development.
- *Post-design threat analysis*: the security evaluation is done after the design has been realized, built, and implemented, following the security requirements defined in the pre-design analysis.

A non-hardware security assessment can be performed using either formal or informal methods. An example of a formal analysis is the Burrows–Abadi–Needham (BAN) model [119]. Today, many software and automatic models are also used, including Scyther and Tamarin [120, 121]. Although they are generally more accurate, it can sometimes be demeaning and difficult to use formal models for a larger and more complex security architecture. In addition, they can be misleading if used inappropriately. Therefore, for faster and more comprehensive analysis, informal methods are used, which can be either theoretical, graphical, or software-based and automated. Some of the better-known methods are attack trees, CVSS, and STRIDE models.

Threat modeling. For the pre-design analysis, *attack trees* are often employed. They are used to graphically model attack points and directions, with the root or higher levels representing a group of threats or general threats that focus more on assets, and the lower levels or leaves used to note specific attacks [112]. Another useful threat modeling tool is *data flow diagram (DFD)* [122]. While it is primarily intended for software, web, and larger service systems, it can be modeled for other environments as well. It consists of defining entities, assets, communication flows, and countermeasures. The placement of each of these elements is important to represent accurate context. A similar graphical approach that can also be used for threat modeling is *goal structuring notation (GSN).* Although not originally intended for security analysis, it can be modeled to also support the representation of dependencies between assets, threats, countermeasures, and any residual risks [123].

Common Vulnerability Scoring System (CVSS). The CVSS is used for a numerical assessment of a security threat [124]. It gives a score based on a set of inputs that can be entered manually, probably from a previous threat analysis. The CVSS is capable of specifying three metrics: Base, Temporal, and Environment metrics. The most commonly used metric is the base metric, which gives the threat a qualitative value based on which the severity of the threat to the particular system can be determined. Several versions exist, with the latest being v3.1 introduced in 2019.

STRIDE. It is a model used to categorize security threats into different classes: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege [92, 115, 114]. More often than not, a threat can have multiple associated threat classes. Based on the knowledge of these classes, it is easier to further devise appropriate countermeasures.

Security standards

The *Common Criteria (CC)* is an international standard used for the security evaluation and certification of secure products. The standard itself proposes different terminologies. The products that are to be evaluated are referred to as Target of Evaluation (ToE). CC provides various guidelines and options for how this evaluation can take place. A Security Target (ST) is a set of specifications about what security features a ToE should possess. They are defined based on a set of security requirements of a Protection Profile (PP) derived from various groups and communities. After a security assessment, a Evaluation Assurance Level (EAL) is assigned on a scale of 1 to 7 [92, 125].

With the increase of awareness for cybersecurity threats that specifically targeted vehicles, a necessity came for a dedicated standard. *ISO 21434* [126] fills this role by proposing a set of cybersecurity requirements for the design and production of ECU and other electronic components in a vehicle. A similar standard that also addresses modern vehicles, specifically EV, is the *ISO 15118* [127]. This standard will play an important role in the future, also in regards to BMS, as it provides guidelines for vehicle-to-grid communication, including with respect to charging stations, and potential security architecture requirements for trust and key provisioning for this environment. Although not primarily aimed at security specifications, *ISO 26262* [128] is a well-known standard used in the safety domain that is often brought together with other security standards when analyzing system solutions, especially in the automotive environment.

There also exist organizations such as OWASP that support security assessment by providing tools and solutions for security assessment, especially considering modern threats and software [129].

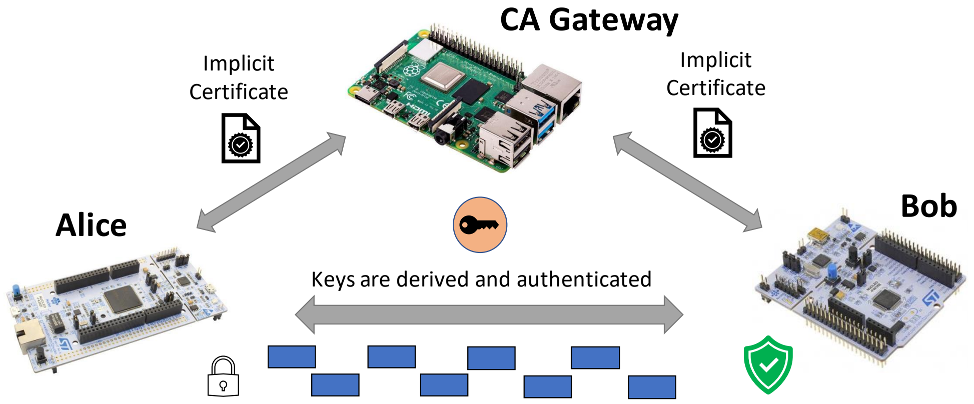


Figure 2.8: Implicit certificates targeted architecture: it contains a central authority (CA) gateway device responsible for generating and transmitting implicit certificates to local devices for intra-network secure communication.

2.3.3 Implicit certificates

When we talk about asymmetric cryptography, we cannot ignore the concept of certificates. Certificates are a collection of data bytes that can be used to authenticate entities through a link between an identity and a public key. If both devices trust a common CA and they have valid certificates issued by it, they can authenticate each other by relying on other security mechanisms, such as signature checks by associated public and private keys [130]. The most common certificates in use today are *explicit certificates*, on which the global Internet architecture and the most widely used security protocol Transport Layer Security (TLS) are built. The format most commonly used for explicit certificates is the X.509 format. It can contain multiple entry points such as ID, public keys, validity checks, etc., but generally has a size of up to 1 kB [131].

Implicit certificates offer the possibility of a significantly smaller certificate size. In theory, the smallest certificate today can contain only an ID and a reconstruction of the public key, which for a 32-bit

EC cryptography size, would come to $32B$ (ID) $+ 2 * 32B$ (Pub. and Priv. Keys) $= 96B$ size. In practice, these certificates would be relatively larger according to the various formats, but in all cases would fall below the size of the corresponding explicit certificate. The main reason for this is that the public keys of the devices are not included in the certificates, but rather are *"implicitly"* derived during the authentication request. The advantage of smaller certificate sizes could be beneficial for constrained and embedded applications [132], as shown in Figure 2.8, but also for networks where BMS are used.

There are several different security protocol schemes that rely on the implicit certificates, with the ECQV being the most popular and researched one [133]. It is also the scheme that has been used as the basis for the local network security for BMS proposed in this dissertation. Its certificate derivation protocol is listed in Table 2.3 with notations in Table 2.4.

Table 2.3: ECQV scheme: Deriving an implicit certificate, private, and public keys from the CA.

Client (C)		**Central Authority (CA)**
$\alpha \leftarrow Rand(),\ P_C = \alpha G$		
	$\xrightarrow{ID_C,\ P_C}$	
		$k_C \leftarrow Rand(),\ U_C = P_C + k_C G$
		$Cert_C \leftarrow Encode(U_C,\ ID_C)$
		$E_C \leftarrow Hash(Cert_C)$
		$S_C = E_C k_C + prk_{CA} G\ (mod\, n)$
	$\xleftarrow{S_C,\ Cert_C}$	
$\hat{E_C} \leftarrow Hash(Cert_C)$		
$prk_C = \hat{E_C}\alpha + S_C\ (mod\, n)$		
$\hat{U_C} \leftarrow Decode(Cert_C)$		
$pub_C = \hat{E_C}\hat{U_C} + pub_{CA}$		
$verify(pub_C == prk_C G)$		

Although the implementation is relatively flexible, there are some important design points that must be considered when employing implicit certificate schemes. As with explicit certificates, implicit certificates can also suffer from denial-of-service (DoS) attacks, but in a different manner. Specifically, this is done by a malicious entity that constantly sends forged requests to generate implicit certificates to the CA. This attack can be defended against by appropriate filtering mechanisms. Another important point is that implicit certificates can potentially be vulnerable to "form forgery attacks", which can be formalized as a generalized Wagner's birthday problem [134], when building long *certificate chains*, with chains of length four or longer [133]. Therefore, it is recommended not to exceed three levels of certificate chains in multi-tier architectures.

2.4 Wireless Communication with NFC

Near-field communication (NFC) is a close proximity wireless technology based on existing RFID systems and standards. The communication is based on either the principle of capacitively or inductively coupled systems. The difference lies in the transmission medium, with capacitively coupled systems using an electric field, while inductively coupled systems use a magnetic field. Regardless of the appli-

Table 2.4: Notation list used in the dissertation for the implicit certificate operations.

Symbol	**Description**
ID_X	Unique identifier of entity 'X'
α	Random value for the certificate request on the client side
P_X	Certificate request value from entity 'X'
G	EC generator point
k_X	Random value to guarantee an unique certificate for entity 'X'
$U_X,\ EX,\ S_X$	Key construction data for entity 'X'
$Cert_X$	Generated certificate for entity 'X'
$\hat{U}_X,\ \hat{EX},\ \hat{S}_X$	Key construction data generated by entity 'X'
$prk_X,\ pub_X$	Private & public keys of entity 'X'

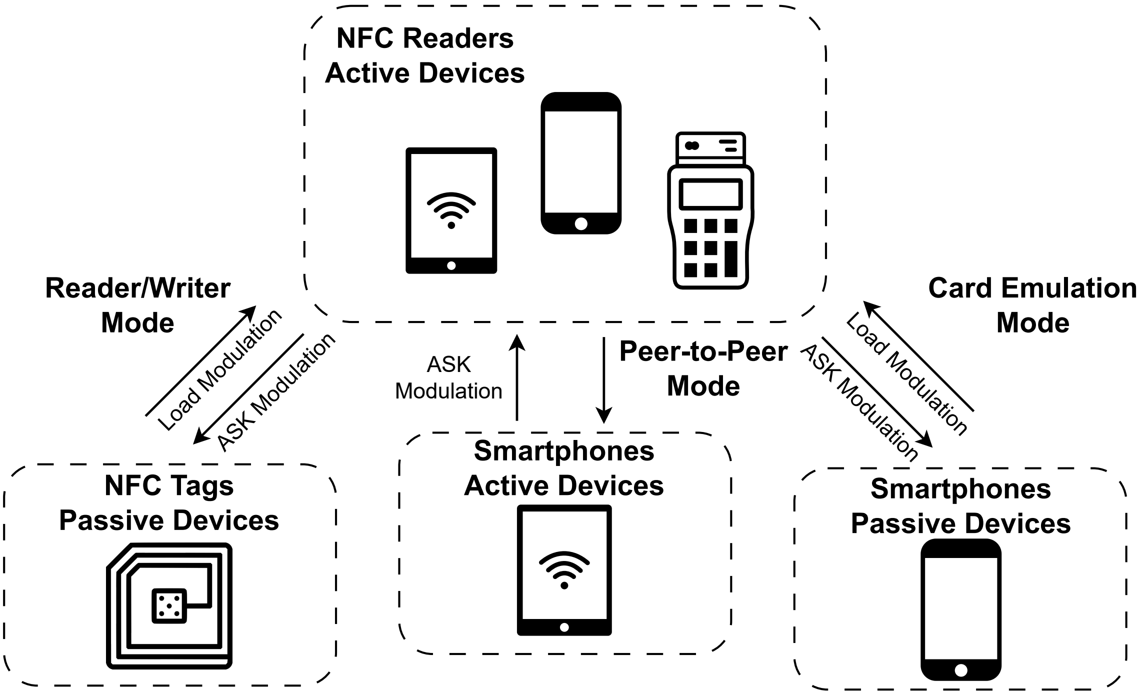


Figure 2.9: NFC operating modes. a) Reader/Writer mode: The communication is started by the active device (NFC reader, smartphone) over a passive tag, b) Peer-to-Peer mode: communication of two active devices where both can provide the field, c) Card Emulation mode: an otherwise active device (smartphone) initializes the communication and acts as a passive device. Active devices rely on the ASK, while passive on the Load modulation.

cation and the standard used, NFC uses a frequency of 13.56 MHz [135, 136]. It relies on the amplitude-shift keying (ASK) as the modulation scheme [137]. The communication field used with the NFC is suitable for both data and energy transmission [138]. This means that an active device with a power source is able to fully power and communicate with a passive device that does not itself have a continuous power source, with all power being supplied via energy harvesting through antenna coupling. There are several standards used for its specification, the one on which this work being the communication specifications defined in ISO 18092, as well as the ISO 14443, ISO 15693, and ISO 21481 specifications intended for RFID data link communications [139, 140, 141, 142]. In addition, key NFC architectural and data structure specifications are based on the proposals of the NFC Forum [143].

NFC can communicate at different ranges, with a typical range being about 10 cm. Data rates are typically 106, 212, or 424 kbps, with some devices supporting data rates of 848 kbps. The communication range, field strength, and position between the devices in question determine the overall detection and transmission latency. Messages are based on the NFC Data Exchange Format (NDEF) message format for the data link structure, a widely accepted approach to data encapsulation that provides low message overhead [143, 144].

Today, NFC is used for many different applications and use cases, ranging from payments, access control (e.g., for vehicles, in buildings, etc.), tickets and e-cards, etc. [145, 137, 146, 138]. It is also used as an auxiliary device for pairing methods such as with devices that use Bluetooth technology, thanks to its short range [147]. This wide range of applications is made possible thanks to its three main operating modes as seen in Figure 2.9 [137, 148]:

- *Reader/Writer mode.* The devices in this mode are divided into *active devices*, or those that are actively powered and can power other devices via energy harvesting, and passive devices, which require energy and communication initialization. The active devices must first detect and initialize communication between the passive devices before the actual transmission of the payload can take place. Instead of providing power during initialization, the active device must provide it throughout the whole communication period. The passive device can also send data back to the active device by relying on the field already established and using "load modulation". This is also the mode on which the design presented in this dissertation mainly relies.
- *Peer-to-Peer mode.* In this mode, both communicating devices are "active devices". This means that they must have their own power source, rather than relying on being powered from the other device. The device that is transmitting is also the one that has established the electromagnetic field. This device is referred to as the *initiator* if it started the communication, whereas the other device is referred to as the *target*. Since both are active devices, they rely only on the ASK modulation for communication.
- *Card Emulation mode.* Combining the characteristics of the other two modes, the card emulation mode enables the passive device to act as the initiator, i.e., to start the communication with a reader device. This can be useful in applications where the passive device would actually be a smartphone capable of storing multiple smartcards.

In the context of this work on BMS, the NFC technology has a disadvantage in its short range compared to other wireless technologies [22], but it provides an advantage in its resilience to interference, security threats, and better accessibility with other NFC-enabled devices [149].

2.4.1 NFC security

In terms of wireless proximity, the NFC has an advantage over other wireless technologies due to its short range, making remote and probing attacks difficult. However, several approaches have been demonstrated that enable network eavesdropping, such as the NFC Gate [150]. Actual attacks are difficult to perform in the real world because NFC standards and the applications that use them are constantly being updated, and many special considerations must be made to make these eavesdropping attacks possible. One of these is the time required for communication. Most applications rely on an NFC transmission, which typically takes a shorter time duration, making it difficult to conduct the attacks in that time frame. However, there are still several threats that are not directly prevented by the NFC standard [151, 152].

Based on the previous research, the following threats and attacks are discussed under NFC that can be grouped into the following three main categories [136, 153, 148, 154]:

- **Data-targeting attacks**. As NFC is a wireless technology and therefore uses radio frequency (RF) waves for communication, any device capable of intercepting and recording messages sent between communicating devices would be able to *eavesdrop* on that connection. This could be done with special applications and a suitable antenna. Likewise, attacks such as *data insertion* or *data modification* that target the integrity of the data could also be launched. These attacks would be more difficult to launch because the attacker would have to exploit modulation at the physical layer to modify messages in a short time frame.
- **Channel attacks**. Any kind of attack that targets the communication channel itself rather than the data or devices would be considered under this category. Specifically, any kind of MitM or replay attacks that would be possible to run under the specific NFC environment. However, these attacks, similar to the data-targeting attacks, would be difficult to pull due to the low number of communication occurrences, short range, and generally fast readout time [3].
- **Device functionality attacks**. While data manipulation and insertion might be difficult to conduct, attacks such as *data corruption* are more prevalent, as they are only limited in trying to make the targeted data unreadable. These kinds of attacks could also be extended to completely block the functional use of the NFC communication through a variation of a DoS attack.

Although the aforementioned security threats mainly affect either data, communication channels, or device functionality, there are also attacks that directly target hardware. Attacks that exploit vulnerabilities in side channels, such as power-glitch attacks, power consumption analysis, laser attacks, and memory usage exploitation, could be launched against devices themselves that rely on NFC, such as smart cards [155]. Since these attacks are device-specific and not directly related to the communication technology used, i.e., NFC, they are not considered a focal point for investigation in this dissertation.

[3] Since the data sent via NFC applications is usually relatively small and not intended for large scale data transfers.

CHAPTER 3

Related Work

Summary: *This chapter discusses relevant related work and the current state of the art. Since both wireless and security concepts with BMS are relatively novel concepts, it is necessary to understand their origins and see what relevant fields and issues have influenced current research development and the one presented in this dissertation. To this end, we take a deep look into current research on BMS security models and related wireless BMS applications, alongside research on vehicle security, cloud connectivity, and related NFC and implicit certificate security models, which are one of the core domains of this doctoral thesis.*

◇◇◇

BMS have traditionally been analyzed under very specific hardware and software disciplines, primarily involving research on batteries, diagnostic calculations, and more cost-effective and optimized system design. With the increasing digitalization of traditional BMS solutions, the research direction has shifted to include topics from other relevant areas, particularly networking and security. Figure 3.1 shows an overview of the relevant topics and subtopics of this dissertation, organized by their general domains. Relevant research papers on these topics were analyzed in detail, and the most important publications are described in the following sections.

3.1 BMS and Wireless Communication

Wireless BMS is a relatively new concept that has regained prominence in recent years. One of the earliest works combining BMS with the wireless design was proposed by Lee et al. [38]. In their pioneering work from 2013, a design for a WiBaAN protocol that operates in the 900 MHz frequency band and achieves data rates of up to 1 Mbit/s is presented. The intended use case is communication between the main BMS controller and individual battery cells. This means that the focus is kept on the centralized BMS topology, but due to the limited literature definition of different topologies from this time, it can be assumed that the proposed protocol could also be used for other topologies in a limited range. However, this aspect, combined with the limited use of the mentioned frequency band and the manufacturing cost, could be a limiting factor for modern BMS. Nevertheless, this is still a notable work that has laid the foundation for future wireless BMS concepts.

In terms of local network-based BMS solutions, Faika et al. [156] propose a design for wireless BMS that benefits from traditional IoT network architectures. In their proposal, a lightweight IoT protocol is presented for the leader election algorithm considered for BMS use cases with external readout. The design is independent of the chosen local network protocol, but the main focus seems to be placed on the IoT design, where the BMS controller is seen as just another node in the network, making the design rather general.

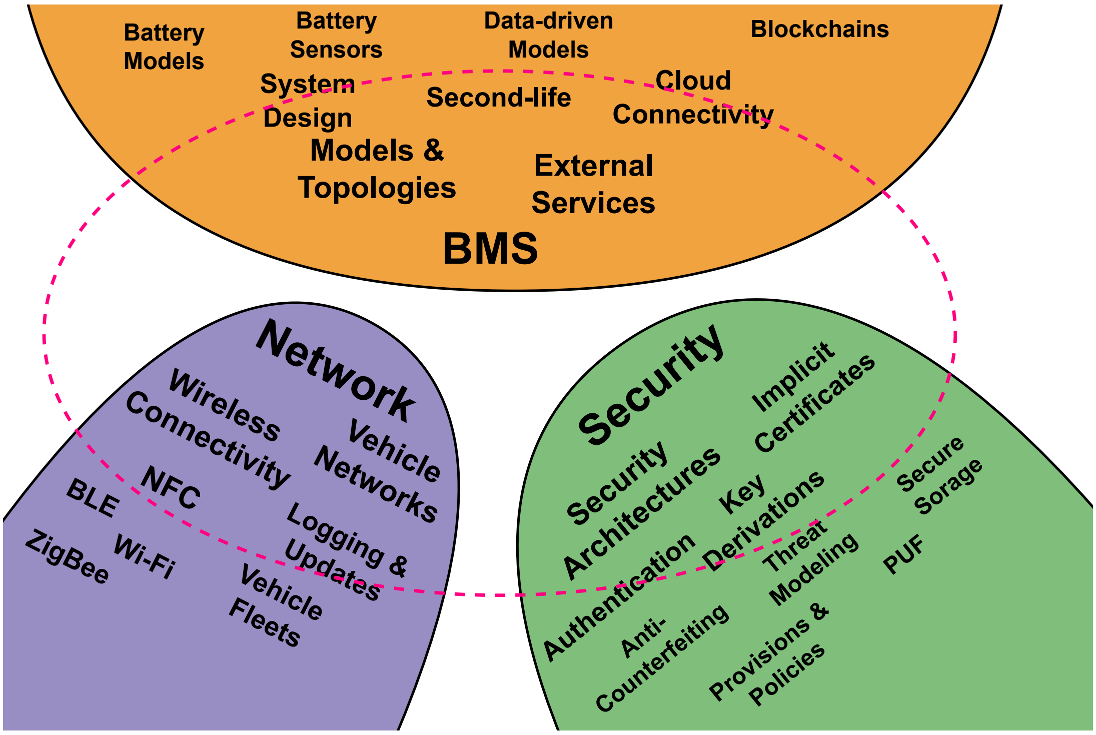


Figure 3.1: Research topics and domains behind the work of this dissertation. Red: main contributed topics.

There have been also several publications that address the broader spectrum of wireless technologies and their application with BMS. Research conducted by Bansal and Nagaraj [157] compares several different wireless technologies for BMS, including Bluetooth, ZigBee, Wi-Fi, and NFC, using a distributed topology model for BMS. No optimal solution is found, but several suggestions are given with advantages and disadvantages for each of the technologies used. Samanta and S. Williamson [22] present a more modern survey on the wireless BMS technologies and an overview of upcoming challenges. The authors point out the importance of storing a large amount of data and having a suitable architecture to handle it. Another important issue is the security of the BMS connected to the Internet, although as pointed out, its sustainability in the real world has not yet been adequately evaluated. To better understand the related research on the most studied wireless technologies under BMS, we will look at them separately under Bluetooth Low Energy (BLE), Wi-Fi, ZigBee, and RFID, summarized in Table 3.1.

3.1.1 BLE

Bluetooth and its low-power cousin technology BLE have been considered in several different BMS architectures and use cases. Bluetooth has also often been the technology preferred by industry when demonstrating wireless BMS capabilities, primarily because of its commercialization and ease of installation [23, 158, 24]. Nevertheless, the intended use cases focus primarily on intra-module communication only, i.e., communication between the main BMS controller and the follower or BPC. Bluetooth and the newer 5.x BLE standards have a limited data rate, which also depends heavily on the channel's noise level [159]. This limitation could often make it difficult to meet the targeted reliability rate required for safety and automotive standards when considering data throughput with BMS. Despite that, several innovative research works have been conducted on this topic.

Table 3.1: Overview of the research SotA and challenges with the most popular BMS wireless technologies.

Technology	Current Research	Challenges
BLE	Shows promising direction with both lab and real-world deployments, but with only use case on the communication between the main BMS and BPC.	It offers low-power consumption and good security properties, but with a relatively low data rate, communication reliability, and security updates.
Wi-Fi	Research with a focus on smart controllers, but mostly as a demonstrative channel, with not much specific technology application research.	It would add extra manufacturing complexity and system cost. No clear communication interference analysis has yet been done in real settings.
ZigBee	Previously had a strong research focus. Primarily presented under lab and specialized environments.	Similar to BLE, it presents low data rates, but also concerns over safety and security standards in automotive.
RFID	Strong focus with initial wireless BMS research. Has the potential, being industry and automotive-compliant.	System design, hardware limitations, data rates, and additional costs, with some solutions answered in this work.

For example, Shell et al.[45] present the use of Bluetooth technology for electric go-karts to reduce susceptibility to failure and facilitate maintenance. De Maso-Gentile et al. [44] consider an alternative hybrid approach by integrating a Bluetooth gateway into the conventional BMS CAN infrastructure. The paper is mainly experimental and practical and does not go into depth regarding the modification of the Bluetooth network stack.

One of the recent papers on BLE and BMS is by Rincon Vija et al. in which an improved version of the well-known Time Slotted Channel Hopping (TSCH) protocol is proposed by implementing a scheduling mechanism based on the low-latency deterministic network group acknowledgment method [21, 160]. The authors claim that they are able to achieve 100% of the network reliability with low power consumption using this method. The method has been tested in an observed environment primarily targeting the intra-module BMS wireless communication.

3.1.2 Wi-Fi and ZigBee

In addition to Bluetooth, Wi-Fi and ZigBee have also been considered for use in the BMS environment. Huang et al. [161] present a smart sensor prototype for battery packs based on the use of Wi-Fi. The authors present a design that enables communication between individual battery cells and the main BMS controller. The idea is to perform modulated operations that are normally performed on the intermediate units, i.e., BPC, bypassed by redistributing parts of the operations to the smart cell sensors. Wi-Fi is used in this work as a channel for the cell balancing controller. However, it should be noted that the focus of the paper is on innovative cell balancing and smart cell sensor control and that Wi-Fi was used primarily for demonstration and testing purposes. Similarly, Gherman et al. [43] utilize Wi-Fi as a carrier technology, but only for demonstration purposes of the wireless technology, with the focus of the paper being more on integrating the wireless BMS into a single system on a chip along with the charging controller.

Rahman et al. [46] also present an architecture for wireless BMS and demonstrate its use with ZigBee technology. The authors were able to show a successful demonstrator capable of operating correctly under their particular test setup and transmitting values, such as cell temperature, from the battery cells to the main module. It should be noted, however, that ZigBee suffers from unstable channels, low data rates, and security concerns compared to Wi-Fi and Bluetooth, so its applicability for BMS outside of laboratory environments is still an open question [22].

Contribution. All cited papers show insight into how the 2.4 GHz technology can be used for wireless intra-module communication in a BMS sub-system. Each technology mentioned, i.e., Wi-Fi [1], Bluetooth and ZigBee suffer from the same general problem, namely interference when operating in complex environments, as they would all share the same bandwidth and therefore compete for the use of available channels. We avoid this problem completely by focusing on NFC with its short range and 13.56 MHz channel frequency.

3.1.3 RFID

One of the first studies on wireless BMS came from the research of Schneider et al. [162] in which a design approach is proposed for wireless sensor readout for the BMS utilizing the RFID technology. However, the main focus is made on galvanic isolation rather than system and communication design. The design also does not take into account modern BMS topologies, as they were limited at the time of the research. But the work was quite novel for the time and opened the doors for further research in this area.

Despite numerous research papers by two separate groups, one interested in wireless BMS and the other in theNFC technology, the combination of these research groups was met with limited interest from the research community, as also indicated by the recent survey by Samanta and S. Williamson [22]. However, as also discussed in the paper, several conclusions are drawn about the different uses of wireless technologies, with Bluetooth and Zigbee being considered suitable for the targeted use cases due to their low power consumption and complexity, but also pointing out that they have limitations in terms of data transfer, security, and reliability when used in real-world applications. These are just some of the key challenges addressed by the wireless NFC design presented in this dissertation.

3.2 BMS and Security

Security with BMS is a relatively new topic that is slowly becoming more and more prominent. It arose from the need to protect the BMS as an important and complex unit within a vehicle and similar systems. Current SotA has mainly focused on general theoretical BMS security model analysis. Kumbhar et al. present such a model that considers the BMS as part of an IoT network and discusses the threats and potential countermeasures, but at a very high and general system level [49]. Similarly, Lopez et al. [51] analyze the threats in the BMS and IoT environments and list similar concerns.

In regards to threat modeling, Khalid et al. [30] investigate the BMS security, but from a different perspective. The paper focuses on threat identification and presents a framework for fault and threat detection. As the authors mention, there is currently no specific standard that addresses BMS security, and analysis must be performed based on an individual use case basis. Scripad et al. [48] examine EV threats in the context of using BMS. While it does not address specific attacks, it lists potential threats in

[1] We are aware that, at the time of writing, there are higher frequency bands used with these technologies under newer standards, particularly with Wi-Fi, e.g., 5 and 6 GHz. However, their use with the wireless BMS would need to be considered separately and is an open issue for future work.

terms of the damage they could do and how they could impact key BMS processes, such as overcharging or discharging battery manipulation, and their impact on the safety of the system.

BMS are often analysed together with other important ECU, specifically, with EVCC. A EVCC is a device that is responsible for relying on and controlling the charging process, and thus can also provide the communication link between BMS and external services [163]. In this field, Fuchs et al. [53] provide a solution for secure communication between a BMS controller and the EVCC. In their design, they specifically focus on the recommendations of industry standards and the EVITA project [164] by providing a system design that considers the incorporation of TPM for security operations. In a recent survey by Babu et al. [52] an analysis was conducted for the current research on lightweight security solutions for dynamic charging systems. In their analysis, they specifically focus on the properties of lightweight protocols and compare different models and their performance. Although not directly related to BMS, the threat and use case analysis performed by the authors is very similar to the general threat model also found in BMS. This is not unexpected since, as mentioned earlier, both domains are generally covered under the same comprehensive complex system, i.e., EV, and should be explored together in the future.

BMS can also be observed from the perspective of other vehicle components alongside the charging controllers. A recent study and analysis by Brighente et al. [28] analyses BMS alongside the charger, in-vehicle communications, and engine controls, and lists the unique features of EV compared to traditional fossil-gas vehicles. The threats associated with BMS are similar to those mentioned in previous papers. The main threats are DoS, manipulation of diagnostic data, malicious code injection, and spoofing. It shows that despite the growing interest and number of research publications in the field of BMS security, there are still many unanswered questions and a lack of standardisation.

Contribution. As it can be concluded from the aforementioned citations, most research publications on BMS and security focus on high-level theoretical threats and system analysis. This leaves many questions unanswered because security applications should not only consider high-level analysis but also provide practical experimental analysis and appropriate design solutions to functional and security requirements of modern BMS. This dissertation expands the notion of BMS security and provides one of the first system design security solutions for BMS considering the overall architecture, protocol integration, and HW/SW development. The researched theoretical BMS security models are used as the basis for the security assessments performed in this thesis.

3.3 BMS Logging and Cloud Extensions

Cloud services are becoming increasingly important in the BMS field, and numerous research publications have appeared in recent years [25]. Like any other vehicle or general industry systems, BMS are systems that generate a large amount of data. Today, this data is mainly processed offline on the main BMS controllers themselves. However, these services are limited in their capabilities. A more powerful server is able to collect and process large amounts of data, not just from one but from many BMS, and process it for various purposes, such as predictive safety control [58]. While cloud services are an already established field, the use with BMS is a relatively new concept with many open questions.

As noted in the 2020 paper by Li et al. [59], real-time tracking of the driving conditions of an electric vehicle is difficult to track with the cloud system because the recording time at the moment can be anywhere from 10 to 30 seconds. Therefore, the current SotA in cloud applicability to BMS is largely focused on providing adaptive solutions for the data-driven models and dynamic calculations of SoH and SoC parameters. This is done to reduce the high computational demands on BMS controllers and to enable not only fast and accurate computations but also predictive analysis. Several research works have been proposed in recent years. The work by Li et al. also proposes a model for estimating internal capacitance and resistance in battery packs [59]. These parameters are then used for the overall estimation of the SoH. The estimation is done using the temperature data and is optimized by the Kalman filter. A fuzzy logic model is also proposed to reduce the observed noise and improve accuracy. While the model shows an optimistic prediction that the maximum estimation error is only 4%, it is unclear how well the model scales with the increase in complexity and size of the BMS, so further research in this area would be necessary. A similar idea using more conventional machine learning approaches was proposed by Wu et al. [78]. They present a data-driven algorithm on the cloud side that is indented for SoH estimation. It uses machine learning methods by first performing feature selection to extract the measured battery data from the charging process, followed by a random forest regression algorithm as input for building the battery degradation model, and finally applying this model for SoH estimation. The secondary goal is to keep the noise in the model as low as possible.

The use of BMS with the cloud is also associated with the use of the novel concept of digital twins. Li et al. [56] describe the use of digital twin concepts as a BMS IoT solution for the comparison between the measured battery data and the data estimated by the cloud. The model is based on the use of adaptive extended H-infinity filters and uses particle swarm optimization for the state estimation algorithms.

Another use case where cloud systems can be used with BMS is *data logging and tracking*. Depending on the use case, BMS data sampling rate can be large enough to tolerate any delays that might be introduced by the cloud extension. One of the early pioneers from the industry comes from Fujitsu in the paper by Tanizawa et al. [80], which envisions a cloud service for sharing battery-relevant information between different BMS. One problem addressed in the paper still remains a challenge to this today, namely how to successfully exchange battery life cycle information during battery replacement, i.e., when handling batteries for their second life. An early concept is proposed that is able to collect, store, and provide this battery-related information using a cloud system. The cloud system is aimed to additional provide easy access for locating and monitoring batteries and tracking changes in battery characteristics. For a complete monitoring application, Friansa et al. [65] propose a battery monitoring system for smart microgrid systems. However, the design presented is relatively generic and could be applied to other IoT or BMS-based solutions. The evaluation result shows a total BMS data accusation time of 19.54 ± 18.00 seconds, which is relatively long, but it must be taken into account that this evaluation was performed using the implementation tools available in 2016. Similarly, Yang et al. [57] present an architecture-level design for cloud-based BMS that can be used for easier and more efficient use of BMS control functions, e.g., related to SoC or SoH estimations, thermal management, cell balancing, etc. They provide a framework that can be used for different BMS derivations called Cyber Hierarchy and Interactional Network (CHAIN). The goal is to use the framework as a starting point for a cloud-based BMS, but no specific details are provided on how to implement the data collection mechanism, leaving it up to developers based on use case requirements. For other specialized use cases, such as the use of lithium-ion battery cells with BESS, Taesic et al. [41] propose a monitoring and fault diagnosis system built on cloud management control. While the design targets specific functions and setup considerations, it can also be adapted to be suitable for affiliated BMS use cases.

Considering standalone logging methods, the current research in this field is primarily targeted at general distributed systems. Mansor et al. [76] propose a secure approach to logging unified vehicle ECU data in support of modern digital forensics, i.e., black box. The approach involves the inclusion of a HSM in each ECU as proposed by the EVITA project [164], with data being monitored and controlled via a mobile and cloud application called DiaLOG. An alternative approach to modern logging methods was recently presented in G.R. Hartung's dissertation [75]. The dissertation focuses on providing a solution to cryptographic log entries for computer systems by providing a method for maintaining integrity and authenticity security against log aborts that is capable of securely verifying only excerpts from log files. A similar scheme may be considered for the BMS in the future, as there are security and log truncations requirements are present in these systems as well. Another approach to log aggregation that focuses on embedded devices is presented by Camara et al. [165]. It is a MAC-based schema that uses symmetric cryptographic primitives and the FssAgg authentication scheme [166].

Contribution. A work specifically targeting the storage of BMS log data was presented by Zhou et al. [55]. It argues for the importance of data storage of BMS log data, especially in the modern era where appropriate methods must be provided for efficient handling even on the vehicle side in the field. The paper shows a compression and storage approach based on the modular distribution of ECU in a vehicle. It uses frequency division as the basis for compression. Compared to the work presented in this dissertation, Zhou et al. only consider data acquisition at the lower physical layer. We extend this concept by also considering the pre- and post-processing of BMS data as well as the security aspects both on the on-premise and in the online cloud.

3.3.1 BMS and Blockchains

Blockchains are a concept still often used for data logging and storage. Their distributed architecture and integrity guarantee make them attractive for general data accusation and management of larger systems [167]. An example of such a system would be BCALS, a secure log management system based on the use of blockchains and distributed cloud systems [168]. Blockchains have also seen interest with BMS [169, 170, 171, 172, 173, 174]. Kim et al. [171] laid a foundation for the implementation of blockchain technologies with the BMS to provide an additional layer of security against cyber-physical attacks. The system is further extended by Aenugu et al. [174] with a data management and analytic platform implemented in AWS cloud and which shows a general framework use that can be adapted between cloud BMS services and blockchain ledgers, and by Justin J. Ochoa et al. [172] presenting its application as a battery energy storage system. Bere et al. [173] further explore firmware handling from the security and update perspective and how it can be used with a BMS-affiliated blockchain system. The use of the blockchain with BMS technology would likely be based on a lightweight blockchain implementation, such as the Hyperledger-Fabric, which is open-source and adaptable for testing [167, 175]. The use of the Hyperledger-fabric can also be adapted for an IoT BMS approach, by controlling the network flow between data that needs to be stored on the blockchain ledger and those that do not, as well as keeping the synchronization between the nodes [170]. Florea et al. [169] present a different approach using Ethereum and the directed acyclic graph. In their paper, they aim to use the blockchain ledger solution as the base of a system architecture that enables easier swapping and regulating of battery packs between electric vehicles, supplementing the batteries' second life idea. Similarly, blockchain technology has also been suggested for a larger use case that considers the tracking of electric vehicles charging use and couples it with other grid-powered systems [176].

Other related work with blockchains. In the light of using blockchain with BMS, several other principles could be applied from related work concerning blockchains with embedded devices. An earlier work by Christidis and Devetsikioti [177] present how the use of blockchain can be applied in relation to IoT market services, also considering the cryptographic properties. While largely focused on the traditional use of blockchains and their business models, the core concept of applying blockchains in IoT solutions could also be used to extend the BMS main functionalities beyond what is currently presented in the BMS and related blockchain SotA [169, 171, 172, 173, 176]. In the paper published by Chiu et al. [175], a model is presented that gives accountability property to the embedded systems through a permission-based blockchain framework. Such a model could also be adapted to be used with modern BMS to bring an additional layer of detection and identification of faulty battery cell modules.

3.4 Automotive Security

BMS are nowadays mainly researched under the automotive use case. Since they play an important role in the automotive domain, it is necessary to also study the SotA security architectures used for in-vehicle communication. We can observe automotive security at different levels, either focusing on in-vehicle security or on external communication with other vehicles, infrastructures, and cloud services, with many security standards and mechanisms in circulation [178]. Security solutions for vehicles that focus on the use of implicit certificates are discussed separately in Section 3.5.

3.4.1 Security for the CAN protocol

The CAN protocol and its more recent variants, e.g., CAN-FD or CAN-XL, are one of the most widely used and oldest protocols for in-vehicle communication. It is a message-based protocol and its simplicity allows flexible use and extensions. However, the original protocol was not designed with a security layer in mind. In recent years, many papers have been published that try to close this gap and provide different security solutions for this old but still very usable protocol [179, 54].

In one of the earlier works, Hazem & Mahmoud Fahmy [180] present a lightweight protocol that integrates security with authentication attribute to the traditional CAN protocol. In its base form, the CAN protocol does not have a security layer and the data transmitted with it must be protected with additional mechanisms. Similarly, Lin et al. [181] also show an earlier experimental security model for CAN that considers communication latency and overhead. While the solution focuses on providing elementary security as well as security against replay and masquerade attacks, its maintenance would be difficult due to the number of pairwise key components. For solutions that consider not only CAN secure channel, Han et al. [54] propose a defence mechanism against DoS attacks in addition to a three-step authentication protocol.

3.4.2 Vehicle networks

For in-vehicle communications, a relatively recent paper by Fuchs et al. [53] proposes a design that relies on the use of a TPM for authentication and secure communication between a BMS sub-system and the EVCC. If we observe the vehicular network further, we arrive at the level of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, usually abbreviated simply as V2X for the combined solutions. At this level, Eric R. Verheul [182] proposes Issue First Activate Later (IFAL), a PKI based on the use of short-lived pseudo-anonymous certificates. IFAL aims to enable flexible policy through code provisioning and control that can also detect and stop services for misbehaving vehicles.

Mundhenk et al. [183] go a step further and also consider the connection of the on-board network to the Internet. The main architecture is based on a central ECU device, and the design includes both device authentication and secure session establishment processes for the ECU devices. The solution combines both symmetric cryptography for secure session derivation and asymmetric cryptography for ECU authentication. Although the authors present an innovative and secure design, it is hard to call it "lightweight", since the main form of authentication relies on the RSA algorithm with relatively long keys and a high computation time measured in seconds, even if it is implemented in the software only for demonstration purposes. The authors themselves mention this point and argue that the authentication approach should be used only once, when a change is detected with the ECU. Moving away from a centralized key management approach, Roca et al. [184] present a semi-centralized framework for dynamic key distribution. The resulting framework is based on identity-based encryption for the public key generation with a special variant that enables broadcast encryption and messaging, hence the title "semi-centralized". The results presented in the paper seem acceptable, but the usability of the framework in a real vehicle system is questionable. The main advantages of the presented framework, i.e., authentication and key derivation in case a ECU is compromised or the gateway is offline, would hardly come into play due to the other safety, security, and also functional vehicle requirements.

As new threats are gradually discovered with new requirements that must be met, it is also necessary to provide a secure mechanism and channel for executing automotive updates. In this scenario, it is necessary to search for secure but also efficient update models for vehicles. Steger et al. [185, 186] present a body of research focused on providing a lightweight and secure update process for configuration and software updates on ECU. In their work, a generic framework is proposed to support the requirements of different vehicle network topologies.

3.4.3 Research projects

Several national and global research projects have already also taken interest in the security of the automotive domain, usually linking BMS as one of the most important factors of protection. EVITA is one of the earlier EU large-scale projects aimed at providing security solutions for vehicle ECU [164]. They provide a general threat overview with recommended countermeasures, which strongly suggests the use of HSM. HEAVENS is a recent project that focused on identifying important security vulnerabilities in modern vehicles and proposing tools for their analysis, testing and verification [187]. A successor model was also proposed by Lautenbach et al. [188] in their own study. INCOBAT is a similar project with a stronger focus on providing cost-effective BMS for the next generation high voltage battery packs for EV [189]. Running until 2021, the EVERLASTING project focused on a set of norms and standards that could be applicable to BMS in the area of battery lifecycle, reliability and safety [11].

3.5 Security Models with Implicit Certificates

Most modern systems that rely on a PKI infrastructure and the use of certificates for authentication rely on a traditional explicit architecture and TLS or similar protocols as the underlying security layer. However, in recent years, several studies have tried to find alternatives to this, especially to find more accessible and lightweight solutions for constrained and embedded devices. One of the proposals was the use of implicit certificates, a security concept described in Section 2.3.3. The use of implicit certificates has been proposed in several areas but was mainly aimed at applications for in-vehicle and local wireless networks since these applications would benefit most from the reduced protocol overhead. On the other hand, several works have also tried to find a more general solution that could be extended to large networks, such as the work by Huang et al. [190] that focuses on the concept of certificate transparency and how it can be used together with implicit certificate schemes. Most of the research on implicit certificates is based on the use of the ECQV scheme as the most optimal and secure implicit certificate scheme currently available.

3.5.1 Implicit certificates with in-vehicle networks

Puellen et al.[191] propose the use of implicit certificates for on-board vehicle networks. They present a solution divided into three main phases: (i) device authentication, (ii) derivation and exchange of implicit certificates, and (iii) derivation of session keys for the ECU. However, their main contribution lies in the session key derivation proposal that takes into account the general network architecture of the vehicle CAN. The key computation protocol is based on the use of the traditional Elliptic Curve

Diffie-Helmann (ECDH) algorithm, after which the output is further processed with an one-time password (OTP). The first phase of device authentication relies on the use of physical unclonable function (PUF), which is efficient but raises some concerns as it requires ECU boards supporting the same functionality, requiring each board to have its own challenge/response table, which increases overhead. In addition, PUF are still not deemed to be fully reliable for use in real-world applications, as they have several security weaknesses associated with them [192, 193, 194].

Vehicle security architectures are often explored in the context of a broader service area that also includes charging infrastructure. Extending solutions that target the network relationship between an in-vehicle network and a charging station, Almuhaideb and Algothami [195] propose a security scheme that relies on the ECQV. The novelty of the work lies in the optimization of the re-authentication process, which drastically reduces the time required.

In further research on external vehicle services and V2X communications, Pollicino et al. [130] apply the implicit certificates of ECQV to ad hoc vehicle networks. They contribute by analyzing the protocol for a real-world environment considering the IEEE 1609.2 standard and evaluating the performance of the ECQV certificate derivation and the sequential ECDSA verification. The authors deploy different hardware based on their performance results and present their findings. They conclude that while even severely constrained devices can support the protocol, special considerations must be made when involving strict vehicle timing requirements.

While the use of implicit certificates in a vehicular environment offers several advantages, Eric R. Verheul points out potential concerns with their deployment [182]. In particular, the possibility of deniability of issued certificates, where a party could claim that even though a certificate is issued in their name, they did not initiate any action with them. This is the problem of non-repudiation, i.e., if two or more valid users who are able to derive certificates from a CA conspire with each other on sharing the same unique identifier on which basis the public keys are derived. In cases where it is necessary to associate an action with a user, it would be difficult to prove which user it was, since public key derivations are tied to their identifiers. In the design presented in this dissertation, this is partially solved implicitly by using the authentication phase before the certificate derivation phase to ensure that only pre-registered devices have access to the gateway, i.e., the CA. This is maintained by using a pre-embedded secret, e.g. a master key, or a PUF.

3.5.2 Implicit certificates with IoT and wireless networks

For use in wireless networks, Porambage et al. present one of the earlier architectures for authentication and key exchange based on the ECQV scheme [196, 197]. However, their design leaves several open points of discussion. First, the key exchange is still based on the SKD scheme, which may be a problem in the case of a node-capturing attack. Second, authentication is based solely on the security strength of the symmetric and distributed key *'K'*. These concerns were taken into account when designing the security architecture presented in this dissertation.

The use of implicit certificates was also observed in 802.15.4 networks. Park [132] presents a design with ECQV that builds on the local network characteristics allowing for extended security handling during the certificate exchange process. Similarly, Dini and Tiloca [198] showed a general security solution for the ZigBee networks. Interestingly, in this relatively early work, perfect forward secrecy is provided, but only through revocation and control by the central device rather than through a method involving the node devices. Although this work was quite novel for its time, its potential seems never to have been fully exploited and is currently superseded by many modern security solutions concerning the same networks. Siddhartha et al. [199] present an alternative solution for using implicit certificates in an IoT environment, where a specially derived *"authenticator"* can be used for mutual authentication between devices. The authenticator is generated from certificate-related data and signed by the CA that guarantees its authenticity. This authenticator token is transmitted to the devices during the certificate generation. Its integrity is proven using a hash function. While lightweight, the protocol is potentially vulnerable to MitM and replay attacks, since an attacker can simply re-transmit its own value using a false identity of the valid node. Another problem is that, as with many other SotA implicit certificate solutions, the derivation of the session key is still based on a traditional SKD rather than a DKD.

In the paper presented by Sciancalepore et al. [200] a solution is proposed for a relatively secure and lightweight ECQV key management protocol. Similar to the previously mentioned papers, the design is still a SKD rather than a DKD and it does not possess additional security features such as perfect forward secrecy. However, the design allows for very fast authentication and key derivation because it does not rely on the use of Elliptic Curve Computation (ECC) and public key infrastructure, i.e., it does not require signature derivation and verification during the authentication phase. However, this symmetric authentication mechanism, along with the static derivation of the premaster secret for the key, leaves open doors for potential vulnerabilities that come with using such approaches.

Similar to the solution presented by Sciancalepore et al., D. Lee and I. Lee [201] focus on a key derivation approach. They present two methods (i) a simple and lightweight approach based on pure ECQV using a SKD with random nonce and previously generated private and public keys, and (ii) a more complex approach that actually provides ephemeral keys through a DKD protocol based on the Schnorr algorithm. While both methods are novel, they lack a crucial step, namely device authentication. The authors consider authentication mainly through public key computation and validation with no signature or symmetric key verifications.

As an extension to existing protocols, Duy An Ha et al. [131] present a design based on the use of the ECQV with the Datagram Transport Layer Security (DTLS) protocol. The intended use case is mutual authentication and key derivation for IoT devices. Although it provides an interesting concept, its evaluation is flawed as the presented ECQV protocol is compared against the ECDSA scheme associated with the explicit certificate solution. In fact, the key derivation process using the ECDH would still rely on the ECDSA for authentication, since the algorithm currently presented in the paper only derives the public keys, but does not verify if these public keys actually come from valid devices or if they have been correctly computed.

As one of the more recent works on implicit certificates, Zi-Yuan Liu et al. [202] extend the use of ECQV for use cases that can benefit from better management in terms of storing and handling a large number of certificates and keys and are intended for dynamic networks that change frequently in their setup and require frequent key updates.

Contribution. As can be observed from the analysis, most of the research on implicit certificates is based on the use of the ECQV scheme, which is currently the most optimal and best-understood implicit certificate scheme. Although similar, each design offers a different approach to a general concept. However, most research proposals only consider the SKD for the secure sessions and disregard the ever-important attribute of perfect forward secrecy. In this dissertation, we address the drawbacks found in proposals for both wireless networks and vehicular applications and present a general BMS security architecture based on the use of implicit certificates that focuses on current challenges.

3.6 NFC Security Models

NFC technology is deployed in environments that benefit from its short range, making many attacks difficult, or even impossible, to execute. Nevertheless, as discussed in the previous Section 2.4, the traditional NFC deployment leaves the channel and data open. Traditionally, Record Type Definition

(RTD) is used to provide an additional layer of protection for the data integrity and authenticity of NDEF messages. There are several versions, with version 2.0 being more commonly used, as the original version 1.0 has been proven vulnerable to certain types of attacks [151, 203]. RTD does not provide data confidentiality, but only integrity and authenticity, and therefore requires additional security protocols to process the encryption. It also relies on asymmetric cryptography, i.e., signatures and certificate chaining, which may prove too demanding for constrained devices that would otherwise employ NFC technology, as well as for the BMS sensor and external readout use cases presented in this work.

Ulz et al. propose several different NFC security infrastructure solutions for updates and configurations [123, 204]. Most notably, they propose QSNFC, a complete NFC security suite [205]. QSNFC enables secure session establishment through certificate authentication and key derivation using Diffie-Hellman key exchange. However, it relies only on 128-bit public keys, which is less than the current recommendation from National Institute of Standards and Technology (NIST), i.e., less than 256 bits. Although this drawback could be easily addressed, this solution is still not fully suitable for the targeted BMS use cases, as the suite is only intended to verify QSNFC server authentication, but not client authentication. This leaves open the possibility of unverified reader updates to BMS controllers. Similarly, Urien and Piramuthu[206] present the adaptation of the traditional TLS protocol in the context of NFC by also relying on asymmetric cryptography. While this approach would provide a stable and secure architecture, it would still be very resource intensive and impractical for use with modern BMS.

Regarding the use of implicit certificates and the NFC technology, Christian Lesjak proposes a security design that employs the ECQV scheme [207]. While using the ECQV scheme would give NFC an advantage in terms of the size of the certificates and their verification, as they are easier to update and process in smaller networks, the design still has the general disadvantages of a full asymmetric cryptography architecture, i.e., it being resource and time-consuming.

RFID and also NFC tags have always been considered cost-effective technology for tracking goods and products. Therefore, an important element to consider is the protection of the security authenticity attribute. NFC tags can be used as a security measure on products to verify the authenticity of the device. Fake devices are regarded as *counterfeited devices*, with the technology and methods that enable their detection being aptly called anti-counterfeiting schemes. Several publications have been written on this topic, with research addressing either specific product authentication, such as wine or retail products [208, 209, 210, 211], or complete infrastructures developed for IoT environments [212, 213]. These solutions could also be extended and used for battery pack authentication via NFC.

Contribution. While the presented research solutions provide interesting architectural NFC concepts, they either lack certain security features that would be required to fully protect the targeted BMS readout use cases or are otherwise too performance-demanding. We extend the notion of NFC security protocol design by proposing a lightweight security model based solely on symmetric cryptography that can be employed with BMS and similar automotive and industrial applications.

CHAPTER 4

Wireless and Secure BMS Connectivity using NFC

Summary: *This chapter contains the first part of the design. The main focus of the dissertation is placed on the BMS security. However, it would be difficult to talk about internal BMS security without considering the proposed design of wireless NFC connectivity. Therefore, this chapter combines both wireless and secure BMS and discusses NFC design concepts. The design is proposed at the system level and includes wireless requirements analysis, wireless system design, and security design model.*

◇◇◇

4.1 System Overview

BMS modules communicate both internally and externally, transmitting important diagnostic information. Traditionally, BMS rely on wired solutions to provide these services. The wired design limits the scalability of the BMS design and applications, greatly increasing maintenance complexity and production cost [22, 21]. Modern use cases also require easier tracking of battery lifecycle data, i.e., diagnostic data that tells us more about the current and previous use of battery packs. Relying on wired communication would be cumbersome and impractical. Tracking battery life cycle information is important, both for in-situ measurements performed directly onboard a vehicle and for tracking data over a certain period of time, supplementing battery packs' use for second-life applications [60] and battery passports [15].

Using different wireless technologies on one system can not only be impractical, but also costly and introduce additional functional challenges, such as radio interference that can affect the reliability of data transmission. One of the goals presented in this dissertation is to avoid these problems by focusing on unified wireless solutions. We achieve this by relying on the use of RFID, in particular the accessible and low-cost NFC technology. In addition, a system design analysis is conducted to complement any new vulnerabilities and exposures that arise from the use of this wireless approach. This is achieved by first performing a high-level security requirements analysis, followed by the presentation of appropriate security design solutions. Wireless sensor readout security also takes into account the discussed internal BMS security on the battery pack module. The system design was done in parallel with the supervised works of Martin Gärtner on internal, and of Claudia Laube on external, BMS NFC readouts [214, 215]. In this dissertation, I further contribute by realizing the system requirements, unifying the system design, standardizing the security design, and extending the HW/SW design concepts discussed in this chapter.

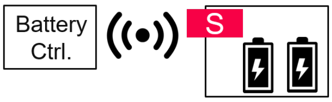
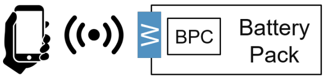
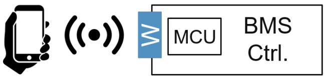
4.1.1 Use cases

The proposed system design should be adjusted to the main three use cases described in Table 4.1. The *active sensor* use case considers the active readout of sensors during an active use inside a system, e.g., during the standard vehicle driving period and being readout by the BMS [149, 216]. For this use case, the system relies on the reader/writer NFC mode. This means that the BPC needs to possess an NFC reader interface as the active component. On the battery cell side, the sensors need to be connected with a passive NFC tag interface, e.g., NTAG [1]. The reader will initiate the communication with the passive element, but it is up to the tag to sample and send sensor data.

The two diagnostic use cases are intended for the status readout of the BMS lifecycle [144]. Primarily, they are used for the battery status readout. In both cases, it is important to detect abnormal behavior of battery cells, since the changes in temperature and storage environments can affect battery cells' life [217]. In the *active diagnostic* scenario, we propose that this connection goes over the central BMS controller through an NTAG interface. The external reader can be a mobile phone or a similar portable active device that works in the reader/writer mode.

The *idle diagnostic* use case is intended for scenarios where battery packs are detached from their actively deployed system and, for example, stored in warehouses. This would happen during the transfer for the second life use, as described in Section 2.2. This line of communication would help in the status readout, but also with firmware and configuration updates [123]. The communication would go over an external reader with the passive NTAG attached to the battery pack's BPC. The interaction of the application layer is very similar to the active diagnostic use case, but the way the NFC communication works in terms of the wake-up and data-handling procedure would be different.

Table 4.1: BMS wireless use case scenarios for the deployment of the NFC readout.

Use case	Deployment	Comm. path	Readout data	
Active sensor	in-system; internal	Batt. pack → BPC	sensor data	
Idle diagnostic	stored; external	BPC → Ext. reader	status & sensor data	
Active diagnostic	in-system; external	BMS ↔ Ext. reader	status & diagnostic	

4.1.2 Security requirements

Most modern vehicles still rely on CAN, ethernet, and FlexRay for communication between ECU. Traditionally, these are unsecured or contain only a rudimentary security layer. They are primarily isolated from the outside world, apart from a few connections, e.g. OBD2. However, a wireless interface could still be vulnerable to remote attacks. There have already been cases where the Bluetooth stack was

[1] For simplicity purposes we will be referring to the passive NFC component as NTAG, but we do acknowledge that other type of tags can be employed as well

vulnerable to format string attacks, where in the worst case an attacker could even gain access to the steering wheel control [218, 219]. An unsecured entry point, i.e., an unsecured BMS, can result in exposing several other potential attack points. A remote manipulation was successfully demonstrated by Rouf et al. [220] by reverse-engineering a tire pressure monitoring system that operated at an ultra-high frequency (315 MHz / 433 MHz). In addition to manipulation, eavesdropping was also possible from a distance of up to 40 m. Another important factor in maintaining the security of a vehicle system is in maintaining its regular updates, be it keys, certificates, ciphers, etc. Today, there is a debate about how old vehicles should be updated in the future. This is very important when new security vulnerabilities are found. All these points show that modern vehicles, even with heterogeneous concepts, still suffer from the traditional security threats associated with unsecured wireless interfaces.

NFC is no exception when it comes to remote attacks compared to other wireless technologies. The short communication range and reserved frequency band provide an advantage by reducing the potential attack surface. However, there are still some pitfalls that need to be taken into account [151, 154]. Here, we also consider the security threat models investigated under BMS and try to adapt the solutions to be compliant with the general system concepts [41, 48, 29]. The security requirements of BMS are analyzed in the context of the research use cases of Section 4.1.1 and observed in the context of security for (i) battery pack internal security, and (ii) diagnostic readout external security.

Battery pack security

The battery pack consists of battery cells, the adjacent interface, and sensor components, as well as the BPC placed inside or outside the pack [221]. The battery cells and sensors are typically enclosed in a tight metal housing [222, 223]. We assume that this enclosure is sufficient to prevent it from being compromised during communication, i.e., it renders remote attacks impassable. Since the communication interface between the sensors and the BPC is to be run over NFC, the short-range would be sufficient to protect the device from serious remote attacks and would suffice the security requirements for communication. However, this feature alone is not enough. We still want to ensure that the devices communicating with each other are trustworthy. Therefore, each interface of the battery pack must be authenticated before it can proceed with the standard BMS monitoring processes.

A battery pack is in itself a very simple device. It does not provide any nominal form of security that is pre-installed. Based on the preceding analysis and discussion, the following two main requirements for battery pack security must be met:

1. *Required: device authentication.* To be able to be used efficiently and safely, battery packs need to come from verified vendors. While malicious attacks through fake battery pack modules could also be possible, it is more likely that we would be dealing with counterfeited battery packs, which could also pose an indirect threat to the users as cheap and untested replacements [51].
2. *Optional: secure channel.* The battery cells and interfaces are in a tightly enclosed metal chassis of a shielded car area, where it would be very difficult to otherwise probe or modify them [224, 223]. The channels could be encrypted, but adding security channels would place a heavy load on the communication line and affect the sampling transmission rate of critical battery data.

External diagnostic security

The external diagnostic security adheres to the Idle and Active diagnostic use cases from Section 4.1.1. Since the readout takes place from the outside using an external reader, such as a mobile phone, the wireless channel would be more susceptible to remote attacks compared to the enclosed battery pack discussed earlier. Here, communication security must meet both the common security threats associated with BMS [29] and those associated with wireless communication [151].

Since the readout can be done with a portable external reader, mutual authentication is necessary to ensure that both sides are valid devices before data exchange can take place. In both of these use cases, authentication is not sufficient, as attacks can also occur remotely in the form of eavesdropping, i.e., sniffing attacks, in the range of up to 10 m, as shown by Haselsteiner and Breitfuss [153]. Alternatively, an attacker could try to target an authentication identity that the vendors cannot secure in the reader/writer mode and which requires additional verification [152]. Therefore, the channels must be encrypted. Other forms of attack could take the form of MitM attacks on the channel if it is left unprotected, replay attacks, DoS, etc. These attacks would ultimately target the BMS or BPC hardware and software integrity [48, 30, 49].

A protocol needs to be developed that will be able to answer the mentioned threats, but also fulfil system functional needs. Summarized, we derive the following requirements for the external readout:

1. *Mutual authentication.* Both the external reader and the NTAG of the BMS or BPC controller need to be able to authenticate each other.
2. *Employment of a secure channel.* The communication that follows the authentication needs to provide data confidentiality, integrity, and authenticity, i.e., to be encrypted and tamper-proof.
3. *Light system model.* Implemented security system design needs to be lightweight in its implementation with minimal performance, storage, and maintenance overhead.

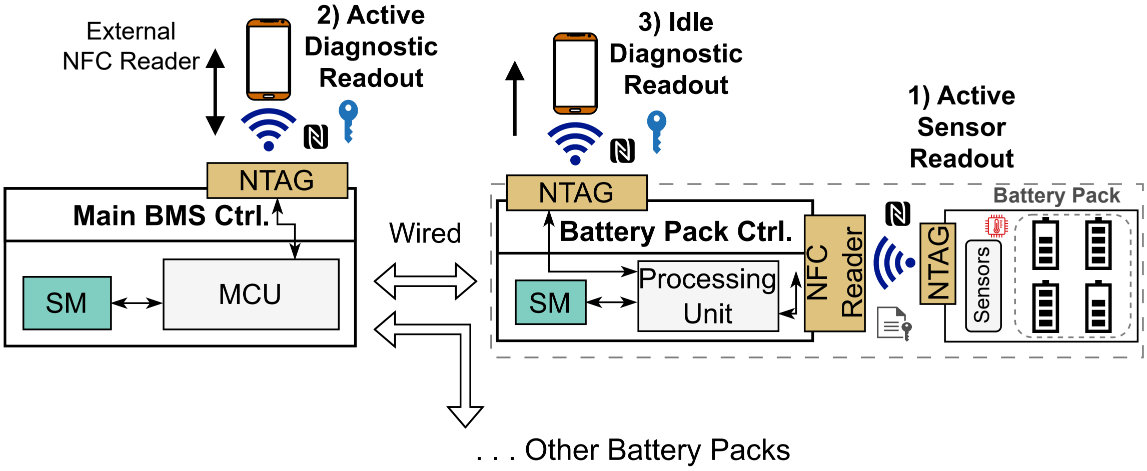


Figure 4.1: BMS NFC system architecture: it shows the main building blocks regarding the three use cases and which part of the BMS sub-system they incorporate under the main design. Each battery pack inside the architecture contains the same system design layout. To make the design cost-effective, it only contains one active NFC reader component on the BPC, with the rest being passive NFC devices. Adapted from *Publication G*.

4.2 System Design

To establish a secure wireless communication that is able to fulfil the functional and security requirements for the use cases established in the previous section, it is necessary to realize a fully usable system design. Figure 4.1 shows our proposed system design for BMS NFC readout that accommodates both external diagnostic and internal battery cell sensor readout use cases.

The system design distinguishes between two main building blocks:

- **Main BMS controller.** Considers the central BMS controller that handles the connection and control of the assigned battery packs. The primary use of the BMS controller for the wireless readout would be in the hypothetical active diagnostics use case. Here, it would be possible to read out the necessary diagnostics data of the potentially complete BMS sub-system through the added interface. In terms of additional components, we see the following:
    1. *BMS MCU*: extending the MCU to support the necessary diagnostic readout functionality.
    2. *NFC passive device*: an NTAG for the purpose of the readout and message relaying between the BMS MCU and the external reader.
    3. *Security module*: for security functions intended for the external readout, but also in the wider spectrum; intended for the complete security encapsulation of the BMS controller.
    4. *Mobile reader*: a mobile phone or an otherwise NFC-equipped portable reader that can be used for the active diagnostics use case.
- **Battery pack.** Consists primarily of the BPC and battery cells with associated sensors and interfaces. For the wireless sensor transfer, a dedicated tag needs to be used that is connected to sensors, as well as a security model, either embedded or attached in addition to the NFC tag. The BPC includes both an internal NFC reader for sensor readout and an NTAG for external readout when considering the idle diagnostic readout use case. Typical derivations of BPC do not take these additional components into account, and therefore the current controllers would either need to be upgraded or replaced with a dedicated MCU that is able to, performance-wise, handle the additional software overhead. In summary, we observe the following additional components:
    1. *BPC MCU*: extended or additional processor for handling the control of the NFC reader and the process extension for the external NTAG communication.
    2. *NFC active and passive devices*: One active reader and one passive tag for the BPC, and one NTAG per battery cell module.
    3. *Security module*: for providing dedicated security functions and for secure storage of important key, diagnostic, or sensor data.
    4. *Mobile reader*: same use as with the previously described BMS controller, intended for the use with the idle diagnostic use case.

All scenarios utilize NFC as the communication channel, with both the sender and receiver adhering to the reader/writer mode. In the following sections, each main use case is discussed in detail, including its specific communication protocol steps. Although the system design depicted in Figure 4.1 mainly aims to address solutions for the distrusted BMS topology, which is the most commonly used, it is adaptable as well to other topologies mentioned in Section 2.1:

> **Centralized.** Both the functions of the BMS controller and the BPC are considered under one unit. Thus, it consists of only one NFC reader and one NTAG. The communicating sensors still require their own NTAG. A question is posed in relation to how many passive tags would be ideal compared to the number of active readers the controller must have. Right now, we observe only one active reader and a limited number of passive tags coming from the battery cell sensors. The idle diagnostic use case would only be feasible if the main controller would be stored for the second life use in addition to the battery cells.
>
> **Distributed.** It follows the design described and contained in Figure 4.1. Each BPC would contain at least one NFC reader and one NTAG, with at least one NTAG coming from the battery cell sensor side. The main BMS controller would also need to possess an NTAG to support the active

diagnostic use case. Since there is a clear separation of duties between the individual use cases, this topology is the most appropriate with regard to possible expansions or modifications.
Modulated. The topology is similar to the distributed one but without a distinct main controller for the BMS. Therefore, every control module must have an NFC reader and an NTAG to facilitate the reading of internal battery cell sensors.
Decentralized. In a decentralized BMS topology, multiple BMS sub-systems work together under one overarching system, but each has its own individual control over its sub-system. This means that the deployment process is similar to the ones described in the previous three topologies. The number of interfaces increases proportionally with the number of underlying BMS sub-systems.

Interfaces. During the production phase, it is essential to provide the current BMS sub-systems with the appropriate interfaces to support extra communication points. This is crucial as it allows the primary BMS MCU to connect with the NTAG and dedicated security module, battery cell sensors, and their respective NTAG. Furthermore, interfaces must be available for the BPC, for both the NFC reader and an NTAG. These interfaces should adhere to standard serial communication protocols such as Inter-Integrated Circuit (I2C) or Serial Peripheral Interface (SPI) links.

It is unclear how many readers are possible compared to the number of passive tags. We generally assume one reader per tag. However, some installations may allow up to four tags per battery pack layout. To fully understand the system, we need to investigate impedance matching, antenna design and orientation, and placement in a metal environment. These aspects were not the focus of this dissertation and require dedicated hardware analysis in future research. Instead, we focused on the system design, including the dedicated communication protocol and data structures, as well as added security measures. In the following sections, we will discuss each readout process individually.

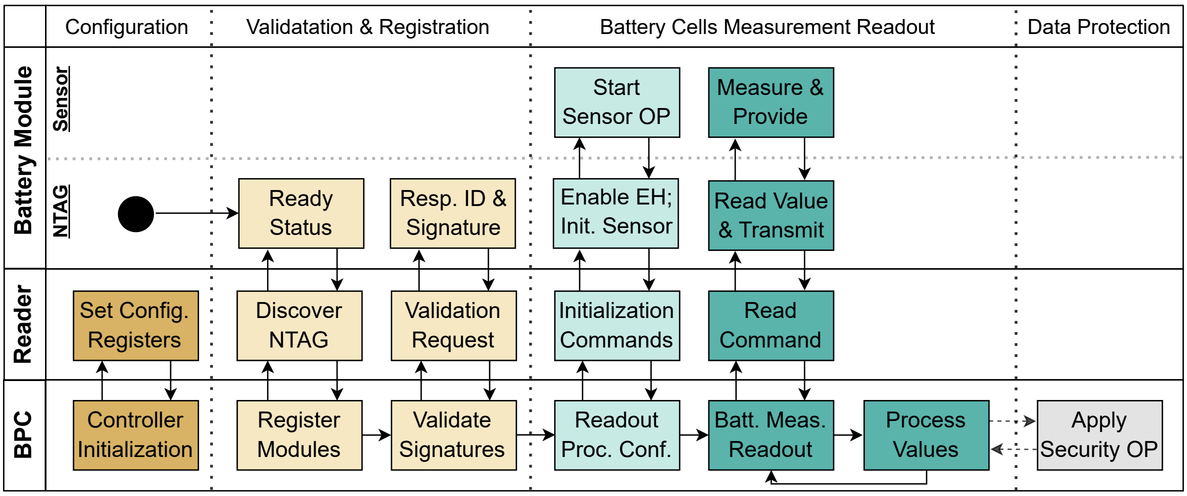


Figure 4.2: BMS NFC sensor readout swimlane diagram. It shows the communication steps between the BPC and its reader, together with the battery module and its NTAG. The devices are first pre-configured during the configuration step, after which the communication starts with first validating and registering the NTAG onto the BPC reader side and moving on to the continuous sensor readout. The configuration step is ideally only done once per deployment. The validation and registration depend on the system design requirements, but could possibly also be optimized to be executed only once during the first interaction between the BPC and the battery module. The following data protection step considers the security operations on the read data and is observed as an optional step that is included depending on the targeted requirements. Adapted from *Publication C*.

4.2.1 NFC sensor readout

To obtain sensor data, the reader/writer mode of NFC is exclusively used. The NFC reader, which is the active device, initiates and begins communication with an NTAG, which is a passive tag. Before starting, the devices must execute configuration and security operations. Once configured, continuous communication can proceed. The reader can theoretically read data from NFC sensors as soon as previous data has been read, i.e., as long as new data is present in memory.

Under configuration we consider the configuration of the devices themselves, and also the discovery loop, a necessary step to detect and register the embedded NTAG. Since NFC devices are designed to be stationary, this step can be optimized to consider only the detection of already registered devices. The devices should be registered when they are newly installed in the system. Regardless of registration, each connection and device must be authenticated each time the system is rebooted during the specified security verification period. This is done because a malicious attacker could copy the same NFC tag attributes but replace them with a malicious device. Modifying and replicating security hardware components is difficult and very costly for an attacker to exploit. Authentication for the sensor readout use case is described in more detail in Section 4.3.1.

Data exchange protocol. The sensors are either integrated directly into the NTAG or installed separately and then connected to the NTAG interface, for example, via an I2C connection. In both cases, the NTAG takes the role of a *adapter module*. On the other hand, the BPC reads the data from the NFC reader in pass-through mode and relies on the static random access memory (SRAM) buffer. The commands sent through the NFC reader include I2C commands that are executed on the NTAG. The commands are mainly aimed at interacting with the sensors for the purpose of reading and transferring their values. The system design only needs to provide the necessary extension option for configuration on the BPC and interfaces for communication with the NFC devices. This makes the design relatively cost-effective and straightforward to produce since no additional components are required.

The complete data exchange protocol is illustrated in Figure 4.2 using a swimlane diagram. It consists of two primary components: (i) a battery module that comprises battery cells and a passive NTAG, and (ii) a BPC with an NFC reader. The communication protocol comprises four main steps:

1. *Configuration:* intended to be run at the start of the system and only once. Certain configuration options can be pre-configured or pre-cached to make the subsequent initialization faster.
2. *Validation and registration:* the BPC instructs the NFC reader to discover and authenticate any registered or otherwise NTAG present in the proximity of the field.
3. *Battery cell measurement readout:* the main process loop that includes sensor value registration, transmission to the NTAG, and subsequent NTAG transmission to the BPC.
4. *Data protection:* an optional step that includes any additional post-processing security operations on the data. Based on our security requirements analysis, this step is not necessary for the encapsulated battery packs, and thus, we consider it an open and optional process.

4.2.2 NFC external diagnostic readout

For the external readout, we focus on the mobile reader as the central device for this communication approach, e.g., an NFC-enabled mobile phone. The mobile reader must be properly configured to be able to communicate with the dedicated NTAG on top of the BMS hardware, i.e., it must have the correct software with also correct security configuration. For both the active and idle diagnostic use cases, the passive tag acts only as a bridge for carrying over the data. The data is temporarily stored, usually in SRAM before being read by the BMS controller. This occurs because the secure embedded hardware

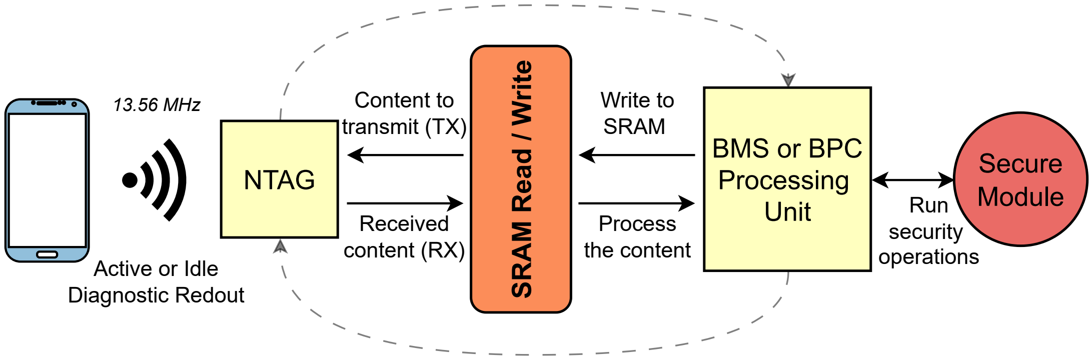


Figure 4.3: External diagnostic readout principle. It accounts for both the active and idle diagnostic readout with SRAM as the buffer between the processing unit and the NTAG. The mobile device is the active NFC device that starts the communication as follows: Wake-up → Authentication → Session communication.

is likely to be found outside the interface connection, as the data must be properly decrypted and decoded before it can be further used by the BMS controller and vice versa on the mobile device. This communication principle is shown in Figure 4.3. Since the handshake protocols for security authentication and the subsequent establishment of the secure session channel play an important role in this communication, they are discussed separately in Section 4.3.2.

When it comes to the diagnostic readout, the devices, and protocols for the external communication for both use cases remain the same. However, there are two key differences in the system design that must be considered between the idle and active diagnostic use cases:

- *Application layer.* The readout application needs to account for different data decoding and processing on the application side, specifically from the external reader. But these modifications are marginal and the application can be set to have the same process runs with both use cases.
- *Wake-up procedure.* To conserve energy, the wake-up process would only be used for idle diagnostic purposes. Since the battery packs are stored before their second life use, the readout should occur only on specific occasions to minimize power usage and extend the batteries' lifecycle. Various methods for achieving this are discussed in Section 4.2.4.

4.2.3 Data exchange structure

The BMS readout process is defined as a service that enables the readout of BMS status data. It runs on top of software layers that enable this interaction. A software layer stack was developed for the wireless NFC readout use cases. The stack is shown in Figure 4.4 and consists of several layers. At the very bottom resides the *hardware abstraction layer (HAL)*, which is vendor-specific and enables control of the HW/SW drivers. In our design, these are primarily communication interfaces for interaction with NFC readers and tags, general-purpose input/output (GPIO) control, event interrupts, clocks, timers, etc. On top of this is the *NFC management layer*, which is responsible for NFC wireless data transmission and NDEF message processing. In order for NFC to transmit its data securely, a dedicated Secure NFC Data Exchange Format (SNDEF) data structure is used, which resides at the *Security layer*. The security layer provides functions for device authentication and key management, among others. It has a dedicated application programming interface (API) to control the security operations between the software execution and the security hardware modules. The transfer between raw values and diagnostic data and their processing in a special application data structure is performed on the *Application layer*. The

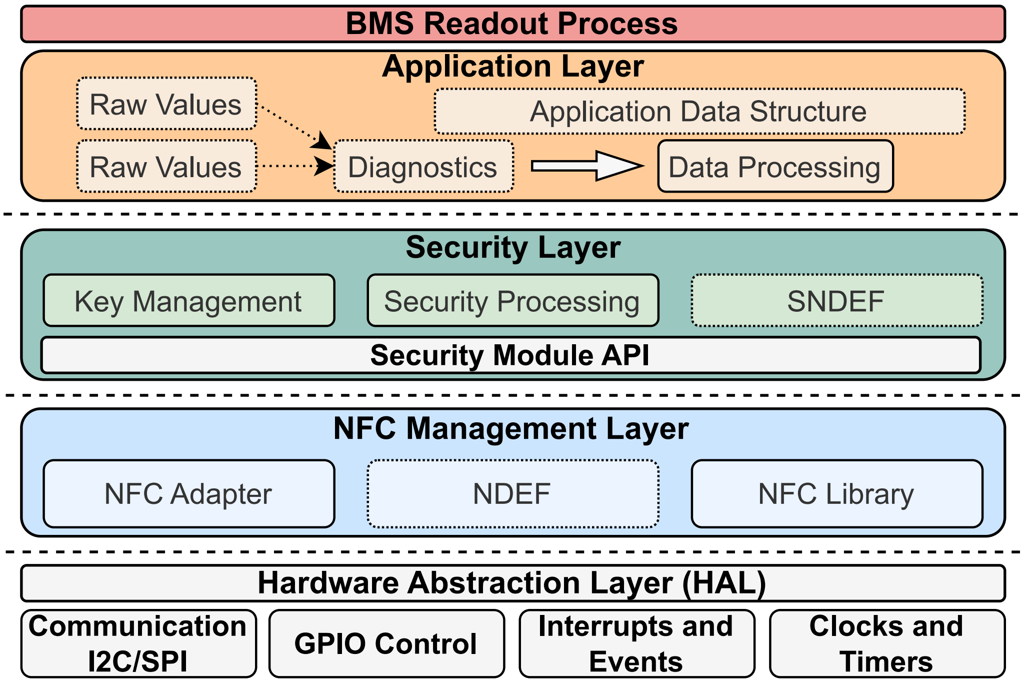


Figure 4.4: BMS software layer stack for NFC readout. The BMS Readout Process is first observed through the Application Layer. The Security Layer is responsible for the secure data processing between the application and the NFC transfer layers. The NFC Management Layer allows for the NFC wireless transfer and data processing, with the HAL being responsible for the vendor-specific driver interactions and configurations.

design of this layer depends on the individual OEM basis BMS design. However, in most derivations and topologies we observe the same methodologies centered around sensor data processing.

For the active sensor use case, the BMS controller would manage the retrieved diagnostic data that has been previously sampled by the BPC through the adjacent battery packs sensors and NFC components. However, the final destination of the data, i.e., further processing from the BMS controller side, would rely on the intended application. According to the digital passport proposal, the battery data can be stored either offline or online in a database [15]. We aim to support both derivatives, with the extended design considered in the following Chapter 5.

Two data models have been developed to support the readout of the BMS NFC data:

- Middleware data structure. Centered around the proposed SNDEF, which is a secure data frame on top of the NDEF structure independent of the processed data.
- Application data structure. Intended for data encoding derived from the BMS monitoring and diagnostic processes independent of the number and distribution of the BPC.

Secure-NDEF (SNDEF)

The communication between the devices is designed to run over the NFC Forums' NDEF records. To enable secure data exchange in response to the requirements listed in Section 4.1.2, and defined security models Section 4.3, a secure NFC message structure called SNDEF has been proposed that builds on the NDEF frame structure. The frame structure is based on a model first proposed by Ulz et al. [123]. It has been extended to accommodate a broader range of security protocols, e.g., for AEAD schemes in addition to the traditional block modes of AES MAC protocols, making it more flexible for modern use.

Figure 4.5 shows the proposed SNDEF frame structure. It contains the following mandatory fields:

- *Cipher specification.* Predetermined coded specification for the security configuration.
- *Initialization vector (IV).* For block encryption algorithms additional security value. It needs to be generally kept unique and/or random.
- *Tag.* Additional information to guarantee the frame's integrity. It can be an appended MAC value.
- *Secret payload.* Contains encrypted application data with the following sub-fields:
  - *Message ID.* Unique message identifier. The value itself does not need to be unique, as long as its combination with the following message counter is unique.
  - *Message counter.* It is used to keep the chain of messages and as a guard against replay attacks. Each message needs to have a unique counter value for each key. The counter value can be repeated but only for a different session key.
  - *Message payload.* Together with message type and message length preamble, with each being 1-byte long. It contains the application message. The message type is application-dependant, e.g., "READ_STATUS", "UPDATE_CONFIG".

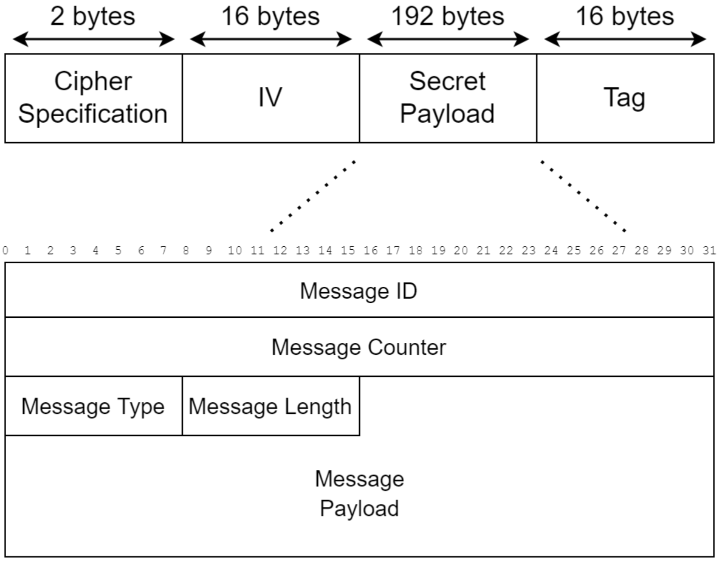


Figure 4.5: Secure-NDEF frame structure. It consists of four main data blocks, among them 'secret payload' that has its own structure and encrypted message payload field for the intended transfer payload. Adapted from *Publication D*.

Due to frame constraints, the message payload is limited to 182 bytes, i.e., it is recommended to use a fragmentation protocol when dealing with larger message sizes. While SNDEF can be used for all three BMS NFC use cases, it is primarily intended to be used for the two diagnostic use cases since the active sensor use case does not require channel encryption, which would only introduce additional overhead.

Application data structure

In order to support the unique design requirements of the proposed BMS NFC use cases, we have developed a specialized message packet structure. This structure provides high flexibility for different system uses while minimizing decoding overhead. We have removed any extraneous data that would increase packet sizes and, consequently, transfer time.

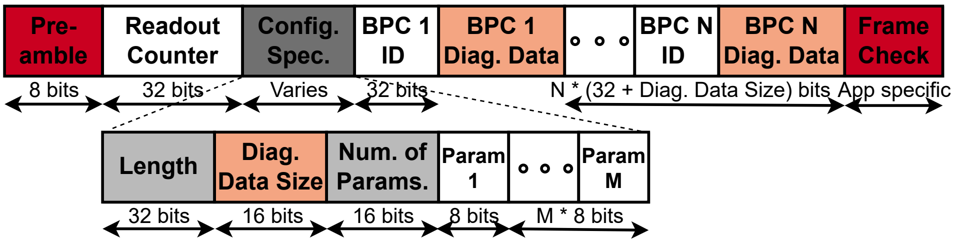


Figure 4.6: Proposed application packet structure for BMS and BPC NFC diagnostic readout use case. The structure is made flexible to accommodate different application uses by containing open configuration fields and expandable diagnostic data size. Adapted from *Publication G*.

The application packet structure illustrated in Figure 4.6 is made up of the following fields:

- **Header.** It starts with a short "*Preamble*" to differentiate between the start and end point of an BMS sample readout. It is followed with a "*Readout counter*" as a timestamp for the data sample order, and "*Configuration specification*" that contains two mandatory fields and a number of optional application-dependant parameters:
  - *Length.* Total diagnostic data size.
  - *Diagnostic data size.* The number of diagnostic entries.
  - *Optional parameters.* Parameters that better describe the targeted data, or contain supplementary battery passport information. "*Number of params*" indicates the total number of *M* parameters, but the parameter code indicators are left to the developers.
- **Data field.** The main part of the message that is to be sent. It contains coded diagnostic information where each diagnostic readout data of a BPC is preceded with its identifier. The structure composition of the diagnostic data is application-specific, but in our research design, is based on the format that we presented in Section 2.1.1.
- **Footer.** Considers the last part of the message that is intended for verifying the integrity of the frame. It can be a simple Cyclic redundancy check (CRC) code.

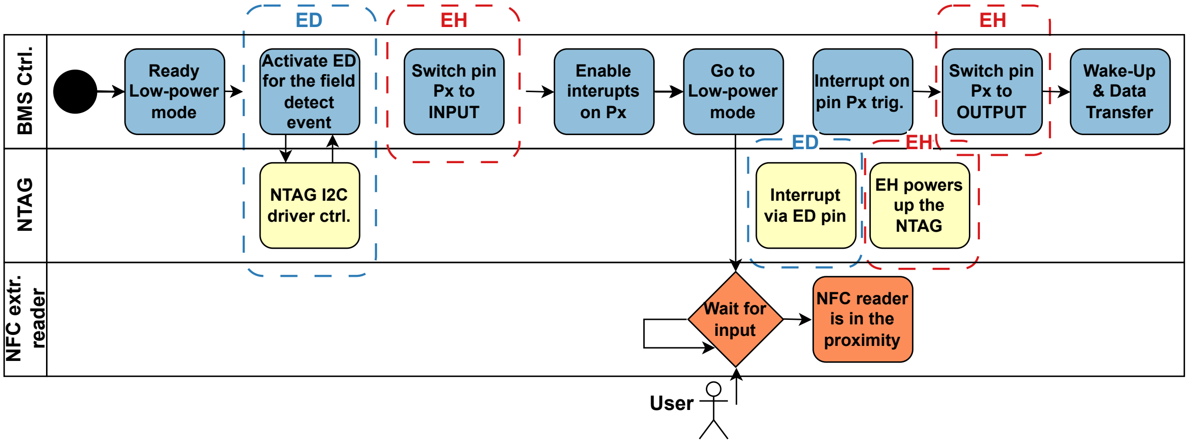


Figure 4.7: BMS and NFC wake-up process flowchart. Both methods share most of the same steps, with individual steps noted in event detection (ED) and energy harvesting (EH) blocks. Adapted from *Publication G*.

4.2.4 Energy harvesting and wake-up process

One advantage of the use of NFC compared to other wireless technologies is that it provides the *energy harvesting* capability [137, 138]. This means that by generating the field, the active device can power the passive device during the data transmission process. The passive device does not need to be connected to its power source, which further reduces the total number of necessary wire elements. The use of energy harvesting requires that the components be configured for a specific operating voltage, which in the modern design, caps at around 3 V [225]. In addition, the total distance between the active and passive devices is limited. However, this is of no concern with the presented system design, as the components and their interfaces are generally produced and packed close together. While energy harvesting can be used in all three specified wireless BMS readout use cases, it is particularly important for the idle diagnostic use case. In this environment, the battery packs would be stored away, and no passive energy leakage or draw should take place aside from during the status readout process.

Wake-up procedure. For stationary storage, it is important that the battery pack components and controllers are able to conserve power. The controllers would need to be 'woken up' during the readout period. Therefore, we would like to enable low power consumption and fast wake-up time by precisely designing the wake-up process [226]. Based on the characteristics of energy harvesting, we observe two main methods, *event detection wake-up (EDW)* and *energy harvesting wake-up (EHW).* An overview of the characteristics and comparisons of both methods can be found in Table 4.2.

Table 4.2: Overview and comparison of the proposed wake-up system design approaches.

Model	Prerequisites	Pros / Cons
EDW	• NTAG needs to have an event pin • requires a constant power source	+ wake-up is possibly faster - needs constant power source - higher power consumption
EHW	• the reader has to have EH enabled • NTAG's EH needs to be specially configured both from HW and SW	+ NTAG is powered off in idle + BPC can supply the NTAG - wake-up takes possibly longer

The main difference between the EDW and the EHW is how the wake-up is triggered. The EDW relies on the use of the event detection features found on the modern NTAG. The event can be configured to trigger the response when the harvested energy exceeds a certain threshold, i.e., when the mobile reader comes into proximity of the BPC's NTAG. The NTAG remains in the low-power state during the idle period. The EHW relies on the wake-up of the BPC to go through the direct energy harvesting function. With this method, the NTAG can remain completely powered down. In both methods, the BPC is kept in a very low power state (VLPS) during the idle period. After the wake-up is triggered, it enters the active state and can then respond to the readout commands. Figure 4.7 shows the proposed wake-up process with individual steps concerning both EDW and EHW methods. In theory, the EHW results in lower power consumption but may take slightly more time for the complete wake-up since the NTAG is powered down in this model. Further analysis is performed in Section 7.1.1.

4.2.5 Novel battery pack system design

After considering the system design requirements and definitions, it is evident that the development of modern BMS and battery packs must prioritize the incorporation of new security and interface components while keeping the additional cost relatively low. This is especially important when realizing the wireless design through the presented use of NFC. To meet these essential production constraints, the following conditions have been established:

1. *Limiting available security functions.* Security modules often come with many additional available functions. More complex silicon also entails higher prices, and custom or low-cost security modules need to be considered when maintaining the balance between features and price.
2. *Controlling the number of additional modules.* Each additional module also entails additional complexity in the PCB design and production.
3. *Accurate communication interfaces.* Additional elements used for the communication interfaces should follow proven and right standards. In our case, this means that the NFC should work over the reader/writer mode with supporting modules.
4. *Minimizing maintenance expense.* Installation overhead should be kept on the production level to minimize any additional cost or complexity that comes from future maintenance. This is especially important with security, i.e., authentication configuration data, that would be shared and maintained between the vendors and the users.

The enhanced battery pack would include one or more smart sensors consisting of two main elements, (i) a **NFC passive element** for wireless communication and (ii) a **security module** for handling relevant security functions. To control the communication and secure readout, the control unit would also need to be extended. The security module may be a HSM or a simplified and user-defined TPM. It must contain at least one secure area, i.e., a electrically erasable programmable read-only memory (EEPROM) for storing intermediate data, keys, certificates, unique identifier (UID), etc., as well as a hardware implementation of the relevant security operations, e.g., for AES, MAC, and EC. If cost permits, the security module could also include various symmetric crypto modes of operations, a random number generator, KDF, and security-related functions. However, based on the analysis from Section 4.3.1, the majority of the security functions would only add overhead and are not necessary for running the only desirable security operation at this level of communication, namely authentication. The proposed additional components of the battery pack system design are listed in Table 4.3.

In order to reduce expenses, it is necessary to minimize the size of the EEPROM and store only a limited amount of pre-embedded data. The amount of data required depends on the available resources and the level of security and design complexity, which is outlined in Table 4.4. If a digital signature is being used, implicit certificates, as discussed in Section 2.3.3, might be a better option. They require a much smaller footprint than explicit certificates, such as the X.509 format.

For a low complexity system design, the minimum data requirements for EEPROM are as follows:

- *UID*: up to 32 bytes
- *Private key*: 256-bit key size, 32 bytes
- *(optional) Public key*: 256-bit key size, 64 or (if compressed) 33 bytes
- *Certificate*: (varies)

If relied on implicit certificates, the public key is optional as it can be derived from its own certificate and is generally not used for native operations. For security reasons, the EC asymmetric keys must have a length of 256 bits. For the certificate length, we take the same as defined later in Section 5.2. The intended EEPROM would require at least $32\,B + 32\,B + 101\,B = 165\,B$, + additional user or other

Table 4.3: Additional components for the novel battery pack sensor unit for a wireless and secure readout.

Component	Part of ...	Description
MCU	Control unit	For the process control of communication and security operations; can also be an SoC, or an ASIC
NTAG	Communication unit	Interface module for communication, relays sensor data; with an RFID antenna placed outside
Security processor	Security module	Adequate hardware unit for hardware-accelerated security operations
EEPROM	Security module	For storing and handling security-related configuration data; needs to be tamper-proof

auxiliary data. Since most EEPROM embedded in security chips today come with at least 1 kB or more available data space [227], the introduced memory limitation would not be a problem even if different levels of security complexity requiring more data are used.

Implementation discussion

The presented BMS with NFC design is studied for any type of sensor found in battery packs, primarily considering temperature and pressure sensors. The architectural solution is also adapted to fit the needs of any BMS environment, either automotive or industrial. The closer the sensors are to the battery cores, the more accurate they can predict their behavior. The utilization of NFC makes it possible to achieve deeper sensor installations, but since the actual placement and installation of NFC components correlates with the readout performance, it would need to be investigated separately.

Concerning hardware, special attention needs to be diverted to the antenna design and placement, as well as the necessary calibrations such as the impedance matching [228]. The quality of the NFC hardware configuration directly correlates with the limits of its range and reliability, especially when introducing higher data transfer rates. Next to hardware optimization, software needs to be also specially adjusted. The optimization of software stacks from Figure 4.4 can be observed under the low-level driver optimizations and higher application stacks. The adjustments to be made need to comply with the NFC standard, but also set to achieve the most efficient and reliable readout process.

4.3 Security Design

When implementing new BMS systems in real-world architectures, it is important to consider the new vulnerabilities that arise when moving from wired to wireless communication. Referring to the security requirements outlined in Section 4.1.2, we can identify two primary areas of protection for the BMS and NFC system design. The first area focuses on battery pack authentication as a security measure for the active sensor readout use case, while the second area involves the implementation of a full security suite protocol for both external active and idle diagnostic readout use cases.

4.3.1 Battery pack authentication

Based on the previous security requirements analysis from Section 4.1.2, we consider a security design for internal verification of BMS modules to protect against counterfeit or malicious devices. Specifically, we study the authentication of battery packs by NFC-related components. Based on the BMS topology, this could mean either (i) authenticating a group of battery cells or (ii) authenticating battery cells with their BPC, depending on the deployment. In the latter case, the authentication would be processed exclusively by the main BMS controller, while in the former case, the authentication would be processed by the BPC controller towards a relevant communication point, e.g., via the sensor communication interface and the corresponding security module. In the presented design, this would be the adjacent NTAG module, which contains both the controller for the wireless interface for communication with the sensors and a security module for security operations.

It is believed that authenticating the battery cells is sufficient to verify the authenticity of the battery pack as a device and prevent counterfeiting. However, it is essential to have at least one method of mutual authentication, meaning both the BPC and battery pack must be authenticated.

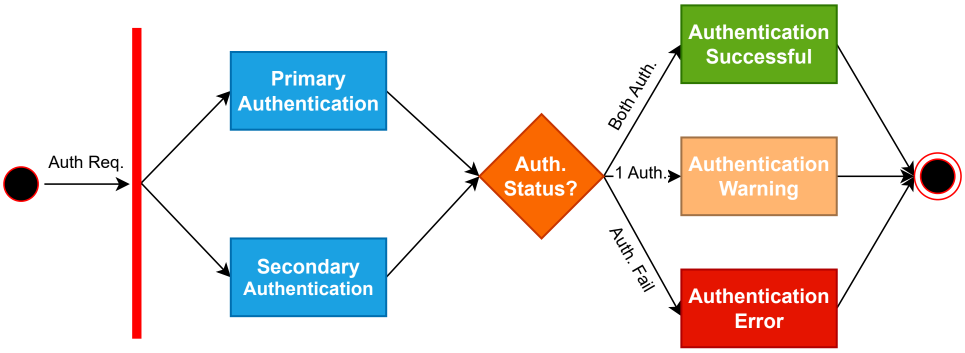


Figure 4.8: Flowchart diagram of the proposed battery pack two-factor authentication design.

Two-factor authentication. A two-factor authentication method is proposed for authenticating the battery packs. One form of authentication is based on something known and embedded with the device, while the other may be an additional form of authentication protocol. To consolidate foreseen constrained requirement of the appended battery pack interface and sensor hardware, the authentication approaches of choice should also be lightweight and simple, yet secure in nature. The conferred authentication overhead is only incurred at system startup or interaction, e.g., vehicle startup, and therefore would not intervene with the rest of the BMS and battery active cycles. In Figure 4.8, we can see the integration of two-factor authentication. The first authentication method is called *primary authentication*, while the second one is called *secondary authentication*. If the secondary authentication fails, the user will receive a warning, but the system will remain in its normal active state. However, if both authentication methods fail, an error indicator will activate, and the action will be treated as a threat until it is further resolved.

The choice of authentication methods depends on the offered system functionality, but in our case we assume smart sensors and modern NFC tags that can provide some level of protection at minimal cost. It is assumed that the verifier supports rudimentary symmetric and asymmetric crypto functions, i.e., AES encryption and EC signature verification. The primary authentication would be based on an "originality" characteristic of the hardware, e.g., PUF, a password, or a digital signature. This type of authentication is usually difficult to forge, but at the same, also difficult to update because it is embedded in the device. The secondary authentication should provide an additional layer of protection but still maintain the requirement to be fast, efficient, and lightweight. It can be a second password, a read-only UID, or a symmetric crypto authentication, e.g., a challenge/response mechanism using AES. Certain authentication methods, such as the mentioned challenge/response mechanism with AES, could also provide the aspect of *mutual authentication*, i.e., verification of both sides of the communication, which could be beneficial in some BMS cases. Table 4.4 shows an example of three different levels of system and security complexity with proposed security mechanisms used for each authentication step.

Table 4.4: Internal BMS battery pack module authentication based on the level of security and system complexity.

Design & security complexity	**Primary authentication**	**Secondary authentication**	**Comment**
Low	UID	Digital signature	Affordable with full authentication, but only on the BPC side; UID needs to be write protected
Medium	Password	Digital signature	Requires authorization knowledge providing partial mutual auth.
High	PUF	Chg./Resp. AES	Would require a proven PUF; AES can provide mutual authentication

4.3.2 Security protocol for external readout

A security protocol is designed for the external diagnostic readout operation. As per requirements, the proposed security design must be not only secure but also lightweight and efficient. To achieve these properties, the protocol design is based on symmetric cryptography. It starts with the secure handshake phase followed by the secure session phase.

Secure handshake phase. It is intended for mutually authenticating both parties, i.e., the external mobile reader and the BMS or BPC unit, depending on the use case. The protocol is shown in Figure 4.9 and can be formalized as follows, with N_R: mobile NFC reader and its identifier, M_N: BMS or BPC and their identifiers, ch_r: request challenge value from N_R, ch_t: request challenge value from M_N, K_M: embedded master key, K_S: derived session key, X & X': concatenated received messages during the entire secure handshake phase from N_R as X, i.e., from M_N as X'.

$$1)\ N_R \rightarrow M_N : N_R, ch_r \tag{4.1}$$

$$2)\ M_N \rightarrow N_R : M_N, ch_t, \{\{M_N, ch_r\}_{K_M}\}_{K_M} \tag{4.2}$$

$$3)\ N_R \rightarrow M_N : \{\{N_R, ch_t\}^{-1}_{K_M}\}^{-1}_{K_M} \tag{4.3}$$

$$4)\ M_N \rightarrow N_R : \{M_N, X\}_{K_S} \tag{4.4}$$

$$5)\ N_R \rightarrow M_N : \{N_R, X'\}_{K_S} \tag{4.5}$$

The protocol incorporates three main steps:

1. *Mutual authentication*: the devices authenticate each other using a challenge/response model.
2. *Session key derivation*: both parties derive the symmetric session key based on the pre-configured KDF and protocol. To protect against certain derivations of replay and “chosen challenge oracle”

attacks, double encryption/decryption operations are employed along with a dedicated check on challenge nonce requirements. Challenges can not be zeros or equal to one another.

3. *Session key possession verification*: during the current handshake phase, both sides confirm the possession of a valid session key by considering the messages that were exchanged. The verification of the key possession is not only useful from a functional standpoint, but it also extends the security confidence that the messages exchanged come from the current session and are not part of a previous reply message.

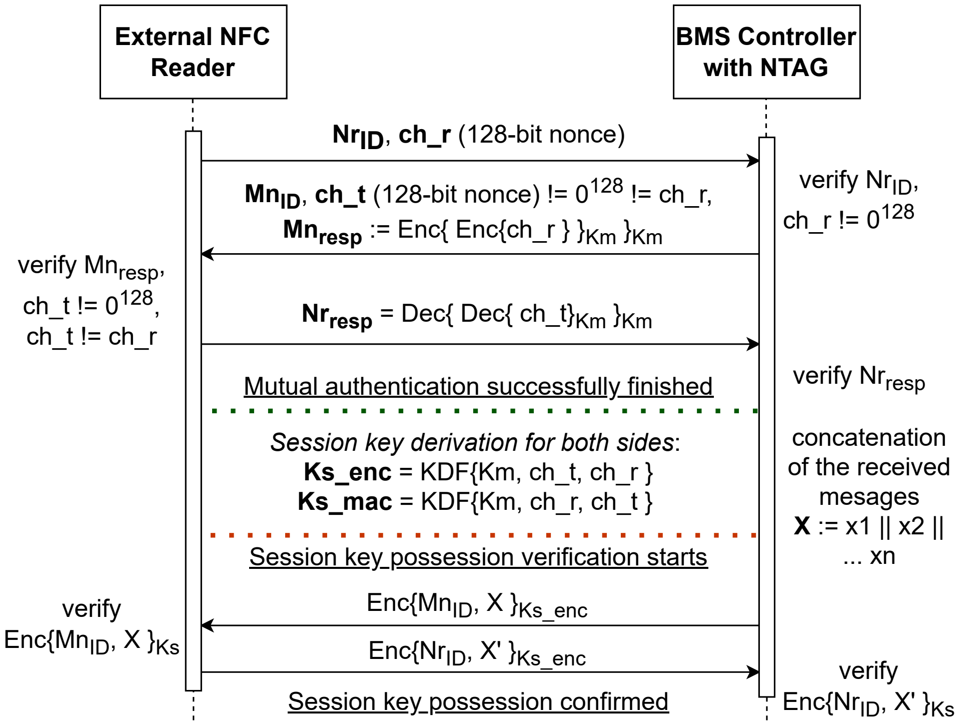

External NFC Reader
BMS Controller with NTAG
Nr_{ID}, ch_r (128-bit nonce)
verify Nr_{ID}, ch_r != 0^{128}
Mn_{ID}, ch_t (128-bit nonce) != 0^{128} != ch_r,
Mn_{resp} := Enc{ Enc{ch_r } }$_{Km}$ }$_{Km}$
verify Mn_{resp}, ch_t != 0^{128}, ch_t != ch_r
Nr_{resp} = Dec{ Dec{ ch_t}$_{Km}$ }$_{Km}$
verify Nr_{resp}
Mutual authentication successfully finished
Session key derivation for both sides:
Ks_enc = KDF{Km, ch_t, ch_r }
Ks_mac = KDF{Km, ch_r, ch_t }
concatenation of the received mesages
X := x1 || x2 || ... xn
Session key possession verification starts
Enc{Mn_{ID}, X }$_{Ks\_enc}$
verify Enc{Mn_{ID}, X }$_{Ks}$
Enc{Nr_{ID}, X' }$_{Ks\_enc}$
verify Enc{Nr_{ID}, X' }$_{Ks}$
Session key possession confirmed

Figure 4.9: Sequence diagram showcasing the designed BMS and NFC security protocol for the external diagnostic readout. The protocol relies only on symmetric cryptography with 128-bit strength. It consists of three main stages: (1) mutual authentication, (2) session key derivation, and (3) session key possession verification.

Secure session phase. The secure channel is now established and protected via data encryption, integrity, and authentication checks. Depending on the system's capabilities, it uses either AES + '*operating mode*', or an AEAD algorithm. The communication relies on the SNDEF for the secure encapsulation of the application data on top of the NDEF structure. SNDEF messages can be configured to better match the intended content of the system. In the proposed SNDEF design, a simple message counter is used instead of a validity field. Originally, the validity field consisted of a timestamp and an internal counter. However, the timestamp can be omitted in cases where we can ensure that each communication session uses a different, i.e., unique, session key. Omitting a timestamp can save on design space and processing time. This can be especially important for systems that are constrained and not able to easily derive the current timestamp, i.e., they may not have a real-time clock.

To protect against replay attacks, it is possible to include the tag of the previous message as additional input to the MAC of a new message. This mechanism is called *tag chaining* and it prevents potential exposure to accidental reuse of a session key. Additional security mechanisms can be provided by the NFC management layer through administrative control of *read & write* commands.

Discussion on the key derivation and updates

The proposed security model follows Kerckhoffs's principle [229], which emphasizes that the security of the model should depend solely on the keys used, with the model and operations being publicly known. Therefore, the master key data must be securely stored on the device in a unique manner. If this is maintained, security operations can be executed without any compromise. However, if the master key is exposed, it could pose a threat to the overall security structure of the model.

Session keys used for the communication are derived based on the standardized and secure KDF by using the following formulation:

$$K_S = KDF(K_M \,||\, \mathit{UID} \,||\, \mathit{PRODT\_STR} \,||\, \mathit{SEED} \,||\, \mathit{PADDING})$$

- K_M is either the master key or a previously derived session key.
- *UID* is a unique identifier for the device.
- *PRODT_STR* is a custom product string that is handled as an optional input.
- *SEED* is a dynamic attribute that can be built from a random nonce.
- *PADDING* as an optional argument to round the KDF input to the same value length.

By relying on a dynamic seed, we are able to achieve *forward secrecy*. This means that each new key derivation is independent of the previous one and its exposure would not affect the keys used for the previous messages [230]. While both sides can rely on the KDF and the previous keys with nonce generation for the key updates, there are also other methods. Specifically, "*Key wrapping*", where a class of symmetric cryptographic algorithms is used for encrypting the communication key, and "*Key sharing schemes*" that rely on wider architectures combined with additional exchange channels [95, 96].

CHAPTER 5

BMS Security Architecture

Summary: *In this chapter, we present the design for the core contributions to the novel BMS security architecture. Where the previous design chapter focused on wireless and local BMS connectivity, here we analyze the extended BMS network communication and security design with the local system networks and remote services. The chapter begins with an analysis of the security requirements, followed by the design overview for the local BMS security architecture, secure key derivations, the design of the secure BMS data structure, and concludes with the secure data propagation model for on-premise and cloud services.*

◇◇◇

The BMS security architecture is envisioned to protect devices and data exchanges at all levels of communication for the targeted system shown in Figure 5.1. We observe the following main entities: one or more BMS sub-systems, other control units, e.g., ECU, a central gateway, and remote services such as the cloud or end-user systems. Each research block builds on the previous one. The security architecture is intended to be consistent with all currently known BMS topologies and models [26, 22]. The architecture covers secure handling of the BMS and associated entities in an internal system network and complements the design with security considerations for full external correspondence.

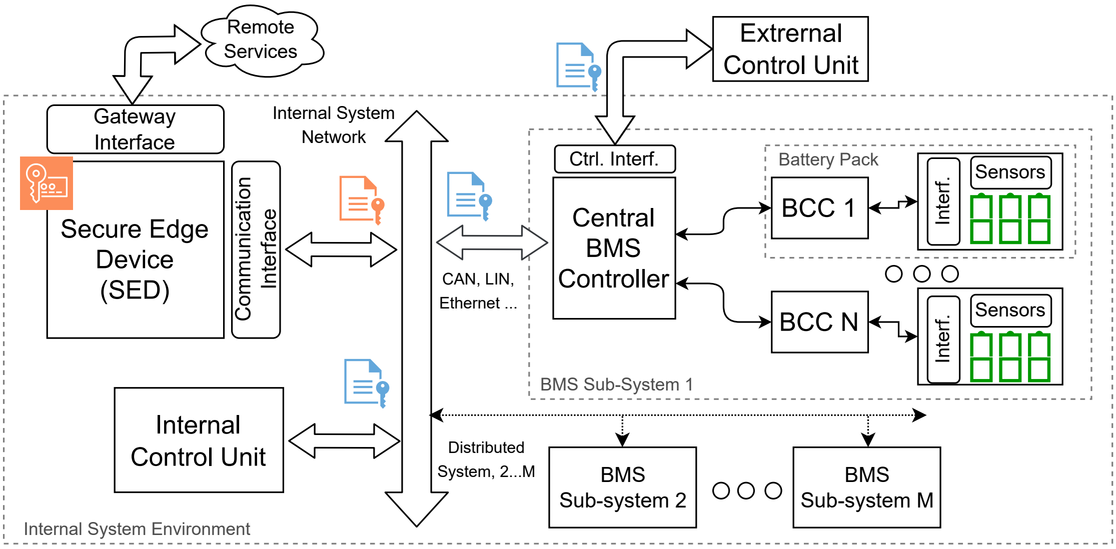


Figure 5.1: Targeted system architecture for the BMS security design. Adapted from *Publication D*.

5.1 Security Requirements

The security architecture must be supported by security requirements and analysis. In this context, we observe the current SotA BMS security analysis [51, 29, 30, 48], as well as the security analysis on vehicles and related systems [28, 163, 52, 179, 115]. In order to exhaust all possible attack vectors and define exact requirements that correspond to real-world system deployments, the network is observed to be vulnerable to any type of foreign intrusions, with a full range of resources available to attackers to mount every capable attack. Thus, we can consider that the attacker can assume the role of an ‘oracle’ with full access to the security protocol information and past messages.

An attacker would be motivated by the possibility of reverse engineering or technology exploitation, including as a form of industrial espionage. In addition, an attacker might want to expose users’ privacy through data generated by the BMS in order to extort ransom, conduct exploitation, or simply vandalism. Depending on the different realized access options, an attacker might launch attacks based on the three main access layers:

1. *Physical*: manipulations directed at the BMS sub-system. Potentially the most difficult for the attacker to exploit, due to the closed nature of the BMS sub-system.
2. *Local*: considers attacks that are launched from the internal system network. This is also the central focus of protection, which comes from the very necessity to protect the main access points of the BMS. Attacks can range from both passive and active attacks.
3. *Remote*: attacks on the BMS-related data going from the central gateway, i.e., the internal system network, to the remote cloud and end-user systems. Conventionally covered by a dedicated security suite, but also requires a wider angle of protection.

With regards to further security analysis and to support the proposed BMS security architecture, we make the following assumptions to our design:

- The central gateway is secure against common threats with both functional and security operations being run under a trusted environment, with the device seen as a root of trust.
- Security operations that run on all involved devices as part of the system architecture are done over a trusted security module.
- The external cloud services are done over a trusted and verifiable service provider.
- Key generation and distribution for the end-system devices are managed by a verified service and known protocol supported by the associated OEM.

BMS security requirements analysis is further divided and analyzed under three main groups of research interest: (i) security considerations for BMS associated local internal system network, (ii) separate analysis of the communication channel requirements for protection against emerged vulnerabilities, and (iii) analysis of the BMS data security requirements on all routing points of data transfer.

5.1.1 Network and device vulnerabilities

Figure 5.1 shows the internal system network that interconnects a BMS to gateway and control units. In the past, these network points, such as CAN, were left unsecured and unencrypted, leading to system vulnerabilities [181, 179]. This allowed malicious devices to attach to the network and provide the attacker with direct access to execute various attacks. Additionally, an attacker could launch node-capturing attacks to take control of an ECU, which could have serious implications [178].

A BMS is interested in protecting its configuration data, log, processing, and diagnostic data, as well as security-related configuration data [29, 49]. These are considered the most important assets when communicating with an external network. Since a BMS communicates with multiple devices on the network, the most important weight of security would be placed on authentication [51]. Each device, including the BMS, would need to authenticate each other. Each new device accessing the network would have to be verified by the gateway before any further communication is possible. Since the BMS data is safety-relevant, it must be protected from tampering. Manipulated data could at best case simply be displayed inaccurately, but at worst could lead directly to unintended system behaviours [48].

5.1.2 Communication session vulnerabilities

The protection of the communication session between a BMS and any other potential device is tied to the protection of the derived symmetric session key. Security is maintained by Kerckhoff's principle [229]. The derivation of session keys must be supported with the *perfect forward secrecy*. Perfect forward secrecy, often abbreviated to just forward secrecy, is a security attribute that guarantees that disclosed session keys do not affect the knowledge of future or previously derived session keys [231, 232]. This means that all previously encrypted messages remain safe from the attacker's eyes and that only messages encrypted with the disclosed session key are affected.

Forward secrecy is a standard component of modern security suites, such as TLS. In the context of constrained embedded devices, e.g., as is the case with ECU, the forward secrecy attribute is often not included as part of the security design. This is mainly due to the old, and partially false pretense, that forward secrecy adds excessive additional overhead to the already-in-place security protocols. Other reasons could simply be that the targeted use cases would not benefit from forward secrecy from the developers' point of view and that attacks targeting session key exposure are unlikely.

We believe that perfect forward secrecy is highly necessary for the modern design of cyber-physical and embedded systems due to the increase in communication with external services and systems, i.e., it is necessary to consider this security feature also for the communication with BMS sub-systems. For designing a secure key derivation and exchange protocol for the devised BMS use cases, we exhaust and observe the following threats that can affect the communication channels going from and in the BMS controller when considering an internal network [184, 191, 183, 30, 28]:

- *Exposure of past messages*: if a session key, or otherwise, session-related material is exposed, past messages can be decrypted and read. I.e., vulnerability due to the negligence of forward secrecy.
- *MitM attacks*: or specifically, the MitM attacks that lead to the exposure and unauthorized derivation of session keys or material.
- *Node capturing attacks*: the key derivation and exchange protocol should be protected against possible exploitation in case of a successful node-capturing attack.
- *Key data reuse*: protocols that rely on the reuse of previous key-derivation data are prone to vulnerabilities in case of exposures.
- *Key derivation exploitation*: the key derivation process needs to be subject to high-enough entropy, with the key-related material only being handled and stored by the valid parties.

5.1.3 Data logging and propagation vulnerabilities

As discussed in earlier chapters, one of the major drawbacks of current BMS is that they have limited data logging capabilities [26]. However, relying only on data logging functionality on itself is not enough; appropriate security measures must also be provided to protect the logged data. With respect to the BMS, we can observe data security based on: (i) BMS sub-system on-premise security, and (ii) cloud and end system data propagation security. Therefore, the protection link must be established from the source of the battery sensor to the end system device that processes the BMS data [11, 67].

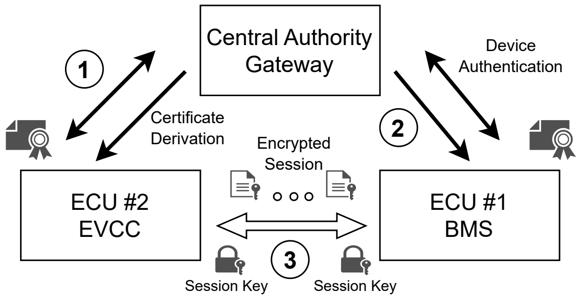


Figure 5.2: BMS security architecture deployment based on a centralized gateway authority with steps ① and ② covering the device authentication and certificate derivation for individual ECU in the network, and step ③ session key derivation and secure session establishment. Adapted from *Publication F.*

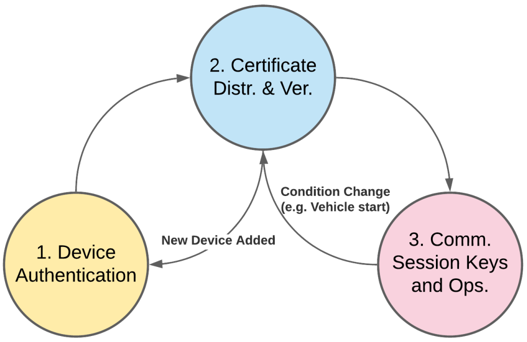

2. Certificate Distr. & Ver.
1. Device Authentication
3. Comm. Session Keys and Ops.
New Device Added
Condition Change (e.g. Vehicle start)

Figure 5.3: Three main deployment cycles and conditions for the BMS local network security architecture.

The attacks on the BMS sub-system log data can be difficult to carry out as the BMS is an enclosed device. Hence, attacks would have to be mounted from the inside via a spoofed device or from the outside in case sealing leaks can be detected [11, 22]. If the attacker can somehow gain access to the intermediary local network, they can launch a type of a MitM attack [49] or attack the log content itself by reordering or manipulating data block packets [168]. Communication with the cloud system can also be impeded by a DoS attack, against which there are few defenses. The system must be made robust enough to cycle easily between logging on-premise and the cloud and vice versa.

5.2 BMS Local Network Security Architecture

The secure system architecture of BMS is based on the centralized architecture design [191] considering communication channels and elements, as shown in Figure 5.1. As illustrated in Figure 5.2, each control unit participating in the communication on the local system network where a BMS resides must first be authenticated and receive the required security configuration information from the CA gateway before it can open a communication session. For example, communication between two ECU, one represented by the BMS sub-system, and the other via the EVCC, would first need to be authorized by the gateway before they are even able to derive the proper symmetric session keys. The derived asymmetric, i.e., public and private keys, are mainly used for mutual authentication. The gateway is also responsible for routing communication with remote services and managing update control.

The BMS secure system architecture is proposed to run through three major system cycles, as shown in Figure 5.2 and Figure 5.3:

1. **Device authentication.** This cycle is intended to be run each time a new device is detected on the network [233]. The central gateway authenticates the device, i.e., the BMS and any other control unit, based on the proposed symmetric cryptography protocol.
2. **Certificate derivation.** The central gateway, acting as CA, issues the certificate for each pending device on the network. This step occurs immediately after device authentication and is retriggered by other prerequisite events.
3. **Secure session.** With the certificates in possession, a BMS is able to establish a secure communication channel for data transfer with any other valid control unit, e.g., ECU, in the network [234]. A session key derivation and exchange protocol is employed before the communication starts.

The device authentication controlled with the gateway must happen each time a new device is detected on the network. On the other hand, the authentication between the devices using implicit certificates is proposed to take place under the following two events:

- *Vehicle start.* During the ignition start of a vehicle, i.e., each time the ECU are initialized.
- *After a system event.* This refers to any significant system update, such as through the on-board diagnostic (OBD)-II port, cloud updates, mobile updates, and so on, as well as routine updates that occur after each configuration cycle.

With respect to the overall BMS secure system architecture, before the system can be deployed, a *fabrication* pre-deployment cycle would need to take place. During this cycle, in particular, the initial configuration for the central gateway system, i.e., for the CA, must be defined, specifically:

- What security primitives and functions are going to be employed, i.e., what hash functions, what elliptic curves, and what is the key length?
- What format should the certificate have and what data must it contain?
- Specification of any other additional auxiliary data, the chain of authentication length, special case considerations, etc.

In the next two sections, we will discuss how the device authentication and certificate derivation deployment cycles and their protocols were further realized. The secure session cycle with the proposed protocols for key derivation and exchange are analyzed under the separate Section 5.3.

5.2.1 Device authentication

The first step after deploying the system or after recognizing a new device on the network is to authenticate it at the device level. The protocol we propose is shown in Figure 5.4. It is based on symmetric cryptography with pre-embedded keys to allow greater flexibility and security for a wider range of devices. Specifically, authentication is based on a challenge-response approach, where 'C' is the initial challenge as a random value and 'R' is the response from the BMS side.

The protocol starts with the gateway receiving an authentication request from the BMS. The gateway prepares the challenge 'C' and updates its keys as temporary keys for the request. It also derives a random nonce N_{SED} in case of a passive attack where the network might be eavesdropped. A MAC algorithm is used to verify the integrity of the sent messages and also to authenticate their source. The BMS will derive its own temporal keys and proceed to verify the received MAC value. To protect against replay attacks and differentiate each request, the protocol relies on the dynamic use of random nonces [191] (N_{SED}, N_{BMS}) by utilizing arithmetic operations to hide potential nonce traces when transferring the response from the BMS to the gateway. The response 'R' contains the gateway's challenge and the dynamic message nonce. It is encrypted and appended with the MAC value.

First, the gateway checks the received MAC and response. Then, it verifies the nonce value by reversing the arithmetic operation. If the authentication is successful, the gateway sends back the encrypted certification configuration data to the BMS. After this, both the gateway and the BMS update their authentication key. This new key will be used as an input for further device authentication, providing a partial forward secrecy attribute.

5.2.2 Certificate derivation

The main foundation behind the proposed security architecture lies in the use of implicit certificates, specifically, the ECQV scheme [235, 191, 133]. The certificates are used for mutual authentication between the BMS sub-system and any other external device. They are also used as a basis for computing

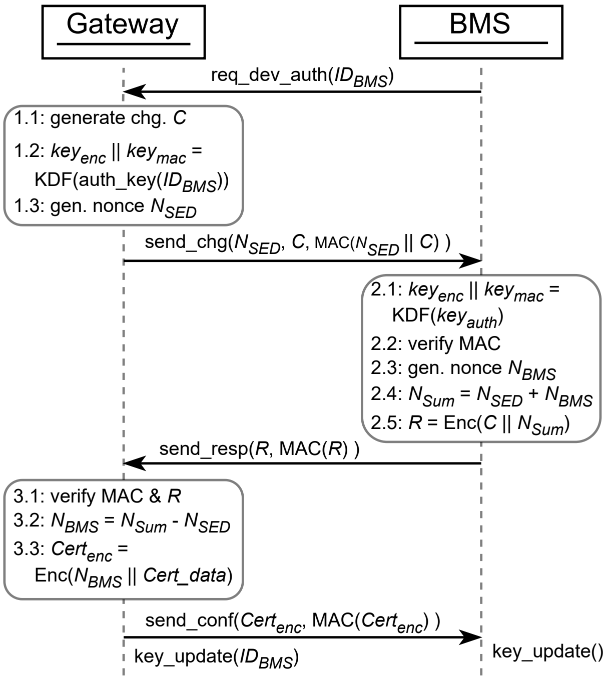

Gateway
BMS
req_dev_auth(ID_BMS)
1.1: generate chg. C
1.2: key_enc || key_mac = KDF(auth_key(ID_BMS))
1.3: gen. nonce N_SED
send_chg(N_SED, C, MAC(N_SED || C))
2.1: key_enc || key_mac = KDF(key_auth)
2.2: verify MAC
2.3: gen. nonce N_BMS
2.4: N_Sum = N_SED + N_BMS
2.5: R = Enc(C || N_Sum)
send_resp(R, MAC(R))
3.1: verify MAC & R
3.2: N_BMS = N_Sum - N_SED
3.3: Cert_enc = Enc(N_BMS || Cert_data)
send_conf(Cert_enc, MAC(Cert_enc))
key_update(ID_BMS)
key_update()

Figure 5.4: Proposed device authentication protocol for mutual authentication between BMS and Gateway in an internal local network. The protocol is based on the challenge (‘C’) and response (‘R’) mechanism. $N_{SED}, N_{BMS}, N_{SUM}$ are nonces used for the authentication step. key_{enc} & key_{mac} are symmetric keys used for the encryption and MAC calculation of the current request. Adapted from *Publication D.*

the *private & public keys.* From Section 2.3.3, implicit certificates are smaller than traditional explicit certificates and are, therefore, faster to transmit and easier for constrained systems to store. The certificates are derived by the CA, which is the central gateway in our case.

The certificate derivation and exchange protocol is based on the original ECQV scheme, extended to also account for other processes regarding the overall design of the proposed BMS security architecture. Namely, it consists of these three main steps as modeled in Figure 5.5:

1. **BMS request specification.** A request is prepared by the main BMS controller connected to the internal local network. It derives the ‘*request value*’ from a random value and the agreed EC ‘G’ point value. Afterward, a random nonce is derived that is used to guard against potential replay attacks, with a MAC value calculated from the request data. The MAC value is used next to the BMS session identifier to validate the source and provide non-repudiation. This is imperative as a guard against future potential certificate possession missuses [182].
2. **Certificate calculation.** The CA gateway derives the certificate and provides the *reconstruction value.* However, before it can send the certificate and the reconstruction value to the BMS controller, it first needs to verify that the request came from a valid source. It checks the session identifier alongside the recalculated MAC value and received nonce. Before the gateway sends the reply back, it also derives its own response nonce together with the response MAC value.
3. **Keys derivation & verification.** After receiving the certificate from the gateway, the BMS will first calculate the MAC value to confirm that it is authentic and not tampered with. It will then proceed to calculate its private and public keys based on the received reconstruction value and the implicit certificate. Finally, it will verify the accuracy of the public key with ‘$prk * G$’.

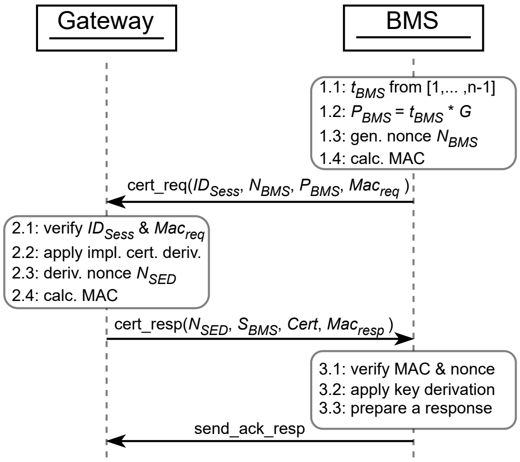

Gateway
BMS
1.1: t_{BMS} from [1,... ,n-1]
1.2: $P_{BMS} = t_{BMS} * G$
1.3: gen. nonce N_{BMS}
1.4: calc. MAC
cert_req(ID_{Sess}, N_{BMS}, P_{BMS}, Mac_{req})
2.1: verify ID_{Sess} & Mac_{req}
2.2: apply impl. cert. deriv.
2.3: deriv. nonce N_{SED}
2.4: calc. MAC
cert_resp(N_{SED}, S_{BMS}, $Cert$, Mac_{resp})
3.1: verify MAC & nonce
3.2: apply key derivation
3.3: prepare a response
send_ack_resp

Figure 5.5: Gateway implicit certificate derivation for BMS sequence diagram. The protocol consists of three steps based on the ECQV scheme and is extended with nonce and MAC verifiers providing protection against replay attacks and repudiation. Step 2.2 follows the Algo. 1 for certificate derivation, while step 3.2 contains Algo. 2. Adapted from *Publication D.*

Algorithm 1 Implicit certificate derivation.

```
Input: $ID_{Sess}, P_{BMS}$
Output: $S_{BMS}, Cert$
Generate $k_{BMS} \in_R [1, ..., n-1]$
$U_{BMS} \leftarrow P_{BMS} + k_{BMS} * G$
$Cert \leftarrow Encode(ID_{Sess}, U_{BMS})$
$S_{BMS} \leftarrow (Hash(Cert) * k_{BMS} + prk_{SED} * G)\ mod\ n$
return $S_{BMS},\ Cert$
```

Algorithm 2 Public & private keys calc.

```
Input: $S_{BMS}, Cert$
Output: $prk_{BMS}, pub_{BMS}$, status
$prk_{BMS} \leftarrow (Hash(Cert) * k_{BMS} + S_{BMS})\ mod\ n$
  $pub_{BMS} \leftarrow Hash(Cert) * Decode(Cert) + pub_{SED}$
  if $pub_{BMS} == prk_{BMS} * G$ then
  |  return $prk_{BMS},\ pub_{BMS}$
else
  |  return $false$
end
```

Recertification question. A crucial aspect to consider when designing a PKI for the presented system is, “*When and how should certificate validity be addressed?*”. This is a common open question when it comes to in-vehicle networks and is particularly relevant to BMS. Specifically, certificates must be updated based on their validity input after a certain amount of time has elapsed. In case the vehicle has been stored for a long time, the BMS might still be valid and with that, the battery packs, but the validity field could be expired. The full authentication of the otherwise valid BMS is now put into question. To address these concerns, several design considerations could be made. First, the gateway could rely on the previous security material and issue a warning until the following system and certificate update occurs. Second, an extended service could be provided by the OEM to manually verify and update the security configuration on the BMS controller, but this would depend heavily on the availability of the target system. Third, the validity field could be relegated to the background during authentication or omitted altogether. This is still an open question that is left for future research work, which would involve the existing PKI deployments in general vehicle systems.

5.3 Advanced Key Derivation

In this section, we will discuss the proposed session key derivation protocols for communication between the main BMS controller and another control unit inside a local area network. Although based on the security requirements analysis, where we strongly recommend perfect forward secrecy with a DKD, we also present a SKD to complete the design and account for BMS environments with highly constrained devices. The performance output of the protocols should be appropriately optimized in terms of the tradeoff between performance and achieved security. For better security coverage, both protocols rely on EC for authentication via ECDSA, while differing in the key derivation procedure.

The SKD is based on the traditional Diffie-Hellman key exchange where the derived symmetric session key is the product of the private and public keys of each party. The protocol is shown in Figure 5.6. On the other hand, to achieve the *perfect forward secrecy* attribute, our proposed DKD method is based on the STS protocol [230]. It is a novel and advanced design and the first in the literature to use the well-established STS protocol with an implicit certificate scheme, in our case that being the ECQV scheme. The STS protocol was adapted to account for the public key derivation and verification with ECQV, as shown in Figure 5.7. To further complement protection against some niche replay attacks, a mechanism can be adapted as seen by Porambage et al. [197], where the SKD protocol can be extended to include an additional verification step at the end. Here, each of the previous handshake messages is appended and afterward encrypted by each party and sent over the network to confirm that all message exchanges occurred only in the current session between the valid devices.

Both SKD and DKD methods use the ECDSA to provide the authentication [236]. The derivation of the public key (2.1 & 3.1 from Figure 5.6, i.e., 3.1 & 4.1 from Figure 5.7) is based on the certificates:

$$Q_X = Hash(Cert_X) * Decode(Cert_X) + Q_{CA} \tag{5.1}$$

The difference between the SKD and DKD methods comes from how the key material is derived. To provide ephemeral keys, the STS uses the EC derivation of a random point on each request:

$$X \in_R [1, ..., n-1] \rightarrow XG = X * G \tag{5.2}$$

Both methods use the same key derivation principle by relying on an KDF as seen in eq. 5.3, but the manner of the input differs. The SKD relies on the Diffie-Hellman (eq. 5.4), while DKD uses the previously derived and exchanged random EC points (eq. 5.5):

$$K_S = KDF(K_{PM}, salt) \tag{5.3}$$

$$K_{PM-SKD} = prk_A * pub_B = prk_B * pub_A \tag{5.4}$$

$$K_{PM-DKD} = X_A * XG_B = X_B * XG_A \tag{5.5}$$

Implementation discussion. The protocols can be used on top of the Controller Area Network (CAN) stack or any other communication standard. The application layer needs to provide the correct cipher spec information. Symmetric encryption can be of any traditional block cipher, e.g., AES, or AEAD derivations. By using the conventional block ciphers it is possible to rely on the implementation optimizations from both hardware and software perspectives.

The duration of one session would need to be specially investigated and is application dependent. In this context, the caching of the previous registration data could be used to allow for a faster connection re-establishment. For both the performance and security design aspects, it is recommended that the security processes are handled through a hardware-based security solution, e.g., trusted platform modules, secure elements, or security co-processors [53].

5.3.1 Dynamic key derivation optimization

The STS-ECQV protocol for key derivation and exchange offers higher security compared to other SotA protocols for implicit certificates key derivation, but it also comes at a slight performance cost, as it is

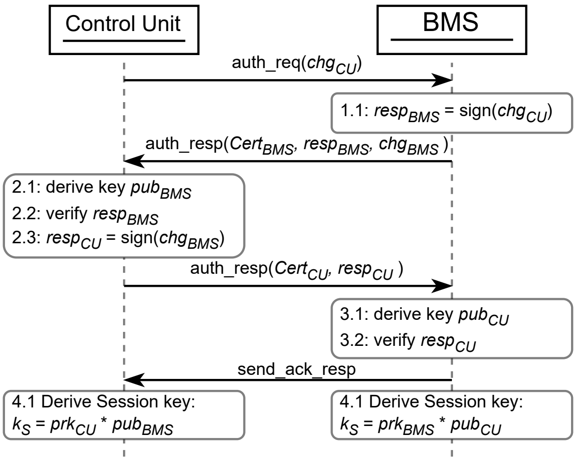

Control Unit
BMS
auth_req(chg_{CU})
1.1: $resp_{BMS}$ = sign(chg_{CU})
auth_resp($Cert_{BMS}$, $resp_{BMS}$, chg_{BMS})
2.1: derive key pub_{BMS}
2.2: verify $resp_{BMS}$
2.3: $resp_{CU}$ = sign(chg_{BMS})
auth_resp($Cert_{CU}$, $resp_{CU}$)
3.1: derive key pub_{CU}
3.2: verify $resp_{CU}$
send_ack_resp
4.1 Derive Session key:
$k_S = prk_{CU} * pub_{BMS}$
4.1 Derive Session key:
$k_S = prk_{BMS} * pub_{CU}$

Figure 5.6: ECQV static key derivation protocol sequence diagram. The protocol is based on the Diffie-Hellman and uses EC signatures for the verification. The protocol can be extended with an additional step at the end against replay attacks by encrypting and sending handshake messages from each party. Adapted from *Publication D.*

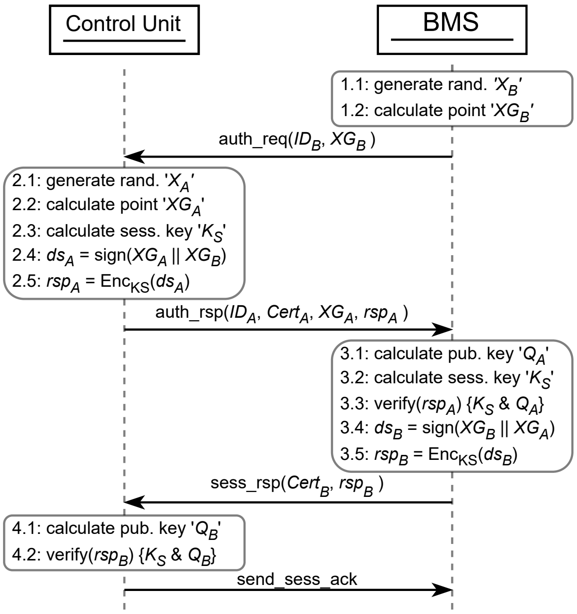

Control Unit
BMS
1.1: generate rand. 'X_B'
1.2: calculate point 'XG_B'
auth_req(ID_B, XG_B)
2.1: generate rand. 'X_A'
2.2: calculate point 'XG_A'
2.3: calculate sess. key 'K_S'
2.4: ds_A = sign(XG_A || XG_B)
2.5: rsp_A = Enc_{KS}(ds_A)
auth_rsp(ID_A, $Cert_A$, XG_A, rsp_A)
3.1: calculate pub. key 'Q_A'
3.2: calculate sess. key 'K_S'
3.3: verify(rsp_A) {K_S & Q_A}
3.4: ds_B = sign(XG_B || XG_A)
3.5: rsp_B = Enc_{KS}(ds_B)
sess_rsp($Cert_B$, rsp_B)
4.1: calculate pub. key 'Q_B'
4.2: verify(rsp_B) {K_S & Q_B}
send_sess_ack

Figure 5.7: STS-ECQV dynamic key derivation protocol sequence diagram. Offers perfect forward secrecy. Adapted from a poster publication [237].

analyzed later under Section 7.2.2. To reduce this drawback, we analyze the structure of the protocol and offer two optimization steps that can be applied through the act of parallelization.

The proposed STS-ECQV protocol from Figure 5.7. can be divided into four operations (Op):

Op1 - Initial request phase with the derivation of the random value 'X' and point 'XG'
Op2 - Generation of the public key via the implicit certificate and session key derivation
Op3 - For authentication, signature calculation and encryption of the request
Op4 - For authentication, decryption of the request and verification of the signature

Figure 5.8 shows the STS-ECQV protocol between two parties after the division and new arrangement of the operational steps. Similar to the work done by Sciancalepore et al. [200], in the first request step (Op1), rather than just sending the 'XG', the certificate is also sent (by Alice) that can already be used by the other party (Bob) for the public key derivation. Bob now immediately sends their own data back to Alice before initiating the next Op2. Since both parties have the necessary data, they can in parallel derive public keys and calculate the foreseen session key. This is followed by Op3 where the authentication data is prepared, and Op4 where this data is verified.

Based on the organization of the operations, we can see two potential parallelization phases, one targeting Op2, and the other Op3. The reason for this division is that, while Op2 does not suffer from any drawbacks in terms of the actual distribution of the operations, Op3 pushes the verification step to be done at the end of the protocol. This means that failed authentications are detected later, which can lead to longer processing time and even possible misuse by the means of DoS, or similar attacks. Nevertheless, the protocol does not suffer from any otherwise security vulnerabilities as it adheres to the fundamental calculation and verification steps of the original STS protocol.

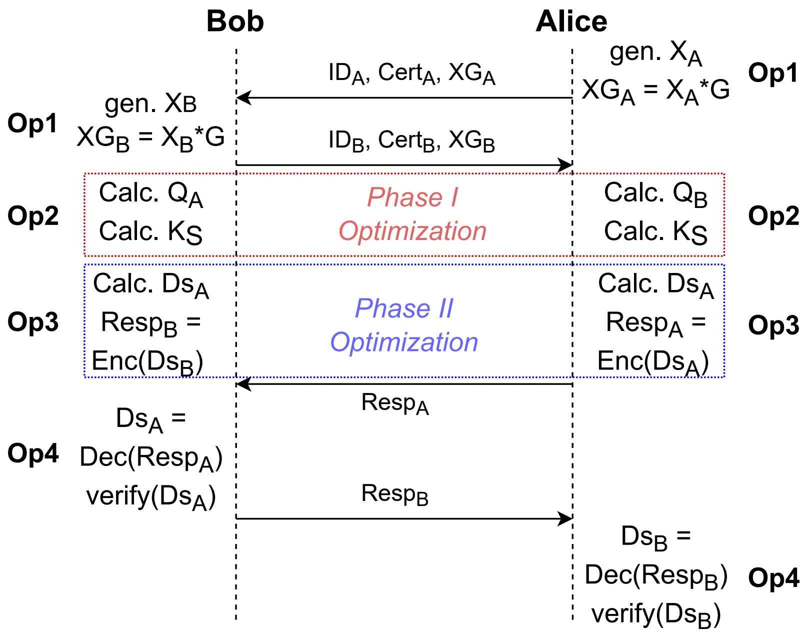


Figure 5.8: Sequence diagram of the optimized STS-ECQV dynamic key derivation protocol.

Execution time analysis

To better understand the advantages gained by applying these optimization steps, we will take a look at the projected time analysis. Figure 5.9 shows the time duration comparison of the individual operations.

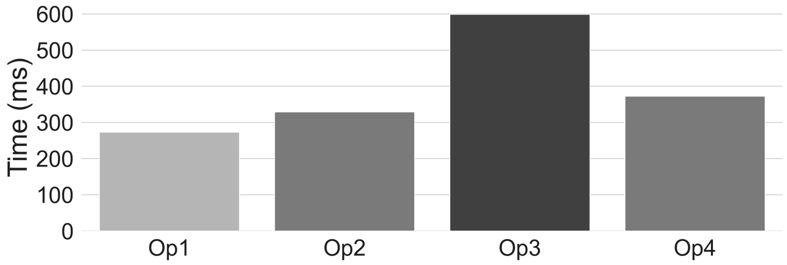


Figure 5.9: Time duration comparison of the specified STS-ECQV key derivation operations. The measurements were done using a program written in *C* and run on an STM32F767 MCU. Adapted from *Publication F.*

Since each step is done once on each side in sequential order (from Op1 to Op4), the total execution time can be presented as:

$$\tau_T = \sum_{i=1}^{N_{Op}} T_{OpAi} + \sum_{i=1}^{N_{Op}} T_{OpBi} \text{ ,with } N_{Op} = 4 \tag{5.6}$$

In the ideal sense, the optimization through parallelization allows saving the total execution time up to the execution time of Op2, i.e., Op3. However, this does not take into account delays that can happen due to the transmission traffic or the differences in device hardware.

We can deduce that the additional time, without considering the network, is equal to the following:

$$\forall x \in \{2,3\},\, T_{OpAx} = \begin{cases} 0, & \text{if } A = B \\ |T_{OpAx} - T_{OpBx}|, & \text{otherwise} \end{cases} \tag{5.7}$$

Based on equation 5.7, we can observe that the total execution time for 'A' when considering only phase I (Opt. I), or both phases (Opt. II) is equal to the equations 5.8 & 5.9, i.e., in the ideal sense when both devices are equal and there is no difference in network transmission, it is equal to 5.10 & 5.11.

$$\text{Opt. I} \quad \tau_T^{'} = 2 * T_{Op1} + T_{OpA2} + T_{Op2} + 2 * T_{Op3} + 2 * T_{Op4} \tag{5.8}$$

$$\text{Opt. II} \quad \tau_T^{''} = 2 * T_{Op1} + T_{OpA2} + T_{Op2} + T_{OpA3} + T_{Op3} + 2 * T_{Op4} \tag{5.9}$$

$$\text{Ideal Opt. I} \quad \tau_T^{'} = 2 * T_{Op1} + T_{Op2} + 2 * T_{Op3} + 2 * T_{Op4} \tag{5.10}$$

$$\text{Ideal Opt. II} \quad \tau_T^{''} = 2 * T_{Op1} + T_{Op2} + T_{Op3} + 2 * T_{Op4} \tag{5.11}$$

Table 5.1: BMS data logging and storage models.

Storage approach	**Advantages**	**Disadvantages**
Local central - one central memory unit on the main BMS controller	Easier to implement, handle and secure; Generally lower cost	Options limitation; Performance hit for larger amounts of data; Reduced portability
Local modular - a memory unit per BCC	Portable with replaced Battery cells; Easier to track log content for battery passport use cases	Potentially higher cost; More work on making the implementation optimal when handling large amounts of data
Remote - dedicated ECU or Cloud	Larger storage capacity; Higher flexibility in using different data formats	Higher implementation complexity; More difficulty to guarantee security

5.4 BMS Secure Data Model

From a design perspective, BMS log data can be processed either individually on BPC units, on the BMS controller, or remotely either via a different controller, e.g., another ECU, or with the cloud. Table 5.1 shows these three main approaches and discusses their advantages and disadvantages.

In this dissertation, we rely on a hybrid approach that uses both local, i.e., on-premise, models, combined with remote or cloud support. To support this notion, a novel secure data structure for the BMS is proposed. It is based on a hierarchical model where we distinguish between three main structures.

Log block. This fundamental structure contains monitoring and diagnostic data specified for each individual BMS, i.e., for each individual battery pack unit. While it is application-specific, the base design remains the same. The structure (Algo. 3) contains a header and the main body with log sample data. The *header* (Algo. 4) always contains the block identifier, the identifier of the next log block in the sequence, and the length of the log block body. The structure is specifically designed to account for low overhead, with the majority of variable data being contained in the *body* of the log block (Algo. 5). The log body sample has been designed based on the BMS log data analysis from Section 2.1.1. To account for changes and placement of battery packs, the log blocks do not have fixed positions, rather each log block points to the next one in the sequence of installation up to the number '*N*' of the given BMS as seen in Figure 5.10.

BMS block. It is used to keep track of each sampling sequence of a particular BMS. The BMS block is also aimed to keep track of static data relevant to the battery passports (Algo 6). It contains mandatory fields such as identifiers, timestamps, but also optional metadata. Rather than keeping a pointer to the next BMS block, the blocks are sequenced and tracked on the remote side using the timestamp.

Secure BMS block. The application exchanges the data over the network using this structure (Algo 7). The secure BMS block is encrypted to protect against eavesdropping and tagged to protect its integrity. The tag makes the footer, while the body contains the encrypted *BMS block* with appended *log blocks*. Cipher-relevant data, lengths, and identifiers are contained inside the secure BMS block header. For security reasons, most recent copies of the secure BMS block reside on-premise with the main BMS controller. The key derivation and exchange between BMS and remote clients are done separately.

Algorithm 3 Log block full structure.

```
Struct LogBlock contains
    LogBlockHdr log_header
    LogSample    log_body;
end
```

Algorithm 4 Log block header.

```
Struct LogBlockHdr contains
    int block_id
    int next_block_id
    uint32 block_body_len;
end
```

Algorithm 5 Log block body.

```
Struct LogSample contains
    Log_Meas* measurements
    Log_Diag* diagnostics
    Log_Fault* fault_regs;
end
```

Algorithm 6 BMS block structure.

```
Struct BmsBlock contains
    uint32 bms_block_id
    uint32 timestamp
    uint16 unit_id
    int init_log_block_id
    uint16 metadata_len
    Bms_Metadata bms_metadata;
end
```

Algorithm 7 Secure BMS block structure.

```
Struct SecBmsBlock contains
    uint16 version
    uint16 length
    uint32 sec_bms_block_id
    uint32 sec_bms_block_serial
    uint16 cipher_info
    uint16 enc_bms_block_len
    uint8* iv
    uint8* enc_bms_block
    uint8* mac;
end
```

The full proposed hierarchical BMS data structure and block dependencies are shown in Figure 5.10. As observed, the remote system maintains a decentralized data chain architecture by only needing to keep track of the BMS blocks. Each block is considered an independent and individually abstracted structure. Before the BMS block is attached to the current data chain structure, the secure BMS block is first "*decapsulated*" i.e., decrypted, on the remote receiver side. Figure 5.11 shows the intended structure of fully secure transmission data from a BMS sub-system to the remote cloud and end-user system.

To complement the battery passport use case, BMS blocks contain the Metadata field, which is optional. This field is intended to be associated with battery passports to more accurately describe hosted BMS sub-systems. Since this data is static and does not change often, i.e., only in the case of major system updates or replacements, it is made to be optional to save storage space. The remote system that logs the BMS data can keep track of these changes using an array stack that adds each new entry with each new BMS block containing the metadata field.

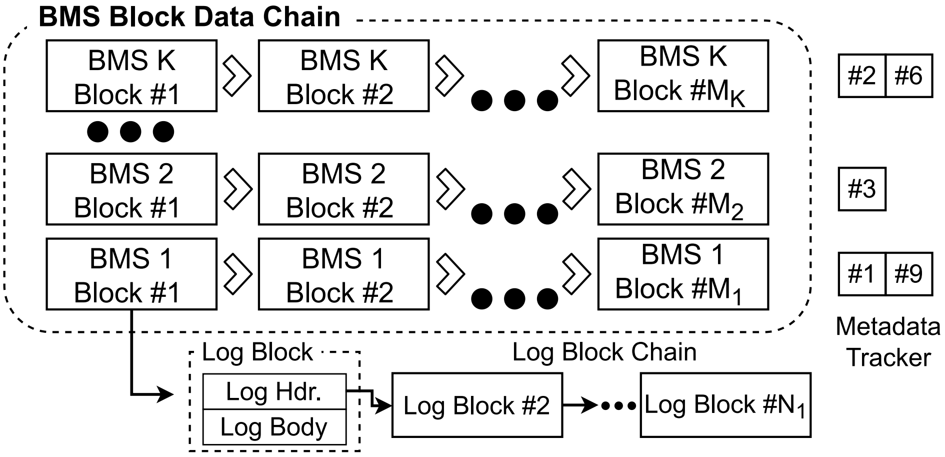


Figure 5.10: Proposed BMS data chain structure for logging lifecycle data. 'K' is the number of currently tracked BMS sub-systems. 'M_i' is the number of currently logged BMS blocks for the 'i' sub-system, while 'N' is the number of the sub-system's log blocks. The structure is independent of the set BMS topology and can be used with different system deployments with the following considerations for one log sample - *centralized*: 1 BMS block, 1 log block; *modulated*: 1 BMS block, N log blocks; *distributed*: 1 BMS block, N log blocks; *decentralized*: K BMS blocks, $N_1, ..., N_K$ log blocks. Adapted from *Publication H.*

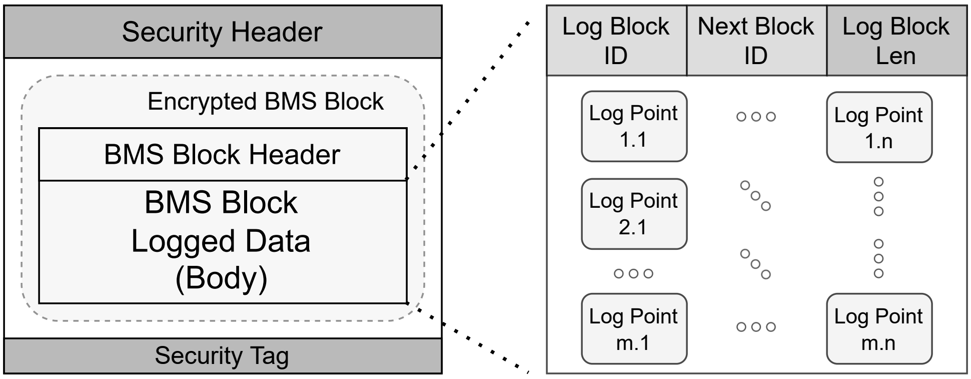
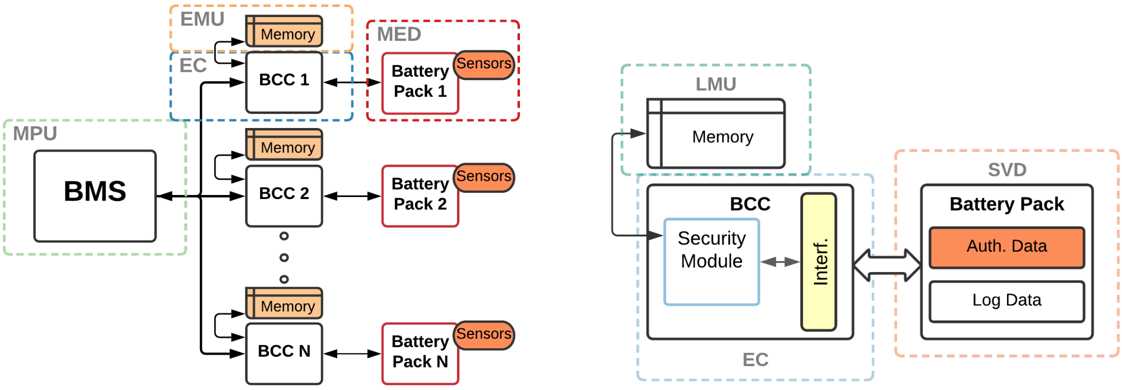


Figure 5.11: Full structure of the proposed Secure BMS Block. Adapted from *Publication H.*



(a) Applying EP2M design pattern.

(b) Applying SEL design pattern.

Figure 5.12: Figures (a) and (b) present the application of the EP2M & SEL design patterns with BMS concerning logging and security functionality for battery cell data. Based on the patterns' solution, the BMS controller takes the role of the Main Processing Unit (MPU), with the BCC being the Embedded Controller (EC). It communicates with individual External Memory Units (EMU) observing the data from the Monitored Embedded Devices (MED), i.e., battery pack sensors. Concerning security, additional hardware and software engines are integrated with the battery pack being seen as a Source Verification Device (SVD) and the attached memory as the Logging Memory Unit (LMU). Appropriate design, sequence, and functionality conditions are to be applied as specified in the pattern's solutions for the addressed system case problems. Adapted from *Publication A*.

BMS system design to support secure data acquisition

Research on the secure system design for BMS data acquisition and processing has led to the discovery of two architectural design patterns [238]. They can be used in system design to overcome the challenges presented with designing a secure data model [239]:

> **EP2M**: a pattern for system design aimed at streamlining the production of embedded devices by decentralizing and modulating the data logging components and tasks.
> **SEL**: a pattern with guidelines for extending the data logging to account for security coverage by focusing on the data pipeline between the main embedded controller and the memory unit.

Both design patterns offer a flexible and cost-efficient solution for developing constrained embedded systems. To supplement their use with a distributed BMS and account for the second-life use, we apply the design patterns for the system design phase. The high-level design aspects are shown in Figure 5.12 [1]. By integrating the EP2M and SEL patterns early on in the development process, designers can determine the necessary interface connections and module positions for BPC and its adjacent units. The EP2M pattern serves as a precursor to SEL, which establishes the necessary connection to an external memory unit and the BCC (or BPC). SEL provides an additional layer of security by protecting data confidentiality, integrity, authenticity, and repudiation. These patterns prioritize a modular approach, making each component a separate unit to allow for flexible replacements and verification.

[1] In the figures and the Paper, the notation Battery Cell Controller (BCC) is used in place of Battery Pack Controller (BPC). BPC serves as a vendor-free general terminology, but they both consider the same component.

Choosing local storage medium

For the on-premise data storage, several design questions need to be answered that depend on what data needs to be stored, what is the sampling frequency, and what is the desired level of security. These can be summarized into the following general requirements:

- The storage medium needs to be sufficiently large
- Data handling and storage need to account for moderate performance usage
- The memory module needs to have a sufficient life duration and be portable
- The data needs to be properly secured, guaranteeing its confidentiality, authenticity, and integrity, at the minimum

It is recommended that each BMS entity consider a suitable storage medium capable of handling the expected BMS monitored data at the design stage of its system. Additionally, it is essential to consider security measures at the hardware level. These measures may include using BGA packaging, employing an SoC design to make it less accessible, and utilizing a secure memory location.

5.5 Secure On-premise and Cloud BMS Data Monitoring

Secure monitoring of BMS data involves security design specifications for each layer of system deployment. Traditionally, when speaking about the BMS network architectures, the perceptual, network, and end-user application layers are primarily considered [11, 67]. To extend the security design at the local level, we introduce two additional layers and extend the architecture to consider five sequential deployment layers in total, as shown in Figure 5.13:

BMS sub-system. It consists of the BMS controller, one or more BPC and battery packs connected in a closed system. In terms of communication, it can only communicate externally via the BMS controller using a network interface or wirelessly with NFC via the main BMS controller and the BPC, as indicated in the design. The layered protection should follow the proposed design models already described in Chapter 4 and Chapter 5.

Internal local network. The internal local network presents one of the most important layers of protection, as it is prone to many vulnerabilities, previously investigated in Section 5.1. Under this layer, the main BMS controller communicates with other directly connected devices across one or multiple bus systems. The security mechanism must include three main operations as proposed in Section 5.2: device authentication, certificate and keys derivation, and secure session establishment with emphasis on perfect forward secrecy.

Central gateway. It enables central configuration, control and, as a gateway, remote data propagation. The central gateway unit, as the central security authority, should comply with common security specifications and standards [125, 126]. It must be adequately protected at both the hardware and software levels and have sufficient resources to manage multiple devices and their configurations. A security breach at the central gateway directly affects the security of the entire local network and, to some extent, remote end systems. For performance reasons, the gateway collects the secure BMS blocks for all network-based BMS and securely formulates them to send them further to the cloud layer.

Cloud data acquisition system. The cloud layer is responsible for receiving secure BMS blocks from the gateway and further propagating them to the end system backend. It relies on the use of the established security architectures schemes and protocols, with the security channel using either the TLS or DTLS protocols, and transmission protocols such as Message Queuing Telemetry

Transport (MQTT) or Constrained Application Protocol (CoAP) with different advantages and disadvantages [240].

End system backend. Under this layer, we consider any server or otherwise user device that processes BMS data. It is also the system that allows for battery passport backend operations. One question still needs to be addressed when it comes to securing the BMS block key transfer. The recommended approach is to use an End-To-End encryption method as it decreases the burden and offers privacy for the middle layers. The implementation of this step depends heavily on the situation, but it is recommended to use solutions that are already established in the automotive industry, such as the key exchange design with charging stations [127].

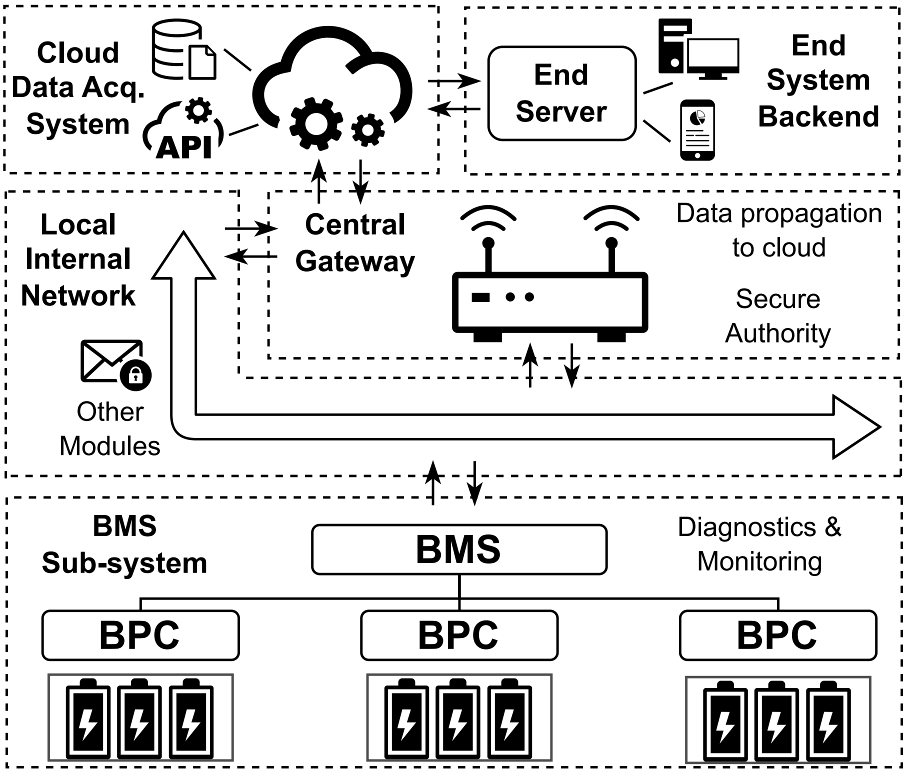


Figure 5.13: Proposed layered BMS security architecture for lifecycle data monitoring and logging. It consists of hierarchically sequential five different layers observed as autonomous entities. Adapted from *Publication H.*

5.5.1 BMS secure data trip: From the beginning to the end

As all important building blocks have been discussed and described, we will now present the full secure data traversal from and back a BMS and the remote backend system. Important design points will be discussed based on the phases of the deployment from the BMS data point of view. The study is done with the vehicle use case in mind, but it is applicable to any other architecture that deals with BMS.

The data traversal is observed from two directions:

- From the BMS to the remote system: standard logging and data acquisition from the BMS.
- From a remote system back to the BMS: as a feedback loop to feed in new BMS data, i.e., for configuration updates, machine learning outputs, synchronizations with other systems, etc.

The communication between the BMS and the remote end system is realized from three angles:

- From a mobile reader: during the diagnostic readout as indicated in Section 4.1.1, where the mobile reader, e.g., a mobile phone, can directly process and propagate data to the backend system.
- Internal network via the central gateway: using a local network, the BMS can communicate and propagate its data back to the backend by relying on the central security gateway or another network unit, e.g., a EVCC.
- Directly from the BMS: although not directly present in current systems and not covered in this dissertation, future derivations of BMS, especially in local independent systems, may rely on direct communication with remote backends.

Pre-deployment phase

Initial configurations take place. The gateway is configured with the appropriate initial keys for itself and other devices, the cypher suite specification, and the network connection to both the cloud and end system backend. The cloud service is also configured and set up to be available online. The end system backend has obtained appropriate keys for decrypting the BMS data structure. The BMS is embedded with its master key and initial cypher and system specifications. When new battery packs are inserted, they must be verified by the BPC, i.e., the BMS controller, based on the model from Section 4.3.1.

Start-up of the system

It is assumed that the BMS has been inserted for the first time into the system with the current BMS controller together with its adjacent devices. It is detected by the network's central gateway:

1. The gateway detects and marks the newly detected devices, i.e., the BMS controller.
2. The mutual device authentication is initiated once, relying on the Section 5.2.1 protocol.
3. After successful device authentication, the implicit certificates are derived for the BMS based on the protocol from Section 5.2.2.

Active monitoring phase

During the active period, data is logged by or fed back to the BMS controller. The data are sampled in their raw form by the battery cell sensors. They are then pre-processed and packaged into the secure BMS blocks as described in Section 5.4. The secure BMS blocks can be temporarily stored on-premise before being further propagated. In the active phase, the BMS would most likely actively send its data to the central gateway unless a special request is received. This request could come from another internal network unit or an external readout device. In all cases, a *secure session* with device authentication and session key derivation must be established before any further data transfers take place. In the case of the external mobile readout device, the communication would follow the guidelines of the proposed symmetric cryptography protocol, as seen in Section 4.3.2. The internal network communication, on the other hand, would rely on the asymmetric implicit certificates and use either a SKD or a DKD protocol proposed in Section 5.3. Any further network and external communication would also rely on the additional network encapsulation to provide a secure data transfer.

The directions described in this section should be applied to finalize the data traversal to the end system. This considers establishing ahead a secure connection between the secure gateway and the cloud system, i.e., the end backend device. With this, the full secure BMS data propagation is established.

CHAPTER 6

Implementation

Summary: *To accurately evaluate the proposed architectural design, an extended prototype BMS was implemented. This chapter addresses the main building blocks considered in the implementation and is divided into three main sections. The first section addresses the overall architecture of the test suite, which was created, maintained, and continuously extended over the course of the dissertation. The second section gives an insight into the structure of the communication architecture and mainly deals with the communication layers used and the protocols implemented. Finally, the third section discusses the implementation of security protocols, which were used for testing the core elements of the research.*

◇◇◇

6.1 Test Suite Implementation Overview

To demonstrate the feasibility of the proposed design architecture and analyze its applicability in a real-world environment, a test suite was designed and implemented. Various requirements were defined at the beginning and during development to allow for the realization of an accurate and usable research prototype test suite:

- *Req. 1: Accurate prototype representation.* Hardware components should match their real-world counterparts as closely as possible to provide a more accurate picture for scientific analysis. They should also be automotive-graded where possible.
- *Req. 2: Correct application extension.* Implementation of any additional functionality and security layers and protocols must be done on top of or in parallel with the existing software stack, which includes the BMS monitoring and diagnostic application layer. The extended code and functions should not interfere with the existing code and performance of the BMS diagnostic process.
- *Req. 3: Adequate security implementation.* Security protocols are to be implemented using either an accessible security co-processor or a trusted security software library.
- *Req. 4: Optimal run time and memory management.* Implemented advanced features should provide optimal usage with minimal additional runtime overhead. Memory buffers for storing, processing, and transmitting BMS data should also have minimal memory consumption since they are and would be executed on embedded devices.
- *Req. 5: Appropriate communication stack.* The communication stack, i.e., the data frame format and transmission protocols should be representative of a sophisticated and accurate usable design. Additional layers should be provided for functional purposes where appropriate, but with minimal resource expenses so as not to add unnecessary overhead.

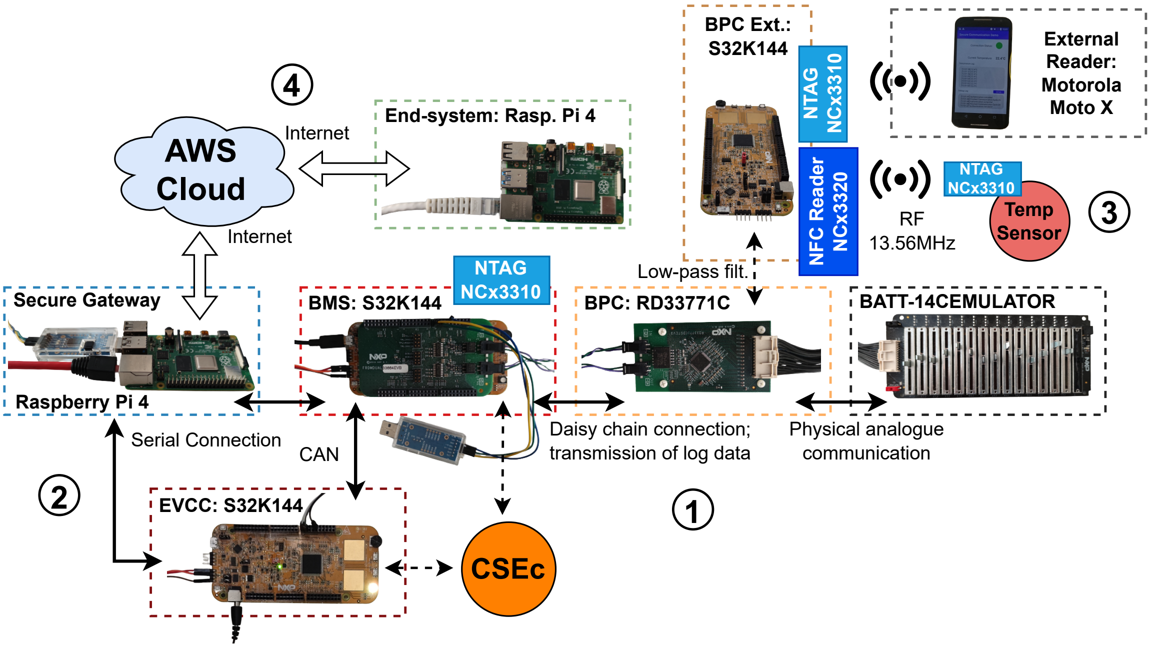


Figure 6.1: Building blocks of the secure and wireless **BMS test suite prototype** used for the development and testing. It consists of ① **a BMS emulator**: BMS controller (S32K144 with CSEc), BPC (RD33771C), and a battery cells' emulator (BATT-14CEMULATOR), ② **a local network** with: Secure gateway (Raspberry Pi 4), an exemplar ECU - EVCC (S32K144 with CSEc), ③ **NFC readout elements**: additional BPC controller (S32K144) for NFC device control with an NFC Reader (NCx3320) and NTAG (NCx3310), a temperature sensor with the NTAG (NCx3310), an external reader (Motorola Moto X), and ④ **remote services**: Cloud (AWS), and an end-system (Raspberry Pi 4).

The implemented test suite [1] is shown in Figure 6.1 with several building blocks consisting of the following main four module groups:

1. **BMS emulator.** The emulator is based on a BMS NXP reference design to fit the *Req. 1* conditions. It consists of three main building components representing the distributed BMS topology.
2. **Local network.** To simulate the representation of the in-vehicle, or other local, network operations, a universal asynchronous receiver-transmitter (UART) communication link was established between the BMS and a central gateway device. An additional controller was used for simulating the intra-session communication between the BMS and another ECU using the CAN link.
3. **NFC readout elements.** To convey all three main use cases, implementation was done using both NTAGs as the passive, and NFC reader as the active device. The readout from the battery pack is programmed and controlled using an additional MCU board. In Figure 6.1, only the names of the NFC chips and their placement are shown due to confidentiality reasons from the NXP side, as the project relied on internal custom boards.
4. **Remote services.** External service communication is represented by a cloud connection that goes over the gateway from the local network. An end-system device is also used for a full data transfer demonstration.

[1] For the research analysis, several other implementations have been done as well. Specifically, it was necessary to test the usability of the proposed key derivation protocols on embedded devices with various resource limitations. Since these implementations are evaluation-bounded, they are described in Section 7.2.2.

6.2 BMS Emulator

The BMS emulator contains three main elements:

1. *Main BMS controller*: represented by a S32K144 MCU board which is designed for automotive applications. The controller itself hosts an initial BMS monitoring and diagnostics applications alongside which the appropriate communication and security challenges are implemented.
2. *BPC*: as an RD33771C BCC, it is a bridge between the battery cells and the main BMS controller. The communication with the main BMS controller is done over the transformer physical layer (TPL) protocol. The S32K144 possesses an additional FRDMDUAL33664EVB shield for communication with one or multiple BPC, the latter being done by a “daisy chain” connection.
3. *Battery emulator*: with a BATT-14CEMULATOR that allows the emulation and precise analog transfer of targeted battery cell data. Specifically, it allows the readout of 14 battery cells with their voltage values and one temperature value for the whole pack. The configuration of the voltage and temperature values is done over adequate potentiometers.

The implementation on the BMS MCU is done exclusively in the programming language *C*. The BPC enables the processing of the battery-emulated data, which is subsequently post-processed by the main controller of the BMS. The original application on the main BMS controller would retrieve the diagnostic data from the connected BPC and process it in a loop. Before this process takes place, enhancements were made to the process of authenticating and establishing a secure connection with the external secure gateway. After this step, the keys are derived and logging and processing of monitoring and diagnostic data can begin. The data is post-processed and prepared based on the BMS block design described in Section 5.4. The authentication and session establishment steps are performed using a finite state machine (FSM). The FSM is timed, meaning that if a substate is not updated or evolved after a certain period of time, the timer automatically resets it to an earlier state to allow the resending of a previous request. To better understand how this complex code was developed, we will examine the code structure and security implementations in more detail in the following two subsections.

6.2.1 Software stack and libraries

In this section, we will briefly describe how the code structure was built. The software implementation was programmed along with the available NXP BMS controller code. Figure 6.2 shows the implemented software blocks and stacks. At the centre of the code lies the *Main State Machine*, which is contained in the code file ‘*main.c*’. The *BMS Diagnostic Functions* contains the main diagnostic and monitoring files responsible for collecting and processing the data from BPC. These files have been left essentially unchanged, except for some code enhancements in the ‘*diagnostic_handler.c*’ file.

The BMS controller communicates with the external gateway and other controllers through the files and libraries of *Communication*. The main file ’*comm_handler.c*’ contains the necessary functions for sending and receiving commands and encoding and decoding messages. The code for CAN and UART communication is extended to the initial code provided by NXP. The processing of the application takes place in the *Test Suite* block. Most functions are called from the *Gateway Handler*, a legacy name mainly intended for communication with the gateway, but also extended to handle application scripts for other communication tests. The *Security Handler* is responsible for performing security-related operations. Depending on the configuration, they can be executed either in software or hardware, which is described in more detail in Section 6.2.2. Communication with other controllers is handled by a separate *Secure Session* block, where both static and dynamic sessions can be executed. The important *Implicit Certificates* block contains functions for processing implicit certificates.

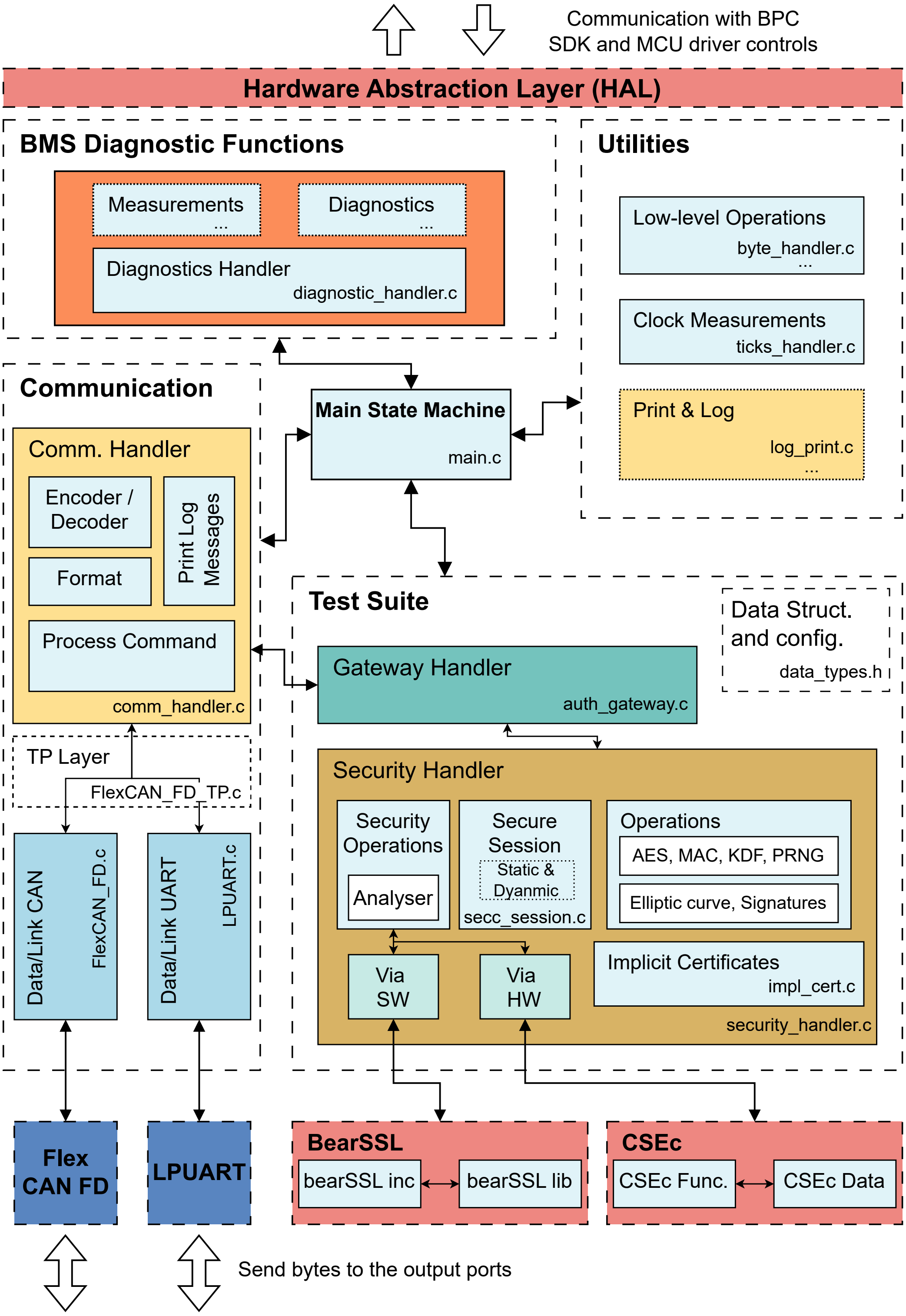


Figure 6.2: BMS test suite software stack. The blocks contain the names of either library, collection of files, or main functions. The most important code files are listed in the blocks. The application is run via the main.c file that contains an automated FSM. The Test Suite contains the application code, security handlers, and tests.

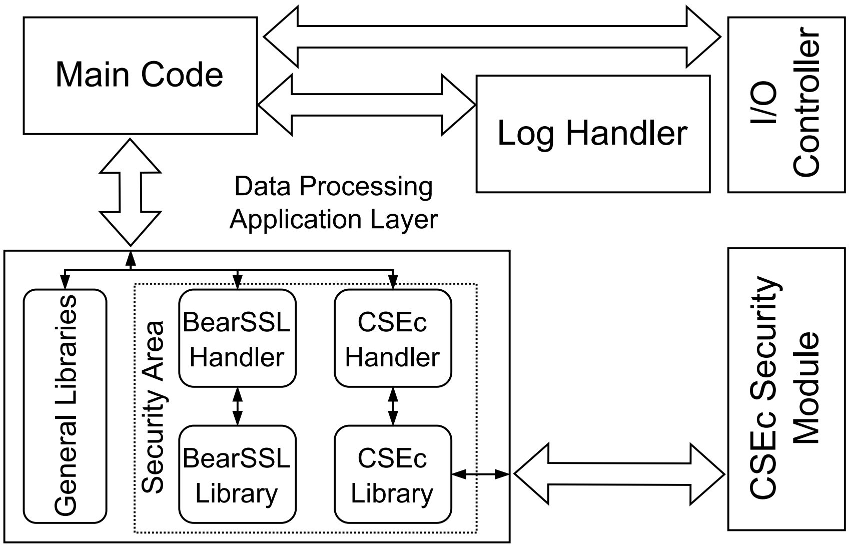


Figure 6.3: Implemented software stack for the security application use.

6.2.2 Security protocol implementations

Dedicated handlers have been implemented to enable the reliable use of security functions for the targeted tests and applications. For accuracy and security reasons, all underlying security operations come from trusted sources, either via software through the well-established and lightweight *BearSSL* library [241] or via hardware through the *Cryptographic Service Engine compressed (CSEc)* security co-processor provided on the S32K144 [242].

Central security wrapper. Depending on the test or application, different configurations can be applied to control the operations either between the BearSSL or the CSEc. From the developer's point of view, the same operations are used, except that the mechanism behind them is hidden using a wrapper, based on the principle of encapsulation [243]. The implemented blocks are shown in Figure 6.3.

Implicit certificate functions have been implemented directly in code using the BearSSL library and are observed as separate functions. BearSSL provides elliptic curve operations, with specific process steps implemented individually. To enable the operations, other auxiliary libraries also had to be implemented, such as the library for computing *big integers*. The decision for the hands-on implementation of ECQV functions was due to two main reasons:

- *Limitation 1*: During programming, finding open libraries that supported ECQV functions in the '*C*' language was challenging. It was crucial to have a library that was not only usable but also flexible for extensions and had proven functional accuracy.
- *Limitation 2*: The implementation of additional processing steps, e.g., the proposed authentication protocol from Section 5.2, and the STS-ECQV protocol proposed in Section 5.3 would be difficult to implement if a proprietary library was used.

The implemented code has been used for the overall BMS test suite security architecture, to maintain consistency with the already implemented functions. However, a separate ECQV implementation has been also done that relies on previous research and provided library from Pollicino et al. [130]. This library was used to provide the baseline functions for the comparison and evaluation between the compared ECQV key derivation and session establishment protocols. More on this in Section 7.2.2.

Using a secure module

For the hardware-accelerated security functions, we relied on the use of the *CSEc* security module found on the NXP's S32K1xx family of devices, i.e., on the S32K144 MCU used in the research prototype [242]. This security module implements the Secure Hardware Extension (SHE) specification [244].

The use of CSEc enables the secure generation and storage of symmetric cryptography keys and the faster and more secure execution of certain security functions. It also provides a real random number generator (RNG) that was useful for generating keys and nonces. However, the module only provides the use of symmetric cryptography, viz. AES for encryption and CMAC for computing the MAC tag. These operations are also limited to 128-bit keys. Due to the limited functionality many other functions, especially those related to asymmetric cryptography and implicit certificates, had to be implemented by using the BearSSL library mentioned above.

The functions are used through an implemented wrapper, as illustrated in Figure 6.3, which, depending on the internal configuration and availability of the target functions, is either run through the CSEc or passed to the implemented security functions that use the BearSSL handler.

6.3 Local Network

Under the local network, we consider a complex channel infrastructure between the BMS and the rest of the external system to replicate a communications environment found under a local area network (LAN), e.g., inside a vehicle on the internal central bus. The following elements are considered when designing the implementation:

1. *Secure gateway:* uses a Raspberry Pi 4, as a central device. The "gateway" program is done in the programming language *Python* to support extended functionalities and also the connection with a cloud system.
2. *EVCC:* is used to represent a controller responsible for the connection establishment between the BMS and a charging station. The software is based on the same stack and functional implementation as on the main BMS controller, the difference being only in the messages sent and ordered sequence for the session key derivation and thereafter communication. Hence, the EVCC is also represented by an S32K144 MCU board.
3. *Battery emulator:* it is also part of the local network, through the communication with the main BMS controller that also acts as a wrapper for the remainder of the BMS sub-system. Its full implementation has been previously described in Section 6.2.
4. *Communication channels:* the local communication between the three main devices works over an implemented network stack, specifically over stacks run on the UART and CAN lines.

The local area network is represented by the use of two different protocol connections: (i) a serial UART connection for communication between BMS and other control units with the secure gateway, and (ii) CAN for intra-module communication between BMS and control units, i.e., ECU. In the dissertation investigation, we have focused on developing a scalable service architecture using the secure gateway. The intent was to make request processing independent of the device, but allow for extensions to accommodate the intended BMS use cases and test applications.

A brief description is given in the following two subsections to better summarize the network design and stack as well as the gateway script.

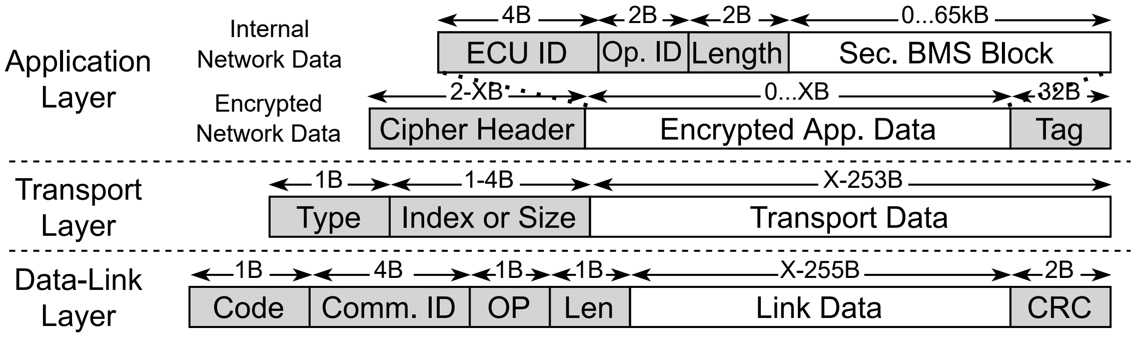


Figure 6.4: Developed frame and packet structures for the communication layers used for the intra-network communication between the BMS controller and the central gateway. Adapted from *Publication H.*

6.3.1 Communication architecture

The test suite relies on the use of different network communication layers running over two physical communication standards, UART and CAN. The UART communication is intended for use between the secure gateway and another, e.g. BMS, controller, while CAN is used between different ECU; in our case, these are BMS and EVCC. The data packet structures used on each layer are shown in Figure 6.4 for the UART stack, where Figure 6.5 shows the data structures on layers using the CAN protocol. The actual layer fields can be tailored to meet the needs of the target system, i.e., they are designed to be implementation flexible to accommodate other potential communication standards.

Serial connection

From Figure 6.4, the application layer holds the main custom packet that is application specific. The secure BMS block is included as the payload to the generally agreed application packet structure on the internal network level. This payload is to be encrypted and together with the added header that contains ECU ID (4 bytes), Op. ID (2 bytes) and Length (2 bytes) is MAC-ed and added as the trailing Tag for integrity protection. The internal network then mainly communicates with these secure packets that are received and decrypted on the symmetric protocol level. The transport layer helps in the fragmentation, i.e., when the payload length is longer than what can be sent with the custom data-link packet structure decoded on the physical receiver. In our test case, the data-link layer packet can only hold up to 255 bytes in one packet, hence, there is a necessity to use the additional transport layer. The transport layer is modelled after the ISO 15765-2 standard used also for the CAN networks [245]. The transport data can be up to 2^{32} bits in length (or up to 4GB), but the actual transport data field is up to 253B long to accommodate the underlying data-link layer, i.e., multiple packets must be formed in case the data is longer than link-layer data field length.

As mentioned above, communication is based on the use of the UART protocol, more precisely, the low power UART (LPUART) protocol version. The baud rate is set to 57600 baud, i.e., 57.6 kBit/s, and a standard communication configuration is used otherwise. To avoid noisy readouts, the data is checked only after receiving the first 7 bytes (header of the created 'data link' frame), after which the content is analyzed for the correct communication code and the device ID. The communication is 'blocking' and tailored to the specifications of the application and the FSM controlled by the main functions. This means that readout occurs at specific times. Timeout handlers are responsible for resending certain frames if the previous ones were not received completely or correctly.

CAN connection

To model an accurate automotive communication environment, physical communication standard CAN was utilized for communication between the BMS and any other adjacent ECU. For the evaluation, it was used to test the proposed derivation of the secure communication session (from Section 5.3) between the main BMS controller and the EVCC. The focus on the BMS and the EVCC was inspired by recent research by Fuchs et al. [53]. Specifically, we have relied on the use of the Controller Area Network Flexible Data-Rate (CAN-FD) protocol, an extended specification of the original CAN protocol that is becoming increasingly popular today. The CAN-FD protocol offers a higher data rate and payload size compared to the original CAN protocol, among other advantages.

The structure of the data packets is shown in Figure 6.5, where the application layer is designed to be lightweight and used for intra-module session communication. During the derivation of the session key, the data can be sent in plain text. After the keys are derived, the '*Application Data*' field may also contain an encrypted message structure very similar to that shown in Figure 6.4.

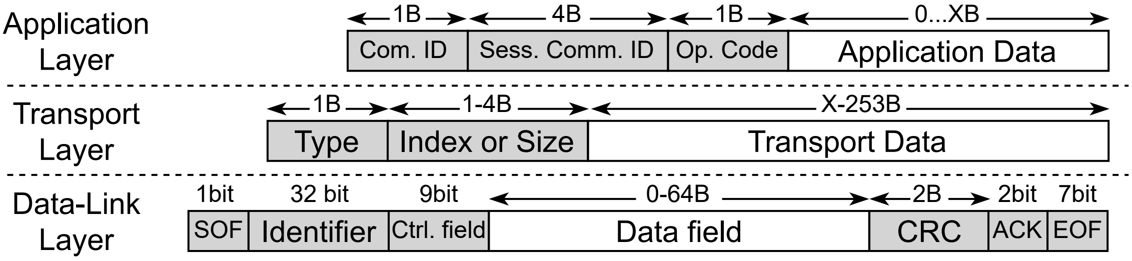


Figure 6.5: Developed CAN(-FD) protocol communication stack for testing the intra-module communication between BMS and EVCC. Extended from *Publication F.*

6.3.2 Gateway protocol administration

The secure gateway is a device responsible for the control of the network, authentication of the devices, CA for implicit certificates, and edge point for the control with the outside, i.e., with cloud services. To allow these functions, we developed a script in Python for Raspberry Pi 4.

The communication control from the gateway is based on the operational codes found at the beginning of the header of the data-link frame. From Figure 6.4, we can observer the following fields:

- *Code:* preamble research prototype code with a fixed value of 0x53. 1 byte long.
- *Communication ID:* unique device serial link communication ID. 4 bytes long.
- *Operation:* code for the operation, i.e., the message content. 1 byte long.
- *Length:* total size of the frame payload data in bytes. 1 byte long.
- *CRC:* 16-bit error detection code. 2 bytes long.

The codes are interpreted using a FSM. In order for the request to be processed, each field must be confirmed, otherwise, an error message is sent as a response. This process is shown in Figure 6.6. Each action is identified by a different operational code. Each code contains only 1 byte, with the most significant 4 bits containing the operation class and the least significant bits holding the sub-identifier. The implemented operation codes, and with that the operations themselves, are listed in Table 6.1.

The secure gateway is instructed to open a new running thread on each new request session, i.e., either for authentication or session communication. In the setup used in the dissertation, these come from the BMS and EVCC devices, but the architecture is flexible to accompany any other ECU.

Table 6.1: Operational codes implemented and used with the BMS test suite.

Action	Text class	Hex class	Description
Device authentication	INIT_DEV_AUTH_{*step*}	0x0{*i*}	Initial device authentication
Certification	INIT_CERT_{*step*}	0x1{*i*}	CA certificate activities
Static session	SESS_STC_{*step*}	0x2{*i*}	For establishing static session
Dynamic session	SESS_DYNM_{*step*}	0x3{*i*}	For establishing dynamic session
Logging & cloud	GW_SESS_{*step*}	0x5{*i*}	Processing BMS log data

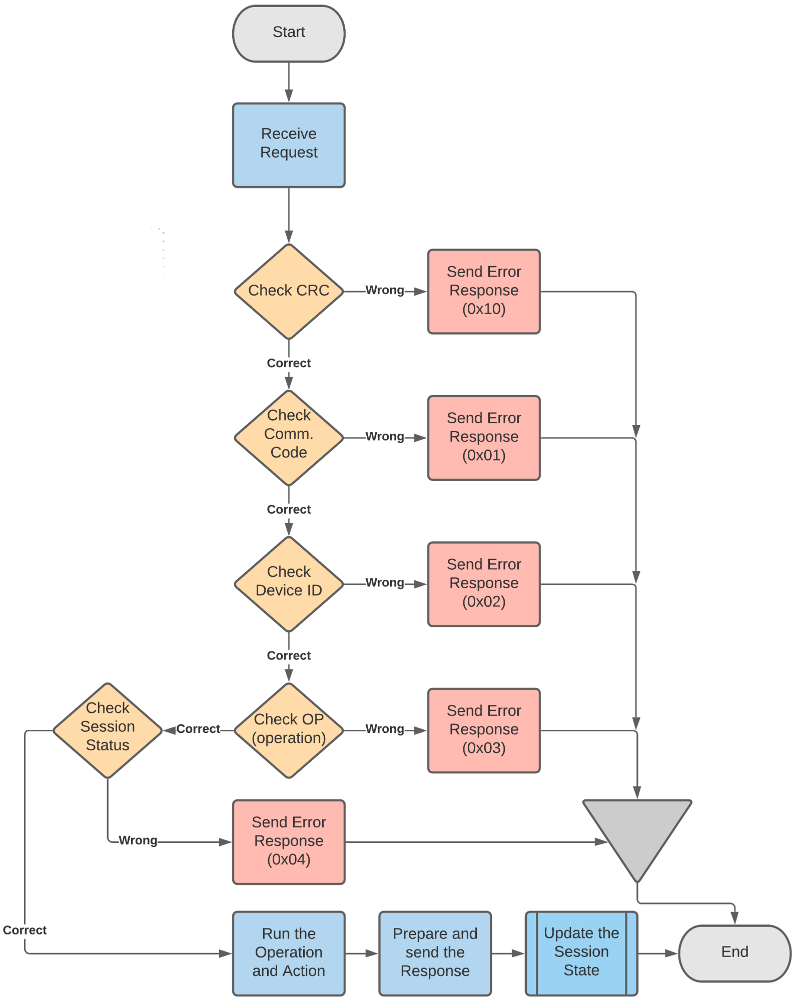

Start
Receive Request
Check CRC
Wrong
Send Error Response (0x10)
Correct
Check Comm. Code
Wrong
Send Error Response (0x01)
Correct
Check Device ID
Wrong
Send Error Response (0x02)
Correct
Check Session Status
Correct
Check OP (operation)
Wrong
Send Error Response (0x03)
Wrong
Send Error Response (0x04)
Correct
Run the Operation and Action
Prepare and send the Response
Update the Session State
End

Figure 6.6: Secure gateway process handler flowchart: (i) the gateway receives a request, (ii) each field is checked individually with corresponding error messages sent in case of an error, (iii) on success, the action is processed, (iv) an appropriate response is generated using the same format as the request and sent back to the device (BMS or EVCC), (v) the state of the FSM is incremented with a new value.

6.4 NFC Readout Elements

To test the usability of wireless BMS readout under the use cases previously specified in Chapter 4, appropriate NFC devices were integrated. The components used could not be just any NFC elements. They had to be automotive-graded, and operational with respect to the specified functional and security requirements. To this end, we resorted to using the new NTAG5 link and boost components from NXP Semiconductors [246].

Specifically, for the test suite, the following NFC devices have been implemented:

1. *NFC reader*: an NCx3320 chip and board have been used, together with an additional S32K144 MCU to provide control over the reader.
2. *NFC tag*: two NCx3310 NTAG have been used; one for the external readout and the other for the temperature sensor readout simulation.
3. *External reader*: a smartphone, Motorola Moto X, was used to test the external active and idle diagnostic readouts.

An important point here to discuss is the use of the separate auxiliary MCU controller to function as an extension to the already existing BPC. As per the proposed design from Chapter 4, both the NFC reader and the NTAG should be directly attached and controlled by the BPC. Since the BPC in the prototype is an integrated application-specific integrated circuit (ASIC), an additional controller had to be used for the development and testing of the NFC functionality. However, this NFC controller still communicates with the BPC by sending the read data through a *low-pass filter* and acts as the analog input instead of the one from the battery cell emulator. The low-pass filter conversion is done to convert from the otherwise digital signals processed on the MCU to the analog values read by the BPC. This is done to bypass the standard "wired" connection and still allow for a full test suite analysis with regards to the NFC components. It is important to note that the MCU modifications for transfer over the BPC are intended for the *'Active sensor'* use case, as for the other two diagnostic use cases, the readout is carried out externally using a mobile device directly from the NTAG of the auxiliary controller.

Elements of the NFC readout components integration have been also developed during the turn of two master theses, with the initial setup by Gärtner [214], and later updated by Laube [215].

6.5 Remote Services

To fully test the secure encapsulation of the BMS block's data and transmission from its battery cell sensor source to the final end-system, it was necessary to extend the simulation to also support remote services. Specifically, a connection was established with a cloud service that would then be able to collect and process the data to a user's end-system device.

The test suite was extended with the following components that consider remote services:

1. *Gateway device*: the secure gateway, i.e., a Raspberry Pi 4, described in Section 6.3.2, is responsible for preparing the received BMS secure log data and forwarding it to the cloud.
2. *Cloud service:* the processing is done over an Amazon Web Service (AWS) cloud instance, as one of the bigger and more reliable cloud services present at the time of this writing.
3. *End-system:* for the purpose of receiving and visualizing the received BMS data, another Raspberry Pi 4 was employed that is connected to the network. Here, a script was implemented using *Python.* It is set up as a server using the *Flask* library.

Communication between the gateway and the AWS cloud runs over the secure MQTT protocol. The MQTT is a standard protocol used for machine-to-machine communication and transmission of frequently sampled data [240]. Security is provided by certificates and keys derived from the TLS protocol. After receiving new data, the AWS further processes it and sends it to the end-system by relying on the HyperText Transfer Protocol Secure (HTTPS) protocol for secure and reliable transmission. Since the data blocks are encrypted, the cloud only knows the destination of the user, with their privacy being protected. The updating of the data is based on the *shadow principle*. The AWS cloud maintains a shadow instance of the gateway for each user assigned to it. As soon as a change occurs, i.e., new data is detected, it is automatically transmitted to the target end-system. The main task of the end-system is to process the received BMS block by decrypting and extracting the required data, which is afterwards displayed in a real-time graph.

CHAPTER 7

Evaluation

Summary: *In this chapter, the evaluation is presented for the solutions contributed by this dissertation. Namely, the evaluation is done on the basis of the presented three research questions from Chapter 1, i.e., divided on each different BMS layer of communication, starting with the internal BMS sub-system, followed with BMS interaction on the local network, and finally, the evaluation regarding the full secure BMS data propagation from the source to the end cloud system. The evaluation relays on the implementation setup described in Chapter 6 with additional analytical and practical evaluation steps. Each layer is analyzed on the important performance and security evaluation points.*

◇◇◇

7.1 Secure BMS NFC Wireless Readout

We start the evaluation by analyzing the proposed wireless and secure design for the BMS sub-system from Chapter 4. Specifically, we take a look at the internal active sensor readout and external diagnostic readout use cases and analyze them on the performance, overhead, and achieved security.

7.1.1 Performance evaluation

The evaluation was done using real hardware as part of the test suite from Section 6.4. For accuracy, it was important to rely on the NFC components that are NFC-forum compliant and automotive-graded.

BMS NFC internal readout

Figure 7.1 shows the performance evaluation for the BMS NFC active sensor readout use case. We divide the process into two main operational phases:

1. *Initialization*: it is run during the first NTAG discovery process. It starts with the security authentication step ($369.3 \pm 0.4\,ms$), followed by the energy harvesting setup ($19.6 \pm 0.3\,ms$), NTAG initialization setup ($29.2 \pm 2.4\,ms$), and sensor initial configuration ($116.1 \pm 1.2\,ms$).
2. *Monitoring*: during this phase, the sensor measurement readout is continuously triggered. Each full readout and transfer takes $27.2 \pm 0.5\,ms$.

The majority of the time is spent on the authentication phase. However, this step only needs to be performed once during the initial device discovery and configuration. It is also highly hardware and software dependent and could be optimized for an integrated design. We can conclude that the reported time is sufficient for BMS applications where the sampling time for one sensor is higher than $30\,ms$.

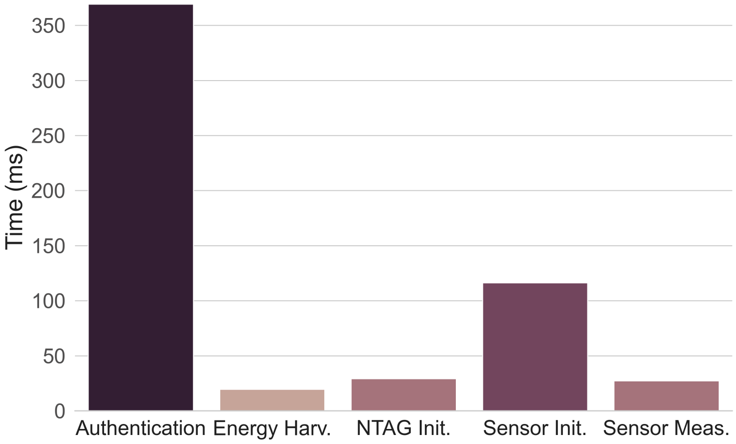


Figure 7.1: Performance evaluation for the BMS NFC active sensor readout use case. Adapted from *Publication C*.

Sensor throughput analysis. The throughput for the active sensor readout use case depends on the number of sensors per battery pack module and the number of BPC and their readers. For our use case, we will assume one reader per battery pack module with one sensor per battery cell pack. During the monitoring phase, we are able to perform 33 measurements. After optimization, each sensor readout requires only $8\,B$, resulting in a throughput of $264\,B/s$. I.e., for one thousand iterations, this would result in $\approx 29.24\,s$ and a throughput of $7.54\,kB$.

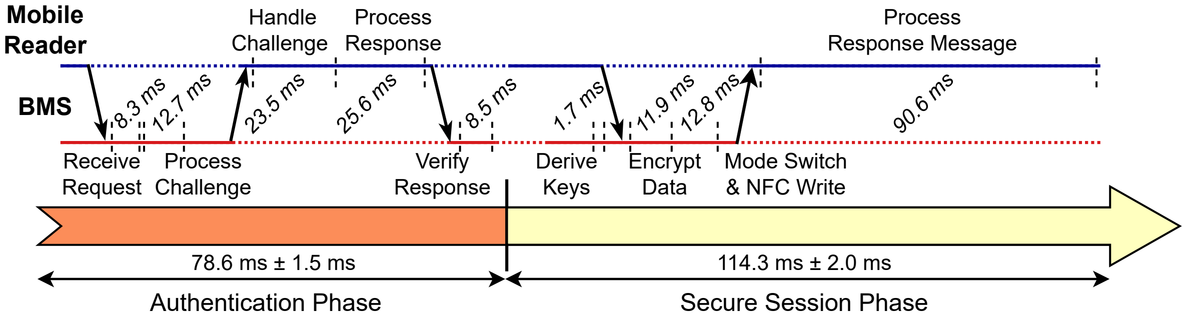


Figure 7.2: Diagram showcasing the timeline for the BMS external readout utilizing the proposed NFC design.

BMS NFC external readout

The external readout was tested using a custom mobile application and an NTAG connected to the emulated BPC. We analyze the security protocol introduced in Section 4.3.2 under the authentication and session phases to simulate a diagnostic readout. The setup considers an ideal NFC antenna positioning and uses hardware-optimized security AES-CBC+CMAC functions. Figure 7.2 shows the full timeline of the evaluation process. Most of the time is spent on the integrated NFC read & write operations. The reported performance time is deemed sufficient for the current BMS readout requirements.

Protocol overhead. The current design uses 16 bytes for the authentication from the mobile reader side, and another 32 bytes from BMS or BPC MCU as a reply. Session application data is sent in SNDEF records. Each record is fixed with a total size of 234 bytes, from which 8 bytes are contained in the record header. The remaining 226 bytes are further divided into 34 bytes of public data consisting of cipher specification, IV, and a tag, and 192 bytes of secret data from which 182 are left for the application. For an exchange that would include a larger data transfer, fragmentation would need to be included.

BMS NFC Wake-up Analysis

The wake-up process relies on the use of energy harvesting to power the passive device, i.e., NTAG. We observe two models as described in Section 4.2.4. Figure 7.3 shows the designed evaluation schematic with different connections for EDW and EHW models. In real hardware development, a level shifter can be used to avoid possible cross currents between the I2C, the return signal port, V_{cc} & V_{out}. The wake-up process depends on the distance between the reader and the NTAG. In our setup, the distance reaches a peak value of 5.4 cm. It was found that a distance of 2 cm between the active reader and the passive tag device is recommended for feasible and reliable communication.

For the devices used, the theoretical wake-up power consumption for the EDW model is $117.81\,\mu W$, while for EHW it is $98.3\,\mu W$. In both cases, the BPC (S32K144) needs to wake up, which requires $1.9\,ms$ for a full startup in normal working clock mode. Since in EHW model, the NTAG is completely powered-off before the wake-up process, it takes slightly longer than the EDW model to reach the trigger interrupt condition. Specifically, to reach the $2.145\,V$ value required for the interrupt, the EDW model requires $1\,ms$ and the EHW model requires $1.9\,ms$. Compared to the current wireless BMS SotA, where Rincon Vija et al. [247] demonstrated a wake-up model for BMS using BLE, this is a significant improvement as NFC enables a fast and energy-efficient wake-up process, which is essential for second-life diagnostic readout applications. The BLE not only takes more time to wake up with the reported $500\,ms$, but is also less flexible for the proposed use cases and incurs higher production costs.

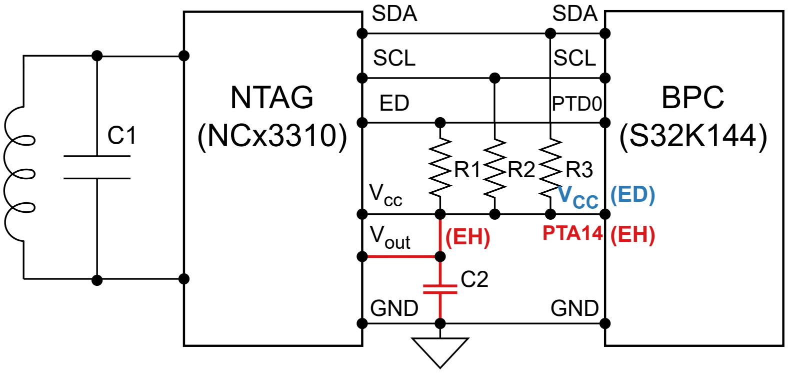


Figure 7.3: Design schematic for evaluating the proposed NFC wake-up models. Adapted from *Publication G.*

7.1.2 Security evaluation

The security evaluation is based on the threat model analysis to evaluate the achieved protection against the common threats [112, 113], and STRIDE classification model [114]. The analysis follows the security requirements and threats described in Section 4.1.2. We list *assets (A)*, *threats (T)*, *countermeasures (C)*, and potential *residual risks (R)*. The same analysis method is also conducted for the other sections of the evaluation. For the BMS NFC design, we separate the security analysis on the use cases, i.e., on the proposed internal and external readout design. We additionally conduct a formal security analysis on the proposed BMS NFC security protocol to further support our claims.

Security model analysis for internal BMS NFC readout

For the internal readout, we argue that every attached pack is deemed untrustworthy before verifying and that no device possesses hardware and software design vulnerabilities. In the internal communication, we want to protect the following assets: (A1) *sensor data*, (A2) *system integrity*, and (A3) *diagnostic data.* The attack surface is modeled after a DFD shown in Figure 7.4.

We list the following threats⟨*with STRIDE categories*⟩, assets, and provided countermeasures:

[T1]⟨S,D⟩ *Battery control obstruction* ↦ (A1), (A3): false sensor and diagnostic data can lead to the disruption of the standard BMS operations.

> **(C1)** *Battery pack authentication model*: by using the authentication model from Section 4.3.1, we are able to validate and protect from malicious devices gaining system access.

[T2]⟨S,T,D⟩ *BMS status tampering* ↦ (A2), (A3): similar to **[T1]**, but considers malicious change of the original message content.

> **(C1)** *Battery pack authentication model*: the same protection mechanism as for **[T1]**.

[T3]⟨S,T,R,I⟩ *Opening a backdoor* ↦ (A1), (A2): an attacker might gain direct system access through an exposed interface or a counterfeited device.

> **(C3)** *NFC technology characteristics*: short range and exclusive band makes remote attacks difficult, along with **(C1)** protecting against counterfeited devices.

[T4]⟨T,I⟩ *Remote channel attacks* ↦ (A1): can be in the form of passive eavesdropping or active MitM attacks targeting the NFC channel.

> **(C2)** *Battery pack sealing*: the battery pack is enclosed in a metal chassis and therefore impervious to outside attacks. **(C3)** also hampers the possibility of this type of attack.

[T5]⟨T,R,I⟩ *BPC data leak* ↦ (A1), (A3): if an attacker can gain access either through **[T3]**, **[T4]** or some other form, they would be able to compromise the data processed on the BPC.

> **(C4)** *Data & device security*: By providing optional security operations and relying on the secure hardware design. Otherwise, the mitigation is difficult and not covered by the original design.

Security model analysis for external BMS NFC readout

The security analysis for the external BMS readout is directly tied with the security protocol for the authentication and secure session derivation, formally analyzed in the next section. However, the full security design model must also consider threats that may arise from system-side vulnerabilities. We performed a security analysis similar to the internal BMS readout by listing threats and countermeasures. For better illustration, the results are shown in Figure 7.5 using a GSN. Here we can see that we aim to protect (A1) the transmitted BMS data and (A2) system material in the form of configuration data. (C1) & (C2) come from the developed authentication protocol, (C3) is based on the internal configuration, (C4) is included in the SNDEF structure, and (C5), (C6), (C7) are design protection mechanisms. We do note that the proposed design has difficulty with DoS attacks, but these would be rare due to the NFC short-range property. On the other hand, there is no direct protection against side-channel attacks, and these would need to be mitigated by proper design controls.

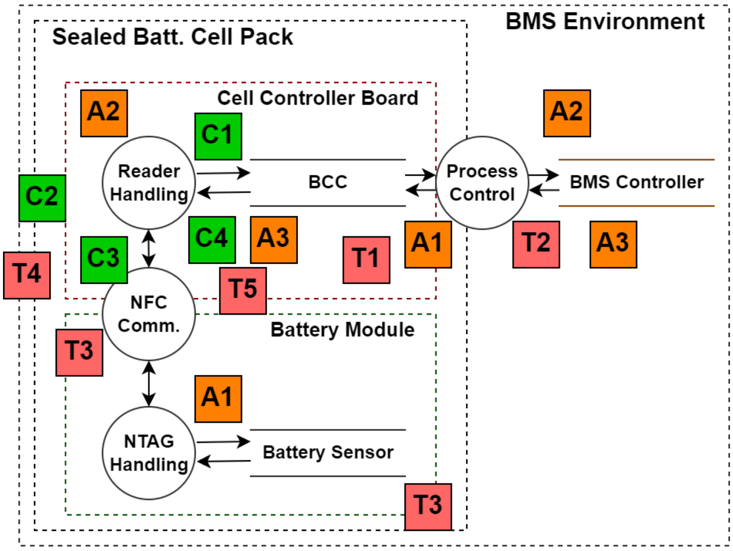


Figure 7.4: Data flow diagram security model for the internal BMS NFC readout. Adapted from *Publication C.*

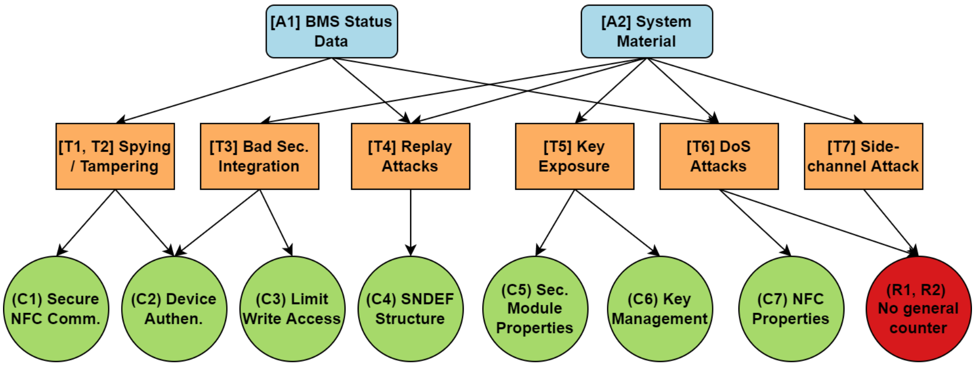


Figure 7.5: GSN security model for the external BMS NFC readout. Adapted from *Publication D.*

Formal analysis of the NFC mutual authentication protocol

For the analysis of the symmetric security protocol for authentication and session key derivation proposed in Section 4.3.2, we conducted a formal security evaluation using BAN logic [119]. It is a formal model based on knowledge and belief [248, 249, 250, 251]. The analysis is based on setting up a specific hypothesis that we want to prove by using the logic formulae shown in Table 7.1 [251, 235]. Based on the hypothesis, goals are set that are then verified using BAN-logic postulates, i.e., the inference rules.

An inference rule is defined as follows, where it is stated that Y holds as long as $X_1 ... X_n$ hold:

$$\frac{X_1, ..., X_n}{Y} \tag{7.1}$$

There are several defined inference rules, but in this analysis, we will be relying on the following:

- *Message-meaning rule*

$$\frac{M \mid\equiv N \overset{K}{\leftrightarrow} M, N_R \triangleleft \{X\}_K}{M \mid\equiv N \mid\sim X} \tag{7.2}$$

Table 7.1: BAN-logic formulae.

Formula	Description
$M \mid\equiv X$	M believes and trusts X; the principal logic rule
$M \triangleleft X$	M sees and has received a message X
$M \mid\sim X$	M once said X at present or past; it believed X at that point
$M \Rightarrow X$	M has jurisdiction and can delegate over X
$\#(X)$	X message is fresh; it has never appeared in the past
$\langle X \rangle_Y$	Message X is combined with Y; Y is a secret that proves the origin of X
$\{X\}_K$	Message X is encrypted with key K
$M \overset{K}{\leftrightarrow} N$	M and N use the shared key K for secret communication
$\overset{K}{\mapsto} M$	K and K^{-1} are the private and public keys of M respectively
$M \overset{X}{\rightleftharpoons} N$	The secret formula is only known to M and N

- *Nonce-verification rule*

$$\frac{M \mid\equiv \#(X), M \mid\equiv N \mid\sim X}{M \mid\equiv N \mid\equiv X} \tag{7.3}$$

- *Belief rule*

$$\frac{M \mid\equiv X, M \mid\equiv Y}{M \mid\equiv (X,Y)} \quad \text{or} \quad \frac{M \mid\equiv (X,Y)}{M \mid\equiv X} \quad \text{or} \quad \frac{M \mid\equiv N \mid\equiv (X,Y)}{M \mid\equiv N \mid\equiv X} \tag{7.4}$$

- *Freshness rule*

$$\frac{M \mid\equiv \#(X)}{M \mid\equiv \#(X,Y)} \tag{7.5}$$

Idealized protocol. We prepare the BAN form based on the protocol definition from Section 4.3.2:

$$1)\ all\ \ plaintext \tag{7.6}$$

$$2)\ M_N \rightarrow N_R : \{\{ch_r, N_R \overset{K_M}{\longleftrightarrow} M_N\}_{K_M}\}_{K_M} \tag{7.7}$$

$$3)\ N_R \rightarrow M_N : \{\{ch_t, N_R \overset{K_M}{\longleftrightarrow} M_N\}_{K_M}\}_{K_M} \tag{7.8}$$

$$4)\ M_N \rightarrow N_R : \{X, N_R \overset{K_S}{\longleftrightarrow} M_N\}_{K_S} \tag{7.9}$$

$$5)\ N_R \rightarrow M_N : \{X', N_R \overset{K_S}{\longleftrightarrow} M_N\}_{K_S} \tag{7.10}$$

Assumptions. Based on the protocol description, the following assumptions are made that are upheld for the full logic arguments.

Firstly, both sides consider that the sent nonces are fresh:

$$N_R \mid\equiv \#(ch_r) \tag{7.11}$$

$$M_N \mid\equiv \#(ch_t) \tag{7.12}$$

Secondly, both sides believe that they possess and use the shared master key:

$$N_R \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N \tag{7.13}$$

$$M_N \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N \tag{7.14}$$

Thirdly, additional assumptions need to be made that are concerned with the freshness of the messages. Since messages X and X' from Eq. 7.9 and Eq. 7.10 respectively are partially composed out of the nonces ch_r and ch_t, it is assumed by the *freshness rule* that:

$$\frac{N_R \mid\equiv \#(ch_r)}{N_R \mid\equiv \#(X)} \tag{7.15}$$

$$\frac{M_N \mid\equiv \#(ch_t)}{M_N \mid\equiv \#(X')} \tag{7.16}$$

Goals. The verification of the BAN logic follows proving the set goals. At the end of the presented protocol, it needs to be made sure that both parties are mutually authenticated and that each side both knows and trusts that knowledge. The following first-order goals are derived:

$$G1.1)\;\; N_R \mid\equiv M_N \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N \tag{7.17}$$

$$G1.2)\;\; M_N \mid\equiv N_R \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N \tag{7.18}$$

For the second-order goals, it needs to be made sure that both parties have their communication keys correctly derived and that each side believes that the other side has done so as well:

$$G2.1)\;\; N_R \mid\equiv M_N \mid\equiv N_R \overset{K_S}{\longleftrightarrow} M_N \tag{7.19}$$

$$G2.2)\;\; M_N \mid\equiv N_R \mid\equiv N_R \overset{K_S}{\longleftrightarrow} M_N \tag{7.20}$$

Verification. The verification will start with proving the first-order goals $G1.1$ and $G1.2$. First, the *message-meaning rule* (Eq. 7.2) is applied on the Eq. 7.7:

$$\frac{N_R \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N, N_R \triangleleft \{\{ch_r\}_{K_M}\}_{K_M}}{N_R \mid\equiv M_N \mid\sim (ch_r,\ N_R \overset{K_M}{\longleftrightarrow} M_N)} \tag{7.21}$$

The *freshness rule* (Eq. 7.5) is used to comply with the freshness of the master key and nonce, and the previous assumption that the challenge nonce is fresh (Eq. 7.11):

$$\frac{\#(ch_r)}{\#(ch_r, N_R \overset{K_M}{\longleftrightarrow} M_N)} \tag{7.22}$$

Next, using the *nonce-verification rule* (Eq. 7.3) on the previous two statements 7.21 & 7.22 yields the following statement:

$$\frac{N_R \mid\equiv \#(ch_r, N_R \overset{K_M}{\longleftrightarrow} M_N), N_R \mid\equiv M_N \mid\sim (ch_r, N_R \overset{K_M}{\longleftrightarrow} M_N)}{N_R \mid\equiv M_N \mid\equiv (ch_r,\ N_R \overset{K_M}{\longleftrightarrow} M_N)} \tag{7.23}$$

Finally, the *belief rule* (Eq. 7.4) is used on the statement 7.23 to verify the goal $G1.1$ (7.17):

$$\frac{N_R \mid\equiv M_N \mid\equiv (ch_r,\ N_R \overset{K_M}{\longleftrightarrow} M_N)}{N_R \mid\equiv M_N \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N} \tag{7.24}$$

The verification of the $G1.2$ (7.18) is symmetrical to the verification of the goal $G1.1$. Thus, both first-order goals $G1.1$ (Eq. 7.17) and $G1.2$ (Eq. 7.18) are proved. For proving the second-order goals, it is necessary to rely on the 7.15 and 7.16 assumptions concerning the freshness of the nonces. Then, using the *belief rule*, the important statements are derived that both sides believe the session key possession:

$$\frac{N_R \mid\equiv (X,\ N_R \overset{K_S}{\longleftrightarrow} M_N)}{N_R \mid\equiv N_R \overset{K_S}{\longleftrightarrow} M_N} \tag{7.25}$$

$$\frac{M_N \mid\equiv (X',\ N_R \overset{K_S}{\longleftrightarrow} M_N)}{M_N \mid\equiv N_R \overset{K_S}{\longleftrightarrow} M_N} \tag{7.26}$$

The remainder of the proof for goals $G2.1$ and $G2.2$ follows the same line of thinking as with proving goals $G1.1$ and $G1.2$, i.e., using the same postulates. The difference is that instead of proving through the master key K_M, the equations are set for the session key K_S. By using the *message-meaning rule*, followed with the *freshness*, *nonce-verification*, and finally the *belief rule* on Eq. 7.9, the following statements are derived:

$$\frac{N_R \mid\equiv N_R \overset{K_S}{\longleftrightarrow} M_N, N_R \triangleleft \{X,\ N_R \overset{K_S}{\longleftrightarrow} M_N\}_{K_S}}{N_R \mid\equiv M_N \mid\sim (X,\ N_R \overset{K_S}{\longleftrightarrow} M_N)} \tag{7.27}$$

$$\frac{N_R \mid\equiv \#(X)}{N_R \mid\equiv \#(X, N_R \overset{K_S}{\longleftrightarrow} M_N)} \tag{7.28}$$

$$\frac{N_R \mid\equiv \#(X, N_R \overset{K_S}{\longleftrightarrow} M_N), N_R \mid\equiv M_N \mid\sim (X, N_R \overset{K_S}{\longleftrightarrow} M_N)}{N_R \mid\equiv M_N \mid\equiv (X,\ N_R \overset{K_M}{\longleftrightarrow} M_N)} \tag{7.29}$$

$$\frac{N_R \mid\equiv M_N \mid\equiv (X, N_R \overset{K_S}{\longleftrightarrow} M_N)}{N_R \mid\equiv M_N \mid\equiv N_R \overset{K_S}{\longleftrightarrow} M_N} \tag{7.30}$$

With this, the goal $G2.1$ is verified. The goal $G2.2$ is symmetrical to $G2.1$ and, thus, is also verified.

7.2 Security Architecture for BMS

The evaluation of the proposed security architecture is divided into the performance analysis of the proposed security design and protocols, as well as the security threat analysis. We start the performance analysis by observing the first two operational cycles, i.e., device authentication and certificate derivation, followed by the secure session analysis in terms of key derivation protocols, and finally the security evaluation concerning the full deployment of the local network BMS security architecture.

7.2.1 Performance evaluation of the BMS device authentication

In this section, we analyze the performance over time execution of the security architecture by considering the device authentication and implicit certificate deployment cycles described in Sections 5.2.1 and 5.2.2 respectively. The analysis is based on the test suite implementation described in the previous Chapter 6. In particular, we look at the functional steps that include communication to and from the BMS sub-system (represented by the S32K144 running in normal mode with a 32-bit Cortex M4 and 80 MHz core clock) and the gateway device (Raspberry Pi 4 with 4 cores 64-bit 1.5 GHz). The analysis on both devices focuses on the most important individual processing steps.

Table 7.2: BMS time measurements of the essential operational cycles and processing steps.

	BMS (S32K144) Cycle and Process	Time (ms)
Device Authentication	1.1 Request preparation for the gateway	12.6 ± 0.1
	1.3 Received challenge handling & response derivation	32.6 ± 0.1
	1.5 Closing configuration & key update	5.1 ± 0
Certificate Derivation	2.1 Implicit certificate request preparation	651.3 ± 1.3
	2.3 Public key derivation	936.4 ± 5.4

Table 7.3: Gateway time measurements of the essential operational cycles and processing steps.

	Gateway (Raspberry Pi 4) Cycle & Process	Time (ms)
Device Authentication	1.2 Request processing from BMS	119.6 ± 3.3
	1.4 Response verification from BMS	7.2 ± 0.2
Certificate Derivation	2.2 Request handling & certification calculation	238.4 ± 6.4
	2.4 Receiving configuration acknowledgment	3.0 ± 0.1

The analysis measured the total execution time for each essential processing step with the experimental results presented in two individual tables. The processing steps are numbered based on the sequence of their associated protocol, i.e., of the deployment cycle, between the two tables. Based on the results, we make the following conclusions:

- *BMS time measurements* (in Table 7.2). The device authentication steps show a relatively fast time for deriving and sending the request and for the final step of updating the key, with most of the time spent preparing the challenge and the response due to the encryption and MAC algorithms. The steps dealing with the implicit certificate, i.e. the derivation of the public key, are the most time-demanding since the security operations involving the EC operation were performed in the pure BearSSL library without hardware acceleration. It would be possible to optimize them by using specially designed security modules and devices.
- *Gateway time measurements* (in Table 7.3). In contrast to the BMS controller, there is less difference between the runtime of device authentication and the certificate derivation. This is also

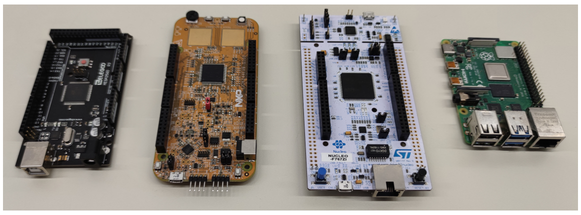


Figure 7.6: Key derivation evaluation devices (left to right): ATMega256, S32K144, STM32F767, Rasp. Pi 4.

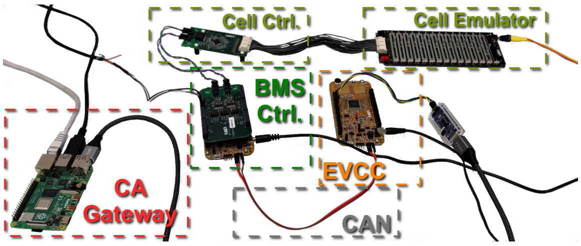


Figure 7.7: Test suite for the secure session establishment evaluation. Adapted from *Publication F*.

due to the different responsibilities and tasks performed on the gateway device. The most time-consuming tasks were the creation of the initial challenge message during device authentication and the certificate derivation calculation step during the cycle of the same name.

While it can be concluded from the analysis that the certificate derivation cycle is much slower than the device authentication cycle, one might wonder why the entire authentication step is not replaced by the proposed device authentication protocol. The problem remains with authentication between a larger number of devices, which adds significant overhead to both the storage and the update mechanism. In this case, all network devices, including the BMS would rely only on symmetric cryptography and would need to update and keep track of each master key. This would also introduce security issues related to node-capturing attacks because if one network device and its keys are compromised, it would affect the entire network. Using the certificate derivation phase is performance intensive, but avoids the functional complexity related to key updates and potential security vulnerabilities.

7.2.2 Performance evaluation of the secure session key derivation

In Section 5.3, we introduced two key derivation protocols as part of the BMS security architecture. We compare them with each other, but also with two other SotA implicit certificate key derivation protocols. Specifically, we compare the proposed SKD [233] as ‘S-ECDSA’ (+ extended ack. step as in Porambage et al. [197]), the DKD [234] as ‘STS’ alongside its two optimized variants, from Porambage et al. [197] as ‘PORAMB’, and from Sciancalepore et al. [200] as ‘SCIANC’. For the first part of the analysis, we evaluate the performances on the protocol basis, whereas, in the second part, we run the S-ECDSA and STS protocols on the BMS test suite and compare them on a real CAN communication.

Evaluation of the performance and overhead of the protocols

We implemented security protocols in pure ‘*C*’ using the *micro-ecc*, *tiny-aes*, and *bear-ssl* libraries, but also the provided library from Pollicino et al. [130] for the accurate ECQV functions. To show the feasibility across different ranges of devices, we compared the protocols on low-end (ATMega2560), mid-range (S32K144 & STM32F767), and high-end (Raspberry Pi 4) devices shown in Figure 7.6. Figure 7.8 shows the results graphically for the run on the STM32F767, with Table 7.5 listing the full execution times for all individual devices.

The SCIANC and PORAMB show the fastest time since they use an algorithm that does not contain EC operations, but with that, they also have more security vulnerabilities discussed later in Section 7.2.3. We can also see that the optimization variants of STS show a considerable improvement over the original, with phase II optimization even outperforming the S-ECDSA in terms of the absolute run time.

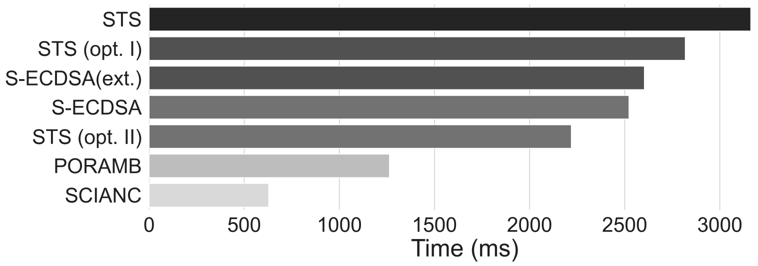


Figure 7.8: Performance evaluation of the key derivation protocols. Adapted from *Publication F*.

Transfer overhead analysis. We have conducted a theoretical overhead analysis based on the total number of transmissions and transferred bytes during one protocol utilization. We assume the minimal size for the implicit certificate of 101 B, identifiers to be 16 B, and 32 B length for challenges, nonce, and MAC. The results are listed in Table 7.4. We can see that there is only a slight difference between the transmission overhead with the STS showing relatively good results.

Table 7.4: Transmission overhead of the analyzed ECQV key derivation protocols.

Protocol	**Steps: Size in Bytes**
S-ECDSA (+ext.)	4(+1): 427(+192) B
STS	4: 491 B
SCIANC	4: 362 B
PORAMB	6: 820 B

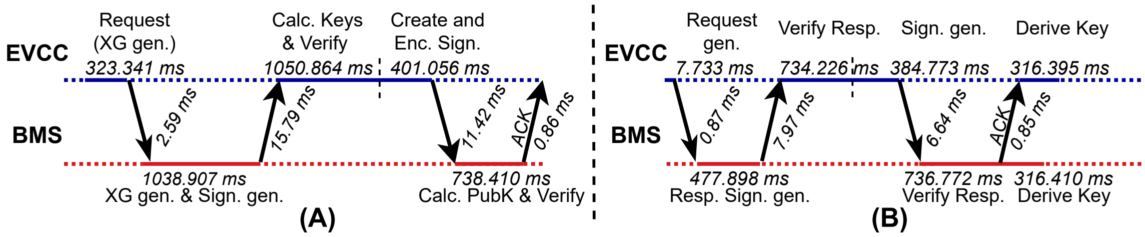


Figure 7.9: Timeline diagram for the BMS ⇔ EVCC test suite performance evaluation of the ECQV key derivation protocols for (A) dynamic STS protocol, and (B) static ECDSA protocol. Adapted from *Publication F*.

Test suite performance evaluation

To depict a common interaction between a BMS and another ECU, we analyze the proposed SKD and DKD for the communication between a main BMS controller and an EVCC over the CAN network [53]. The communication uses the format depicted in Section 6.3.1. To get accurate measurements, both devices are represented with the S32K144 board. The CAN protocol in question is actually CAN-FD with the nominal phase bit rate configured at 0.5 Mbit/s and the data rate phase set at 2 Mbit/s.

We compare the STS with the S-ECDSA without the optimization for a fair comparison and more accurate real-field deployment. Figure 7.9 shows the results of the analysis. The transfer time shown in the graph also includes message processing, with the CAN-FD transfer time being negligible and for most cases $< 1\,ms$. The reported total run time for the STS is $3.28\ s$ with S-ECDSA being at $2.68\ s$. This means that the proposed STS ECQV protocol only accounts for $21.7\,\%$ of the additional time overhead, which is advantageous when considering the security benefits received from its DKD design.

Table 7.5: Execution time in milliseconds of the KD protocols for ECQV for the respective embedded hardware.

Protocol / Device	ATMega2560	S32K144	STM32F767	Raspberry Pi 4
S-ECDSA	36859.3 ± 0.2	2894.1 ± 9.8	2521.8 ± 5.9	18.8 ± 0.1
S-ECDSA (ext.)	36882.6 ± 0.2	2976.2 ± 11.6	2602.7 ± 8.6	18.7 ± 0.1
STS	46262.0 ± 0.1	3622.7 ± 7.0	3162.1 ± 7.5	23.3 ± 0.1
STS (opt. I)	41680.2 ± 1.2	3246.6 ± 13.0	2818.0 ± 11.3	20.9 ± 0.1
STS (opt. II)	32410.8 ± 1.1	2556.8 ± 13.1	2219.3 ± 11.3	16.3 ± 0.1
SCIANC	8990.5 ± 0.0	721.7 ± 0.3	628.1 ± 0.3	4.58 ± 0.0
PORAMB	17932.2 ± 0.1	1471.7 ± 0.6	1263.0 ± 0.4	9.00 ± 0.0

7.2.3 Security evaluation

To complete the evaluation of the BMS security architecture, we perform a security analysis. To do this, we list assets, threats, and countermeasures in a manner similar to Section 7.1.2, but refer to the security requirements from Section 5.1. This list is by no means exhaustive, but at the time of writing it contains the most relevant threats deduced from the previous security requirements analysis.

The system aims to protect the following assets: (A1) *BMS functional operations*, (A2) *BMS data*, and (A3) *BMS network integrity*. We list the following threats and provided countermeasures:

[T1]⟨S,T,R,I,E⟩ *Inserted malicious messages* ↦ (A1), (A2): False update, status, or authorization data.

> **(C1)** *Authentication protocol*: device and certificate authentication protocols as the cornerstone of the proposed design. Also **(C2)**, as no messages are accepted that cannot be authenticated.

[T2]⟨I⟩ *Passive network eavesdrop* ↦ (A2): reading unprotected network content.

> **(C2)** *Secure session*: by relying on the secure key derivation protocols and establishment of secure sessions, i.e., encrypted channels between BMS and communicating units.

[T3]⟨T,I,D⟩ *System update compromise* ↦ (A1), (A3): by obstructing the regular system updates.

> **(R1)** *Missed configuration*: no clear answer against obstructed configuration updates. **(C1)** & **(C2)** protect against update message manipulations and spoofing.

[T4]⟨S,T,R,I,D⟩ *Node-capturing attack* ↦ (A1), (A2), (A3): we reference the attack as described in [197].

> **(C3)** *Regular certificate updates*, & **(C4)** *Dynamic key updates*, controlled by the system design.

[T5]⟨S,R⟩ *Counterfeited devices* ↦ (A3): fake devices intended as a catalyst for other attacks.

> **(C5)** *Gateway access control* as a supporting function along with **(C1)**.

Security evaluation of the key derivation protocols. As can be observed in the previous analysis, we did not list key-related attacks. Here, we analyze them separately and compare the achieved security of the previously analyzed key derivation protocols, including our S-ECDSA and STS, against the list of key-related security threats from Section 5.1.2. We present the results of our analysis in Table 7.6 with symbols: X - weak or no protection, Δ - limited protection, ✓ - fully protected.

The advantage of both S-ECDSA & STS methods rely on the ECDSA for the authentication during the session establishment, which is proven to be secure against passive attacks [236]. STS has an advantage over all other protocols thanks to its perfect forward secrecy attribute. It also possesses a slight advantage in terms of node-capturing attacks, as only future, but not past messages can be compromised. Nevertheless, we do acknowledge that no protocol is fully protected against this kind of attack.

Table 7.6: Security analysis of the ECQV key derivation protocols.

	S-ECDSA (Fig. 5.6)	STS (Fig. 5.7)	SCIANC [200]	PORAMB [197]
Credentials exposure	X	✓	X	X
Node capturing	Δ	Δ	X	X
Key data reuse	X	✓	Δ	X
Key derivation exploit	Δ	✓	Δ	Δ
Authentication procedure	✓	✓	Δ	Δ

7.3 Secure BMS Data Acquisition and Propagation

To demonstrate the important step of BMS data logging for monitoring and diagnostic operations, we evaluate the security propagation design architecture presented in Section 5.5 along with the proposed novel and secure BMS data chain structure from Section 5.4. All evaluations are performed using the implementation test suite from Chapter 6 extended to include the cloud and external systems described in Section 6.5. The tests use 162 bytes for a log sample with a sampling time of 112 ms per BPC for real-hardware analysis, unless otherwise noted. We conclude that our design is feasible with the modern BMS models and topologies in terms of both security and functional dependencies.

7.3.1 Performance evaluation

We analyze performance in terms of the time increase for the BMS block encoding, the impact of header overhead on the increase in block payload size, and the processing time for full network propagation over the cloud.

Secure BMS block encoding analysis

Under block encoding, we consider the whole process of creating the secure BMS block from the sampled log data, to the log block, BMS block, and finally, secure BMS block encodings. We also consider the subsequent secure network block encoding described in Section 6.3.1. The analysis is done for the total encoding time using real hardware and simulated log sampling data to account for more BPC:

- *Using real hardware*: the analysis was performed with either 1 or 2 BPC. The results are shown in Table 7.7 with the individual encoding steps. As we found, most of the time in the operations is spent on security operations, where the focus of possible optimizations through optimized programming or dedicated hardware should be considered. The standard deviation in our tests was negligible and was $< 0.01\,ms$ for all encoding steps.
- *Using simulated data*: the simulation was done on an arbitrary-created sampling data for up to 10 BPC. Figure 7.10 shows the encoding time for three different log sizes. A stable linear growth of the encoding time is observed, with a continuous decrease in the absolute time difference between the BPC sets due to the constant header size, more discussed in the next evaluation section.

Table 7.7: BMS block encoding time using real test suite devices and emulated BMS log data.

Encoding	Log body	BMS block	Secure BMS block	Secure network packet
1 BPC	$0.09\,ms$	$0.24\,ms$	$18.16\,ms$	$20.68\,ms$
2 BPC	$0.17\,ms$	$0.38\,ms$	$23.63\,ms$	$26.33\,ms$
% incr.	$88\,\%$	$58\,\%$	$30\,\%$	$27\,\%$

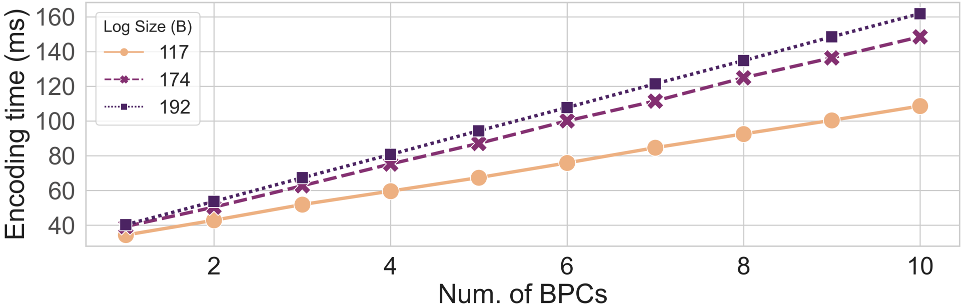


Figure 7.10: BMS encoding time using simulated data for 117, 174, & 192 B log sizes. Adapted from *Publication H.*

Table 7.8: BMS block overhead compared with the log block number.

# of Log Blocks	1	2	4	8	12	16	32
Overhead (%)	21.4	14.7	11.0	9.0	8.3	8.0	7.4

Gateway decoding analysis. To complete the evaluation, we also analyze the decoding and processing time of the network and secure BMS blocks on the gateway side. Decoding a secure BMS block takes $0.48\,ms \pm 0.01\,ms$, where decoding the full network takes $1.35\,ms \pm 0.11\,ms$. While this aspect of the analysis is highly implementation- and device-dependent, since we use devices that resemble real-world deployed devices, we can safely assume that the decoding time of the gateway is negligible compared to the BMS side and would not create a bottleneck in a real-world system when considering multiple BMS devices, e.g., in a decentralized topology.

Secure BMS block header overhead analysis

Since the payload data is dynamic and depends on the application, we can analyze the overhead of the structure from Section 5.4 based on the header size impact. In our test cases, a secure BMS block consists of its own header (16 B) and the header of the underlying BMS block (16 B) with log block headers (each 12 B). We refer to this as the *static header* data size since it always has the same length and can be calculated as $(16 + 16) + 12 * X$, with 'X' being the number of the log blocks, i.e., BPC. A *dynamic header* component comes in the form of the BMS block metadata.

We analyze the theoretical overhead. Figure 7.11 shows the header influence over the projected payload size for a secure BMS block, while Table 7.8 shows the influence when considering the previously

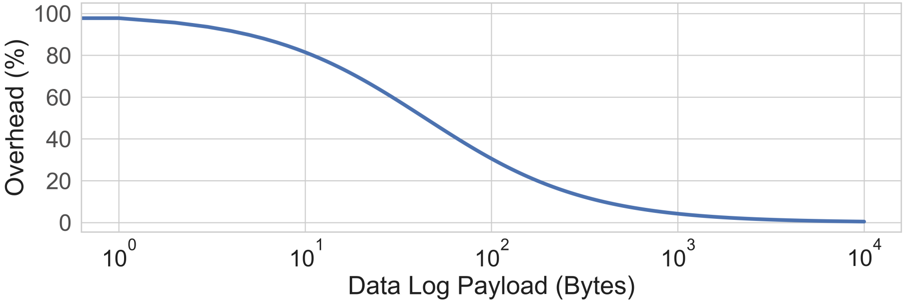


Figure 7.11: Secure BMS block overhead correlated to the sample size for one BPC. Adapted from *Publication H*.

mentioned test suite deployment simulating a different number of log blocks, with the noted payload size of 162 B. As it can be observed, the proposed structure introduces a minimal overhead to the total BMS block size that scales well with the increase of the number of BPC, i.e., also their log block size.

Full BMC block propagation measurements

The propagation was tested using the real hardware and test suite and with one BPC. Figure 7.12 shows the full transfer from the BMS controller, over the gateway and AWS cloud, to the end system. We measured two main points of interest:

1. The transmission from BMS to the gateway that took $85.2\,ms \pm 3\,ms$. The gateway decodes the data and updates the cloud based on the *device shadow* principle, which requires $1.37\,s \pm 0.2\,s$.
2. Cloud receives the data from the gateway and forwards it to the end system that decodes the data, processes the BMS block data, and visualizes it. This step on the end system side requires only $1.6\,ms \pm 0.4\,ms$.

To further verify the time requirements, we have also tested the propagation with different block payload sizes up to 1 kB. We noticed only a slight increase in the total time, with the gateway decoding now taking $1.39\,s \pm 0.21\,s$, and the end system processing showing $2.2\,ms \pm 0.9\,ms$ for 1 kB payload. Additionally, we have analyzed the timing behavior when increasing the total number of sampling cycles. Figure 7.13 graphically shows the result of this investigation.

We can conclude that the main bottleneck lies in the transfer from the gateway to the cloud system. It is possible to perform multiple BMS sampling cycles until one full secure BMS block is further propagated. Nevertheless, it should be noted that the solution presented also accounts for the temporal on-premise data storage, which for safety reasons is more critical. It is expected that the data would be updated over the cloud at certain intervals and not in real-time operations.

7.3.2 Security evaluation

To finalize the proposed system design's security evaluation, we conduct a security analysis as in Sections 7.1.2 and 7.2.3 on the secure BMS data propagation design. The threats are derived based on the security requirements analysis from Section 5.1, i.e., Section 5.1.3. For the study, we relied on the use of DFD to model *threats* (T), *assets* (A), and *countermeasures* (C) shown in Figure 7.14. The assets that we want to protect are: (A1) *BMS log data*, (A2) *gateway-to-cloud payload*, and (A3) *cloud-to-end-system payload*. To narrow down the security analysis and better target our design under the complex external and cloud environments, we also make the following *assumptions* (As): (As1) BMS is considered to be

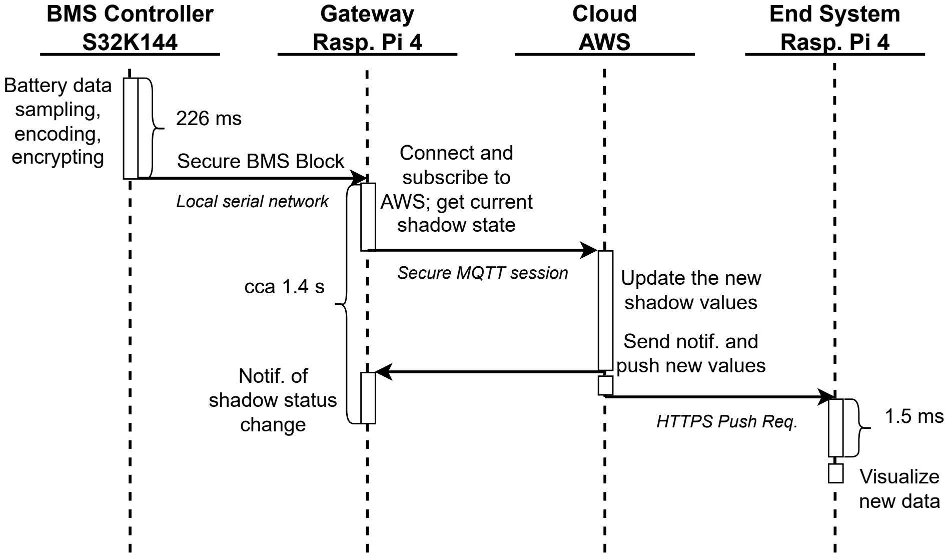

BMS Controller
S32K144
Gateway
Rasp. Pi 4
Cloud
AWS
End System
Rasp. Pi 4
Battery data sampling, encoding, encrypting
226 ms
Secure BMS Block
Local serial network
Connect and subscribe to AWS; get current shadow state
Secure MQTT session
cca 1.4 s
Update the new shadow values
Send notif. and push new values
Notif. of shadow status change
HTTPS Push Req.
1.5 ms
Visualize new data

Figure 7.12: BMS data log propagation steps of the implemented evaluation suite.

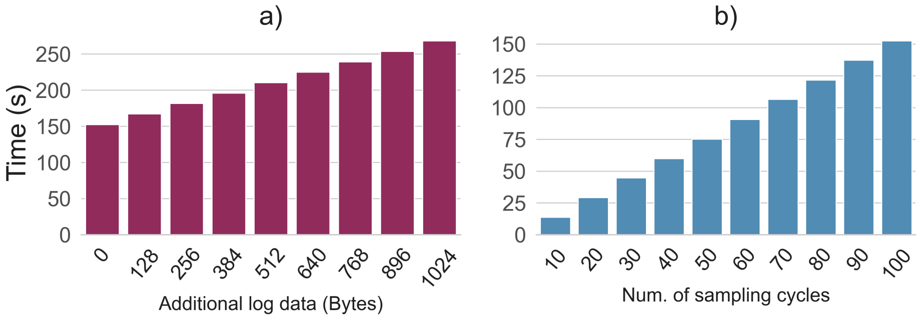


Figure 7.13: BMS log data propagation time measurements correlated with: a) varied payload data for one log sample added to the initial 162 B, b) the number of samples for one test session. Adapted from *Publication H*.

internally secured, (As2) physical attacks are unfeasible, (As3) the secure gateway of the local network is treated as a root of trust, (As4) external systems (cloud and backend) are run from a verified OEM.

The majority of the threats come in the form of “network attacks”, as eavesdropping, tampering, replay and MitM attacks. The results of our analysis are contained in the following threats:

[T1]⟨T,I,D⟩ *Network attack: internal local network* ↦ (A1): passive or active attacks on the network packets with BMS log data payload.

(C1) *Secure BMS block design* (§ 5.4), and with **(C2)** *secure internal network session* (§ 5.3).

[T2]⟨S,D,E⟩ *Spoofing as the gateway* ↦ (A1), (A2): trying to mimic the gateway access to the BMS.

(C3) *Gateway as a secure authority*: enforced with device authentication and with (As2) & (As3).

[T3]⟨T,I,D⟩ *Network attack: update data to cloud* ↦ (A2): the data packets relayed from the gateway.

(C4) *Secure application layer protocols*: integrated protocols, e.g., secure MQTT and TLS (§ 5.5).

[T4]⟨S,T,R,I,D⟩ *Cloud-related attacks* ↦ (A2): a type of spoofing, or privacy attacks on the stored data.

(C1) User’s privacy and content is protected with the secure BMS block.

[T5]⟨T,R,I⟩ *Network attack: backend transfer* ↦ (A3): cloud-relayed packets to the end system.

(C4) Relies on the same Internet-based secure application layer protocols.

[T6]⟨I⟩ *Confidentiality compromise of BMS data* ↦ (A1): hypothetically, if the end system is breached or spoofed akin to **[T4]**, the BMS log data could end in unintended hands.

(C1) Requires a pre-agreed key to decrypt the BMS blocks. The key transfer is fulfilled by (As4).

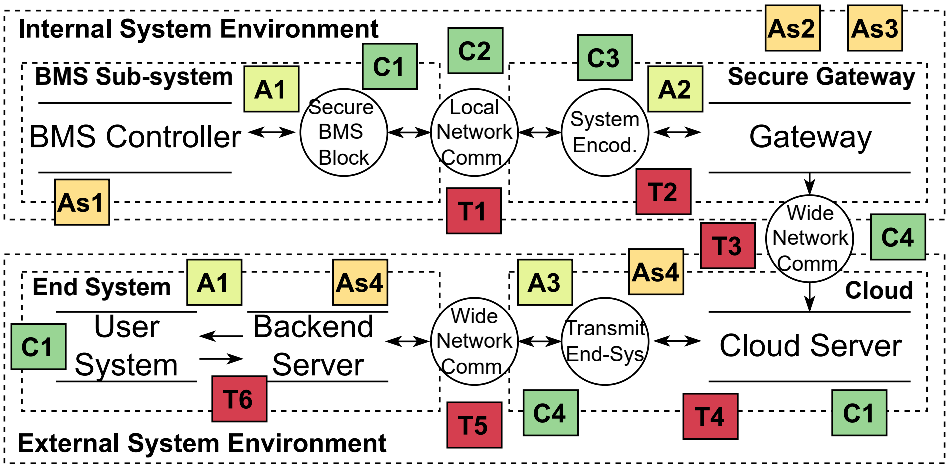


Figure 7.14: Data flow diagram security model for the full BMS data propagation design. The security analysis is set on the BMS log data and its transfer over the five specified propagation layers. Adapted from *Publication H*.

CHAPTER 8
Conclusion

"We study not to know everything, but to know in which books to look for knowledge."

- My Father

Summary: *This chapter concludes this dissertation by providing a summary of the solutions realized for the presented research questions and achieved scientific contributions. An outlook on possible future topics in the field of wireless and secure battery management systems is also given.*

◇◇◇

At the beginning of our research investigation, security with BMS was still considered a largely unexplored field with many open questions and challenges. Designers are perplexed about how to accurately design a security architecture for modern BMS considering new applications and use cases, especially with modern battery passports and second-life usage initiatives. A modern BMS generates a large amount of important battery-related data. It has proven insufficient to simply rely on solutions available in other similar embedded or vehicular networks, as many system design points still remain open. Extensive research has been conducted to address this gap and provide new insights into BMS security and the use of alternative wireless technologies. The contributions achieved in this dissertation are threefold, each dealing with one aspect of the BMS communication layer.

The first layer starts with the BMS sub-system itself, with the intra-module communication and the battery sensor readout processes, where we particularly considered wireless communication in addition to security aspects. We have also extended this concept by considering the readout of external interfaces to complement the diagnostic tracking of the life cycle of batteries for second-life use. Traditional BMS interfaces rely on the use of wired communication, which only adds additional complexity, maintenance, and cost. In recent years, there have been several proposed solutions for wireless BMS design, but as mentioned earlier, they were primarily designed for intra-module communication without considering battery sensor readout or second-life use cases. To achieve an efficient and secure wireless BMS design, we have developed a system design architecture utilizing NFC. We improved on the current SotA NFC security concepts and designed a full security suite with a novel secure NFC data structure, authentication, and secure session protocols.

For the second layer, we analyzed the local BMS network. A BMS communicates with an internal network for the purpose of relaying to and receiving back feedback about battery cell usage. There exist a few different network topologies associated with BMS, but we have noticed a trend associated with the use of a high-performance centralized security gateway, which we have also considered in our design. For the local network that considers a central gateway, we have presented a security architecture that uses novel implicit certificates for device authentication and certificate exchange. The use of implicit certificates may seem unconventional, but we recognized their potential in the growing embedded security community and wanted to extend the design to accommodate modern BMS. We have also broadened the field by presenting a novel design for efficient dynamic key derivation and session establishment with perfect forward secrecy based on the STS protocol, which can be extended to communicate with other ECU. The current limitation of the proposed architecture is in the form of non-standard hardware support, which can otherwise be tailored specifically to the targeted security functions for the BMS. We see this as a potential point for further research where the proposed security architecture can be used as presented, i.e., independently, or alongside other security mechanisms.

For the final BMS communication layer, we integrate the previously proposed secure BMS solutions for enhanced external services for both on-premise and cloud connectivity. The current BMS subsystems lack a common design for transmitting log data to external systems, mainly working with ad-hoc solutions. We address this gap and present a unified and secure BMS chain data structure based on hierarchical block data. We also analyze the important security issues and apply SotA solutions for external cloud management to support the presented BMS lifecycle monitoring design and mitigate potential external threats. Combined with the design of the local BMS architecture, we in fact achieve a fully secure propagation thread that extends from the battery cell sensors, through the BMS controller and the local network, to the cloud and the end systems, providing secure encapsulation.

To accurately demonstrate the integration of the proposed BMS secure and wireless design solutions, an important aspect was to make the design compatible with current automotive devices in the field. To this end, we have realized and implemented a full BMS test suite that includes all three communication layers and integrates the proposed hardware and software design components. The solutions were evaluated in terms of usability, performance and security. We believe that the obtained evaluation results show that our proposed BMS design provides the necessary efficiency and security grade expected from modern BMS applications.

The presented solutions only scratch the surface, as many more security vulnerabilities can be found in modern BMS as research continues to advance. Nevertheless, the work contained in this dissertation proves to be an important step in raising awareness of the need for security research, primarily for the BMS, but also for other related modules and entities that form a large digital ecosystem. We see this as an important contribution and hope that it will lead to the discovery and formulation of many future secure and efficient BMS system design solutions.

Future Work

While the presented design solutions provide a complete suite for a secure and wireless BMS system architecture, many limitations were discovered during the course of the research that can be addressed in future work.

With respect to the proposed BMS NFC design, it would be interesting to deepen this topic and analyze the design in terms of hardware and physical aspects. In particular, the optimization of the BMS NFC communication in terms of the number of communicating devices, attributed antenna design, and customized link layer control. It would also be interesting to investigate alternative high frequency RFID technologies and compare their use against NFC for the targeted BMS use cases.

To complement current BMS data logging processes, as already mentioned in Chapter 3, log aggregation-based methods are slowly gaining interest [75, 165], and their adoption for BMS may well be possible. An important study would need to be done here to determine the necessary requirements compared to different BMS data models and their topologies.

We also see the extension of current security solutions at the local BMS level with modern IoT security protocol solutions for constrained devices such as OSCORE [240], which specifically aim to provide alternative approaches to key derivation and exchange. These solutions could also be coupled with the proposed implicit certificate architecture to further improve system performance.

In case the security architecture continues to rely on implicit certificates, there are currently two main open questions that need to be considered. The first issue arises from the deniability of signatures based on implicit certificates, i.e., a *violation of non-repudiation* addressed by Eric R. Verheul in his paper [182]. As discussed in Section 3.5.1, this security attribute is not violated in our presented design. However, for larger networks and those that do not have adequate device authentication, alternative security solutions should be considered. The other open issue arises from making the security architecture *post-quantum secure*. The ECQV scheme, like any other EC security model, suffers from not being post-quantum secure, which is discussed in a recent paper by Bindel & McCarthy [252]. It would be interesting to see if this is indeed the case, and if the current models can be adapted to retain this security property, or if any additional security layers need to be introduced instead.

Finally, on the topic of BMS module authentication, we have noticed that the PUF concept has been gaining momentum recently. A PUF is a physical entity that uniquely describes a device by relying on the unique but random hardware characteristics introduced during the manufacturing process [253]. They are primarily intended for authentication and key rotation. Currently, there exist many different PUF solutions, such as intrinsic solutions, of which memory-based PUF, e.g., using SRAM, are the most popular. PUF may be considered for use with the battery packs, as they would provide a lightweight and resilient authentication property that could be helpful for the detection of counterfeited devices. However, at the current state of research, several PUF solutions have been shown to be insecure, while many others are costly and complex [192, 193, 194, 254].

CHAPTER A

Appendix: Publications

This dissertation is based on the collection of the following peer-reviewed publications that have been published and presented in recognized journals and conferences. For each publication, a short description is given describing my personal contributions. Figure A.1 further highlights the impact and contribution of each publication on the overall BMS system architecture.

Publications

A. F. Basic, C. Steger, and R. Kofler. "Embedded Platform Patterns for Distributed and Secure Logging," in *26th European Conference on Pattern Languages of Programs (EuroPLoP'21)*, Association for Computing Machinery, 2022.

B. F. Basic, M. Gaertner and C. Steger, "Towards Trustworthy NFC-based Sensor Readout for Battery Packs in Battery Management Systems," in *2021 IEEE International Conference on RFID Technology and Applications (RFID-TA)*, pp. 285-288, IEEE, 2021.

C. F. Basic, M. Gaertner and C. Steger, "Secure and Trustworthy NFC-Based Sensor Readout for Battery Packs in Battery Management Systems," in *IEEE Journal of Radio Frequency Identification*, vol. 6, pp. 637-648, IEEE, 2022.

D. F. Basic, C. R. Laube, C. Steger and R. Kofler, "A Novel Secure NFC-based Approach for BMS Monitoring and Diagnostic Readout," in *2022 IEEE International Conference on RFID (RFID)*, pp. 23-28, IEEE, 2022.

E. F. Basic, C. Steger, C. Seifert and R. Kofler, "Trust your BMS: Designing a Lightweight Authentication Architecture for Industrial Networks," in *2022 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1-6, IEEE, 2022.

F. F. Basic, C. Steger and R. Kofler, "Establishing Dynamic Secure Sessions for ECQV Implicit Certificates in Embedded Systems," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1-6, 2023.

G. F. Basic, C. R. Laube, P. Stratznig, C. Steger, and R. Kofler, "Wireless BMS Architecture for Secure Readout in Vehicle and Second life Applications," in *8th IEEE International Conference on Smart and Sustainable Technologies*, IEEE, 2023.

H. F. Basic, C. Seifert, C. Steger, and R. Kofler, "Secure Data Acquisition for Battery Management Systems," in *26th Euromicro Conference Series on Digital System Design (DSD)*, 2023, *In Press.*

Other Publications List of other publications which were not included (poster, demos, etc.).

P. F. Basic, C. Steger, and R. Kofler, "Establishing Dynamic Secure Sessions for Intra-Vehicle Communication Using Implicit Certificates," in *2022 EWSN, poster session*, p. 196–197, ACM, 2022.

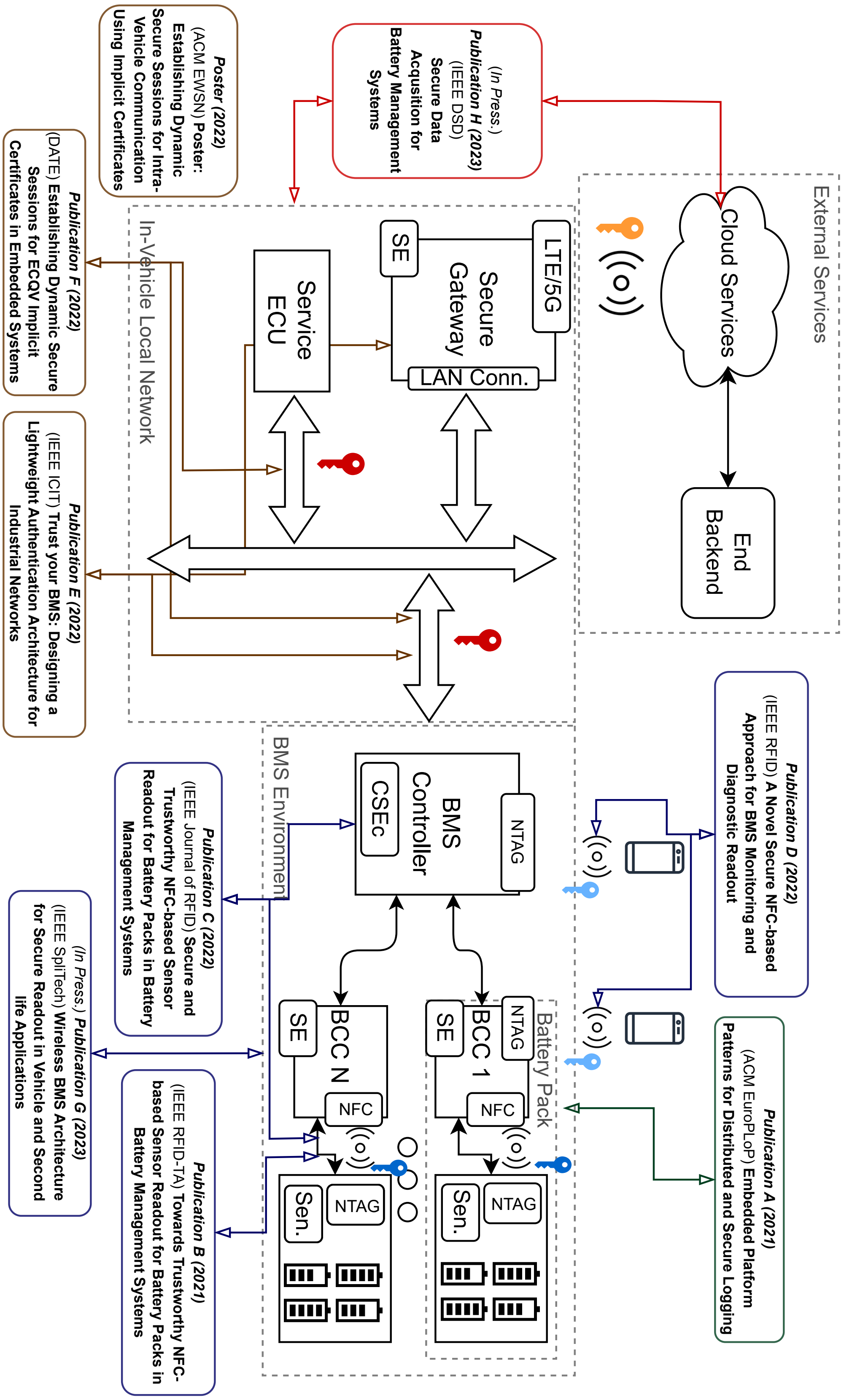

External Services
Cloud Services
End Backend
LTE/5G
Secure Gateway
SE
LAN Conn.
Service ECU
In-Vehicle Local Network
BMS Environment
BMS Controller
NTAG
CSEc
Battery Pack
BCC 1
SE
NTAG
NFC
Sen.
NTAG
BCC N
SE
NFC
Sen.
NTAG
(In Press.) Publication H (2023) (IEEE DSD) Secure Data Acquisition for Battery Management Systems
Poster (2022) (ACM EWSN) Poster: Establishing Dynamic Secure Sessions for Intra-Vehicle Communication Using Implicit Certificates
Publication F (2022) (DATE) Establishing Dynamic Secure Sessions for ECQV Implicit Certificates in Embedded Systems
Publication E (2022) (IEEE ICIT) Trust your BMS: Designing a Lightweight Authentication Architecture for Industrial Networks
Publication D (2022) (IEEE RFID) A Novel Secure NFC-based Approach for BMS Monitoring and Diagnostic Readout
Publication C (2022) (IEEE Journal of RFID) Secure and Trustworthy NFC-based Sensor Readout for Battery Packs in Battery Management Systems
(In Press.) Publication G (2023) (IEEE SpliTech) Wireless BMS Architecture for Secure Readout in Vehicle and Second life Applications
Publication B (2021) (IEEE RFID-TA) Towards Trustworthy NFC-based Sensor Readout for Battery Packs in Battery Management Systems
Publication A (2021) (ACM EuroPLoP) Embedded Platform Patterns for Distributed and Secure Logging

Figure A.1: List of provided publications in relation to the extended BMS system architecture.

A.1 [A] Embedded Platform Patterns for Distributed and Secure Logging

F. Basic, C. Steger, and R. Kofler. “Embedded Platform Patterns for Distributed and Secure Logging,” in *26th European Conference on Pattern Languages of Programs (EuroPLoP’21)*, Association for Computing Machinery, 2022.

Abstract. With the advent of modern embedded systems, logging as a process is becoming more and more prevalent for diagnostic and analytic services. Traditionally, storage and managing of the logged data are generally kept as a part of one entity together with the main logic components. In systems that implement network connections, this activity is usually handled over a remote device. However, enabling remote connection is still considered a limiting factor for many embedded devices due to the demanding production cost. A significant challenge is presented to vendors who need to decide how the data will be extracted and handled for an embedded platform during the design concept phase. It is generally desirable that logging memory modules are able to be addressed as separate units. These devices need to be appropriately secured and verifiable on a different system since data compromise can lead to enormous privacy and even financial losses. In this paper, we present two patterns. First, a pattern that allows flexible logging operation design in terms of module and interface responsibility separation. Second, a pattern for the design of secure logging processes during the utilization of constrained embedded devices. The introduced patterns fulfil the following conditions: (i) flexibility – design is independent of the chip vendors making the logging memory modules easily replaceable, (ii) self-sufficiency – every logging controller is maintained as a separate entity in a decentralized topology, (iii) security – through providing authenticity, confidentiality, and integrity by means of using a dedicated security module.

My Contribution. As the main author of this publication, I contributed by providing the core of the paper concerning the main pattern definitions by defining their context, problems and solutions. I also provided by writing the majority of the included text. Robert Kofler and his team from NXP provided us with an overview and input on how the presented design pattern solutions can be applied when realizing a novel BMS application. Christian Steger provided guidance regarding the main body structure, related work, and the structure of the paper itself. I realized and explained the system examples concerning the use of the patterns on the BMS.

Embedded Platform Patterns for Distributed and Secure Logging

Fikret Basic
Graz University of Technology
Graz, Austria
basic@tugraz.at

Christian Steger
Graz University of Technology
Graz, Austria
steger@tugraz.at

Robert Kofler
NXP Semiconductors Austria GmbH
Co & KG
Gratkorn, Austria
robert.kofler@nxp.com

ABSTRACT

With the advent of modern embedded systems, logging as a process is becoming more and more prevalent for diagnostic and analytic services. Traditionally, storage and managing of the logged data are generally kept as a part of one entity together with the main logic components. In systems that implement network connections, this activity is usually handled over a remote device. However, enabling remote connection is still considered a limiting factor for many embedded devices due to the demanding production cost. A significant challenge is presented to vendors who need to decide how the data will be extracted and handled for an embedded platform during the design concept phase. It is generally desirable that logging memory modules are able to be addressed as separate units. These devices need to be appropriately secured and verifiable on a different system since data compromise can lead to enormous privacy and even financial losses. In this paper, we present two patterns. First, a pattern that allows flexible logging operation design in terms of module and interface responsibility separation. Second, a pattern for the design of secure logging processes during the utilization of constrained embedded devices. The introduced patterns fulfil the following conditions: (i) flexibility – design is independent of the chip vendors making the logging memory modules easily replaceable, (ii) self-sufficiency – every logging controller is maintained as a separate entity in a decentralized topology, (iii) security – through providing authenticity, confidentiality, and integrity by means of using a dedicated security module.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; *Embedded software*; System on a chip; • **Security and privacy**;

KEYWORDS

logging, design pattern, system design, embedded, cybersecurity

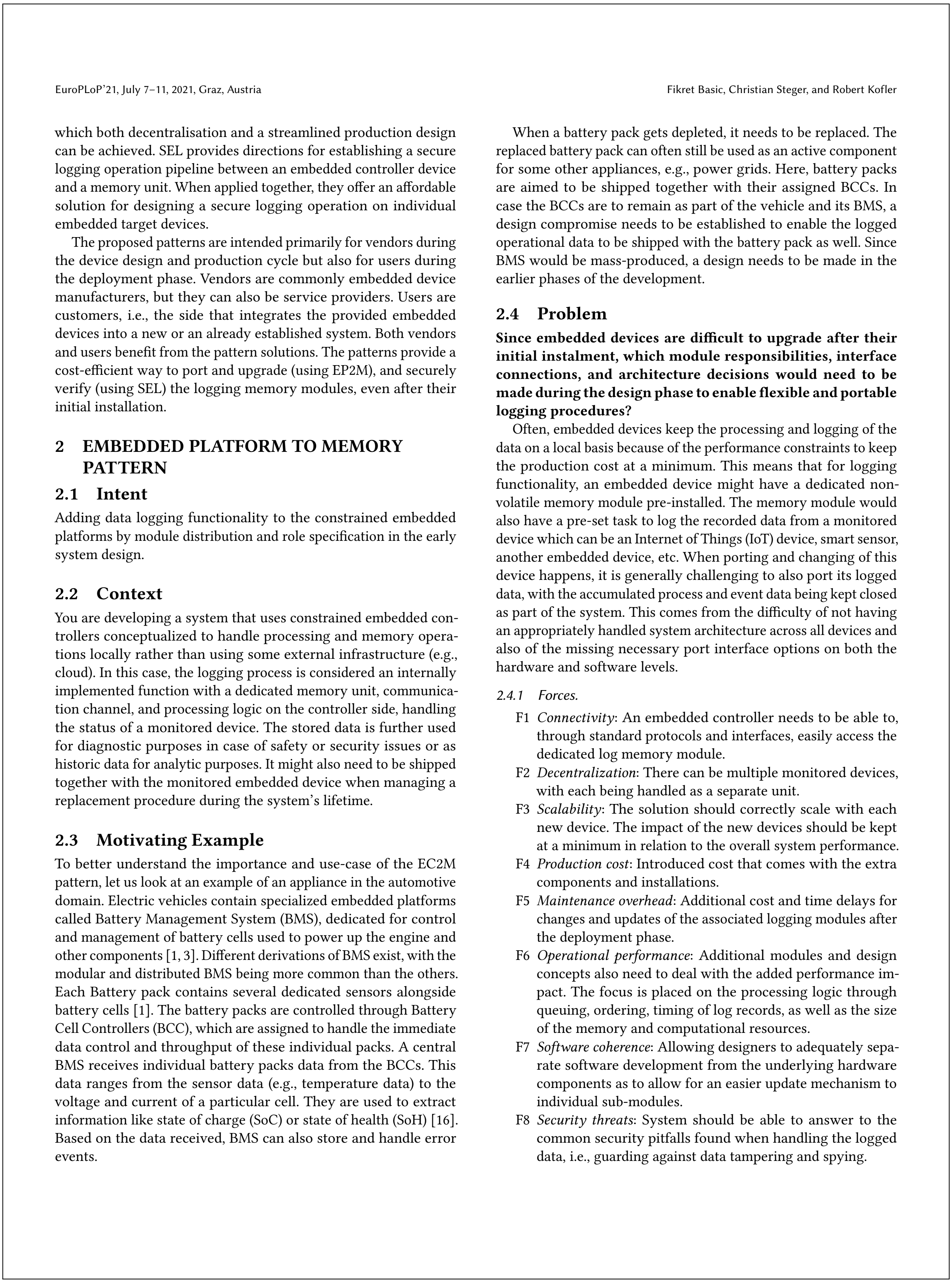
ACM Reference Format:
Fikret Basic, Christian Steger, and Robert Kofler. 2021. Embedded Platform Patterns for Distributed and Secure Logging. In *European Conference on Pattern Languages of Programs (EuroPLoP'21), July 7–11, 2021, Graz, Austria.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3489449.3490004

1 INTRODUCTION

Even today, many embedded devices are still considered constrained, offering only limited resources compared to some more complex platforms. The constraints are presented through the limited size of the provided internal memory (both volatile and non-volatile), limitations on the processing power, reduced employment of communication standards and ports, and even the lack of some extended features (e.g., restrictions on the security capabilities). As the systems become more complex, a necessity arises to capture important log data during its lifetime. The log data is usually used for control and diagnostic purposes, but it can also have secondary uses being a dataset for various machine learning algorithms. This integration is often found today in many modern applications, ranging from surveillance systems to smart grids and vehicles. In addition to having an implemented logging procedure, an essential requirement from many users to vendors is to have the data sufficiently protected so as not to be spied on or tampered with by malicious intruders. These design considerations are becoming increasingly important today, as the preservation of secured data and user's privacy are becoming an increasing topic of interest. From manufacturers of these devices, a considerable effort is required to design a system that fulfils challenges of having (i) limited or absent network capabilities, (ii) security as a co-process, (iii) synchronization between main logic, logging and security operations, (iv) options of porting and changing devices, and (v) option of removing and handling the logged data as a separate unit.

A few of the design patterns previously published in the original Gang of Four (GoF) patterns book have already been in use for the logging process [7]. Historically, this was often achieved using the Chain of Responsibility pattern. Furthermore, the Factory pattern was often used with a combination of Command or Memento to handle the log messages. To supplement the security constraint, some more specialized design patterns like the Secure Logger were introduced as well [15]. They are generally handled as implementation design patterns, and hence, they are not focused on explaining the integration in higher-level designs, especially those concerning modern embedded platforms. As we are going to discuss in the problem statement, this is often a special case. In fact, with the embedded platforms, the controller is often seen as an independent unit from other components, such as sensors, actuators, other controllers, central units, etc. Moreover, many modern patterns are primarily focused on Cloud solutions and do not take into account local and restricted devices. To overcome these restrictions, we introduce: (i) *Embedded Platform to Memory* (EP2M), and (ii) *Secure Embedded Logging* (SEL) patterns. EP2M presents a solution during the design process to handle the division of modular tasks between individual units by proposing a methodology with

EuroPLoP'21, July 7–11, 2021, Graz, Austria
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8997-6/21/07…$15.00
https://doi.org/10.1145/3489449.3490004



which both decentralisation and a streamlined production design can be achieved. SEL provides directions for establishing a secure logging operation pipeline between an embedded controller device and a memory unit. When applied together, they offer an affordable solution for designing a secure logging operation on individual embedded target devices.

The proposed patterns are intended primarily for vendors during the device design and production cycle but also for users during the deployment phase. Vendors are commonly embedded device manufacturers, but they can also be service providers. Users are customers, i.e., the side that integrates the provided embedded devices into a new or an already established system. Both vendors and users benefit from the pattern solutions. The patterns provide a cost-efficient way to port and upgrade (using EP2M), and securely verify (using SEL) the logging memory modules, even after their initial installation.

2 EMBEDDED PLATFORM TO MEMORY PATTERN

2.1 Intent

Adding data logging functionality to the constrained embedded platforms by module distribution and role specification in the early system design.

2.2 Context

You are developing a system that uses constrained embedded controllers conceptualized to handle processing and memory operations locally rather than using some external infrastructure (e.g., cloud). In this case, the logging process is considered an internally implemented function with a dedicated memory unit, communication channel, and processing logic on the controller side, handling the status of a monitored device. The stored data is further used for diagnostic purposes in case of safety or security issues or as historic data for analytic purposes. It might also need to be shipped together with the monitored embedded device when managing a replacement procedure during the system's lifetime.

2.3 Motivating Example

To better understand the importance and use-case of the EC2M pattern, let us look at an example of an appliance in the automotive domain. Electric vehicles contain specialized embedded platforms called Battery Management System (BMS), dedicated for control and management of battery cells used to power up the engine and other components [1, 3]. Different derivations of BMS exist, with the modular and distributed BMS being more common than the others. Each Battery pack contains several dedicated sensors alongside battery cells [1]. The battery packs are controlled through Battery Cell Controllers (BCC), which are assigned to handle the immediate data control and throughput of these individual packs. A central BMS receives individual battery packs data from the BCCs. This data ranges from the sensor data (e.g., temperature data) to the voltage and current of a particular cell. They are used to extract information like state of charge (SoC) or state of health (SoH) [16]. Based on the data received, BMS can also store and handle error events.

When a battery pack gets depleted, it needs to be replaced. The replaced battery pack can often still be used as an active component for some other appliances, e.g., power grids. Here, battery packs are aimed to be shipped together with their assigned BCCs. In case the BCCs are to remain as part of the vehicle and its BMS, a design compromise needs to be established to enable the logged operational data to be shipped with the battery pack as well. Since BMS would be mass-produced, a design needs to be made in the earlier phases of the development.

2.4 Problem

Since embedded devices are difficult to upgrade after their initial instalment, which module responsibilities, interface connections, and architecture decisions would need to be made during the design phase to enable flexible and portable logging procedures?

Often, embedded devices keep the processing and logging of the data on a local basis because of the performance constraints to keep the production cost at a minimum. This means that for logging functionality, an embedded device might have a dedicated non-volatile memory module pre-installed. The memory module would also have a pre-set task to log the recorded data from a monitored device which can be an Internet of Things (IoT) device, smart sensor, another embedded device, etc. When porting and changing of this device happens, it is generally challenging to also port its logged data, with the accumulated process and event data being kept closed as part of the system. This comes from the difficulty of not having an appropriately handled system architecture across all devices and also of the missing necessary port interface options on both the hardware and software levels.

2.4.1 Forces.

F1 *Connectivity*: An embedded controller needs to be able to, through standard protocols and interfaces, easily access the dedicated log memory module.

F2 *Decentralization*: There can be multiple monitored devices, with each being handled as a separate unit.

F3 *Scalability*: The solution should correctly scale with each new device. The impact of the new devices should be kept at a minimum in relation to the overall system performance.

F4 *Production cost*: Introduced cost that comes with the extra components and installations.

F5 *Maintenance overhead*: Additional cost and time delays for changes and updates of the associated logging modules after the deployment phase.

F6 *Operational performance*: Additional modules and design concepts also need to deal with the added performance impact. The focus is placed on the processing logic through queuing, ordering, timing of log records, as well as the size of the memory and computational resources.

F7 *Software coherence*: Allowing designers to adequately separate software development from the underlying hardware components as to allow for an easier update mechanism to individual sub-modules.

F8 *Security threats*: System should be able to answer to the common security pitfalls found when handling the logged data, i.e., guarding against data tampering and spying.

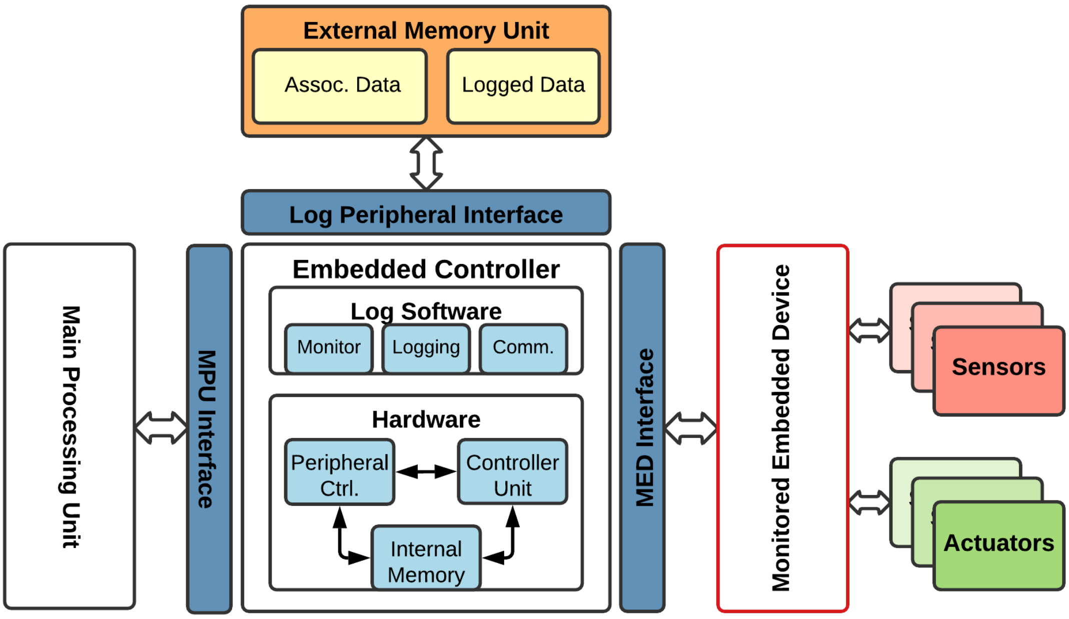


Figure 1: Block demonstration of the suggested modules and connection points during the design of the embedded logging process on a generic device using the proposed solution through the Embedded Controller to Memory pattern.

2.5 Solution

Divide the core logging component into multiple distributed individual embedded controllers, each containing interfaces to the central control module, monitored device, and an external memory unit while keeping process handlers open for relevant events.

A careful interface separation and task assignment during the early system design is necessary to avoid cost increase. A designer needs to identify which core module components are required in the overall system design and what role they have. This role decoupling needs to be done for two reasons: (i) higher responsiveness to faults; in case the data source device gets damaged or corrupted, it is still possible to retrieve the prior logged data, (ii) flexible interchangeability; local device groups appear independent from each other. Therefore it is easier to replace individual components without the added complexity.

Based on the set Forces, as part of the solution, we consider the following devices:

- *Embedded Controller* (EC): The embedded device responsible for the logging and data processing between the monitored targeted device and the system's central unit. This device represents our main target of interest, where the design logic for the embedded logging process is placed. It can mean an expansion of an already existing standard device used for process control as part of an embedded platform.
- *Monitored Embedded Device* (MED): The end-device of the system that is responsible for the data gathering and event action, i.e., it is the targeted device from which the log data is extracted.
- *External Memory Unit* (EMU): The module that stores the log data gathered from an MED, as well as the associated data (configuration, metadata).
- *Main Processing Unit* (MPU): The device tasked for the main system logic control, service providing, and connection to the external services and sub-systems. Since the ECs are designed to be constrained devices, usually found in a decentralized network, a more powerful device is needed to control all, or a group of, ECs in the system. In the system implementation, this device can be the same as one of the ECs as long as the resources offered correspond to the requirements presented by the overall system.

The proposed design is shown in Figure 1. A Microcontroller Unit (MCU) can be used as the hardware control unit to construct an EC. It is used to handle, through software, the logging logic, MED process monitoring, and communication flow control, among other assigned operations. Another function of this controller unit is to handle the synchronization and sampling rate of the MED. These also include administering the commands from the MPU and controlling the internal operational states (e.g., active, idle, sleep). Optionally, an EC can also internally incorporate volatile and non-volatile memory, as it is indicated by the internal memory block.

These parts can, however, highly influence the end-design cost and are recommended to be considered sparingly. Due to the limitation in functions, an adequate Application-Specific Integrated Circuit (ASIC) chip can also be provided instead of the more costly MCU. An EC also provides separate interfaces for the communication with the MPU and the dedicated MED. These can either be wireless or wired, depending on the design constraints. Examples of wired interfaces would be Inter-integrated Circuit (I2C), Serial Peripheral Interface (SPI), Universal Asynchronous Receiver-Transmitter (UART). For the wireless communication interface we recommend Bluetooth Low Energy (BLE), low-frequency Radio-frequency Identification (RFID), and higher-frequency RFID, like Near-Field Communication (NFC), among other standards. While the wireless interfaces offer more applicability in their use-cases, it should be noted that they also require additional handling and construction cost for error corrections and cybersecurity preservation.

2.5.1 Log Memory Interconnection. The pattern establishes cheap, flexible, and extensible handling of a memory module dedicated to logging purposes. To this end, an interface is provided to the EC for the logging handling. Here, we propose the use of an *External Memory Unit* (EMU) module. The EC needs to treat the added memory module as an external unit rather than a pre-embedded component that is part of the EC. This is done to achieve the portability and flexibility in adding and removing the memory that houses the logged data. It is recommended that the new EC device already has a pre-built interface port for communication with the extensible EMUs. These ports can use different communication standards. It is recommended to use a well-established and long-term lasting standard. Among others, these include I2C, SPI, and UART (serial). A wireless standard can also be used, although it is not recommended for this interface. A wireless interface would add an additional increase in cost and complexity, where it would also have less support when porting it among the vendors.

2.5.2 Protocol Handling. As already noted, the programming logic for the logging and memory handling should be appropriately handled through software implementation as part of the controller unit of the EC. The logging process should not interfere with the main controlling procedures but rather work as an extension. Timing delays are to be expected; hence the sampling rate for the logged data needs to be adapted accordingly. Figure 2 illustrates the logic behind the logging process and individual components. The logging phase starts with first establishing the connection, followed by an interactive communication between the EC, MED and the Log Memory to catch and store the targeted data during the system's run-time. The last phase, closing of the communication, considers the remaining processes that are carried out after the main logging phase is over, e.g., calculating and storing associated operational data. The manner in which the logging processes are managed is an implementation task and is therefore left to the individual system designers. Here, we only indicate the main principle behind the logging procedure. At the end of the dedicated lifetime of a MED in a system, it might be necessary to replace the component, and with that, to also port the old one together with the previously used EMU. This activity can be easily achieved since the system is intended to be modular, with each unit having the capability to be individually transported and replaced.

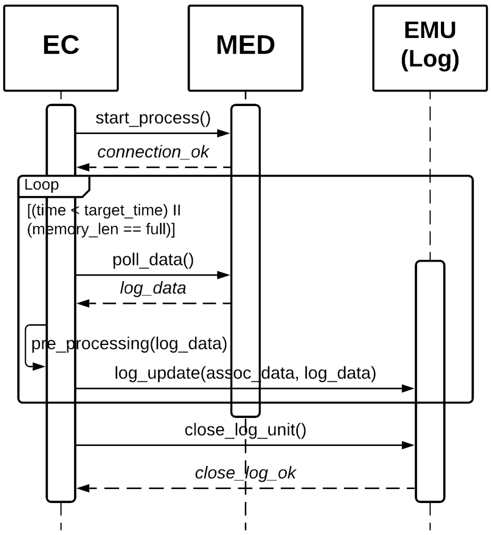


Figure 2: Sequence diagram with function calls for the starting, run, and closing phase during the logging operation.

2.6 Consequences

To better assess the suggested pattern, we are going to list benefits and liabilities corresponding to the Forces from Section 2.4.1.
The benefits when using the EP2M pattern are:

F2 Each pair of EC and MED is independent and unique, along with the dedicated memory component.

F3 The main logic control of the embedded platform is managed by a separate unit (MPU), that also controls which devices are added and handled, but does not cover the actual logging procedure. Therefore, it is possible to expand the system by adding additional ECs and MEDs, as long as the number adheres to the limitations set by the MPU.

F4 Since the solution proposes a modulated system, each component can easily be processed in a streamlined production line. The amount of the overall hardware and software necessary for the cross-platform support on the EC would result in its reduction as well.

F5 After the deployment, each memory module is easily replaceable. Also, the overall complexity is reduced when handling the logging procedure; it requires no special consideration, other than the design points already implemented in a pre-deployment phase.

F6 MPU, which represent the central logic, is free from the logging process. This frees up the resources necessary for the general system run.

F7 Through the careful hardware & software design separation on the EC, the software is able to adequately access the necessary resources on the underlying hardware layer.

The liabilities when using the EP2M pattern are:

F1 Since the production of the EC is handled separately from the memory module dedicated for data logging, an additional interface is needed for the EC for it to be able to communicate with an external log memory module. This adds additional cost and handling complexity.

F3 Expansion of the system is limited by the system resources offered from the central MPU device. These are fairly predefined during the design phase.

F5 Maintenance and manual covering of individual devices could be an issue as the system scales. Additional devices would put a lot of constrains when handling them.

F8 The pattern helps in protecting system availability through its distributed solution. However, it is not focused on providing cybersecurity protection for secured log data.

2.7 Known Uses

The proposed solution can generally be found under two scenarios:

- *End-consumer aimed applications*: special home appliances, mobile phones, and surveillance systems [10, 11].
- *Mission critical industrial applications*: process control systems, cellular base stations, medical systems, remote environmental data loggers and monitors [8, 9].

Among these, the most common application today can be found as part of the more prominent industrial solutions where the utilization is necessary for traceable failure analysis. It is often used in the aeronautic and automotive domain inside the control "Black Boxes". Initially, these systems were aimed to provide a removable and safe memory module that logs the operational data during a dedicated session, where today they are also intended to provide sufficient security considerations [2]. Black Boxes are slowly becoming a norm in modern automobiles, designed to serve relevant operational data in case of accidents [12].

2.8 Realized Example

To better understand where and how the EC2M pattern can be employed, we are going back to the example specified in Section 2.3. Here, we will apply our solution and analyze the outcome. As already noted, it is necessary to use a design solution to the BMS that covers the logging process for the sensor data received from battery packs. The outcome of the integrated design modules can be seen in Figure 3. As demonstrated, the BCC has been modified and extended with an interface for communication with an external memory module. Furthermore, the software in the MCU is developed to handle channel control to the memory module and appropriately cover the logging sample rate. The BCC communicates through additional interfaces with a battery pack on one side and the central BMS on the other end. In our applied solution, BMS is the MPU. BCC represents the EC, memory module is the EMU, with the battery pack being the MED. Each BCC and its assigned battery pack are handled as an individual group unit. An important aspect on why the solution had to be applied in the earlier stage of the development cycle, as suggested by the pattern, is the design of the essential interface connections. The system is expandable; hence additional BCCs and their battery packs can be attached in a daisy chain connection as indicated in Figure 3.

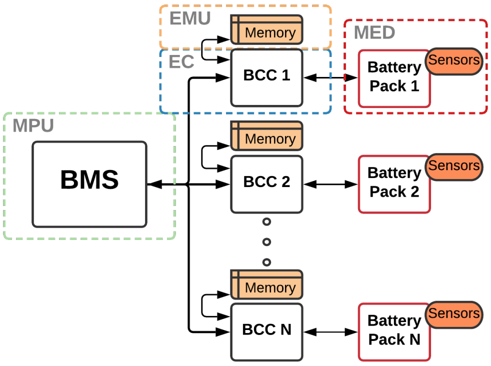


Figure 3: Realized motivating example using the Embedded Controller to Memory architectural pattern. Employed on a use-case concerning Battery Management System (BMS), having the ECs represented through Battery Cell Controllers (BCCs) that log the data from the Battery Packs.

2.9 Related Patterns

The Chain of Responsibility [7] is a behavioural design pattern that is structurally similar to the presented EC2M pattern. It can also be used to handle logging or auditing functionality. However, it does not account on its own for the modular responsibility distribution during the system's design phase. It is primarily implementation-oriented and can be applied on the software stack.

3 SECURE EMBEDDED LOGGING PATTERN

3.1 Intent

Answering to the security needs by extending the data logging capabilities in the embedded platforms by adding security modules and services. The proposed pattern can be used to add the logging security features together with a design-focused logging solution. An example for the embedded logging design solution would be the Embedded Platform to Memory pattern described in Section 2.

3.2 Context

An embedded platform is being designed which uses local memory devices to handle the storage of lifetime logging data. For the reasons of the cost and memory size limitations, as well as not having, or having limited, access to a wide network, it is intended for the platform to rely on local solutions rather than remote services. Often, these types of systems are closed and protected under a specific group. It is critical that the stored data maintains its integrity and is only managed through authorized handlers in this environment. The embedded system would consist of a selection of hardware modules, interfaces, and implemented software functions. The hardware modules are divided by their respective tasks and placement. These are usually tied to a specific architecture and their upgrade can be very difficult, or sometimes not even possible.

3.3 Motivating Example

For a complete example of the usage of the pattern, we will focus on the BMS use-case explained in Section 2.3. As stated, it is desirable to enable the porting of the stored memory units together with the battery packs as to be able to track the health of the used battery packs or for any additional data post-processing. It is of critical importance that only valid battery packs are being transported and that it is possible to authenticate the memory units used with the previous battery packs. This constraint is essential to make sure that no malicious attacks through a modified memory unit are possible. Additionally, the data that is stored needs also to be secured. The reason for making it secure is to guard it against any potential malicious attacks or even faults that can arise from oversights during service and maintenance.

The constraint is still present to handle these design steps in the initial development phase. This is done for the fact that the battery packs would be mass-produced. Any change that would otherwise be done later could jeopardize the security of the battery packs and add an additional cost.

3.4 Problem

How to design an embedded platform that is able to securely handle, but also port and verify, logging data from its source to a designated entity?

In embedded platforms that use distributed module placement, logging process and porting of the logged data often introduce security risks. Porting would need to be done either by using a manual external device or having a connection to a network, both of which might be difficult, or even not feasible, under the platform constraints. Additionally, changes introduced to the system on a physical layer may hamper security during the transfer of the saved data and present a high level of porting complexity. Modules used would need to account for security functionality and have pre-defined elements that supplement them. These considerations result in making it a very challenging and expensive task.

Different malicious attacks can be mounted aimed directly at the content of the logged data during both the active logging period and during the offload transfer period. It is challenging to derive a definite list of threats, as these are usually use-case or application dependant. Here we focus primarily on generic threats that are found in embedded logging systems. Specifically, we consider the following main threats:

- *Spying on the targeted process*: If not properly secured, an attacker can derive information, and even knowledge, from the stored log data by a direct port access.
- *Logged data tampering*: Unauthorized change of the current, or previously stored, log content. This includes active attacks on the communication points during the ongoing logging process, but also direct tamper attacks on the devices.
- *Counterfeited sources*: Each logged data is tied to an affiliated monitored device that is also supplied from a certified manufacturer. During the offload transfer period of the device, i.e., when the change of the targeted monitored device happens, the device can be replaced with a counterfeited or a malicious one. A different attack would be by using the same device but replacing the data inside it.

3.4.1 *Forces.*

F1 *Streamlined HW/SW integration*: Implementation of the hardware and software elements associated with the security functionality need to be easily replicated across multiple devices and vendors.

F2 *Production cost*: Changes made to the hardware and software design of the embedded systems can result in an increased manufacturing cost.

F3 *Limited resources*: The embedded system needs to be able to execute all necessary functions under different constraints.

F4 *Security - confidentiality and integrity*: Necessary measures need to be taken which should prevent the logged data to be tampered or spied on.

F5 *Security - authenticity*: The logged data that is stored needs to be able to be properly identified and verified that it comes from a valid source entity. This authenticity is also necessary each time the data needs to be accessed during the active period, i.e., when the data is retrieved for the analytic or other operational purposes.

3.5 Solution

Ensure that the monitored logged data will be securely protected through an integrated security module relaying data to the memory module and authenticated by using necessary hardware and software critical components embedded during the deployment phase.

When implementing a logging procedure as part of the constrained embedded platform, the security requirement is achieved by integrating a Security Module (SM) as part of the EC. While adding the SM to individual EC devices adds to the overall cost, it does make the system more decentralized. Furthermore, this ensures that the security operations are distributed without heavily impacting the performance. EC device vendors could also not guarantee that the logged data would be secured since the EC itself would not handle that constraint. Therefore, it is necessary to also couple the security operations as part of the EC to appropriately address the security design and attest that the information stored will be protected. Figure 4 depicts the design behind the solution and shows the recommended building blocks. The following components are listed:

- *Embedded Controller* (EC): Contains necessary interfaces for the communication, main driver logic, and the control bridge between the data that is to be stored and the security driver.
- *Logging Memory Unit* (LMU): Dedicated device for storing the encrypted data; contains necessary description data, encrypted security keys, and the encrypted data.
- *Security Module* (SM): Provides security operations; works as a security bridge between the EC and the LMU.
- *Source Verification Device* (SVD): Device tied to a particular LMU and used for the authentication purpose; can contain necessary authentication data, i.e., private-public key pair, and/or a certificate. It is also generally seen as the device from which logging process data is retrieved (data source).

Software functions and associated security data would be handled by the SM itself. At the same time, it would use the logic controller of the EC to drive the overall processing and data preparation when

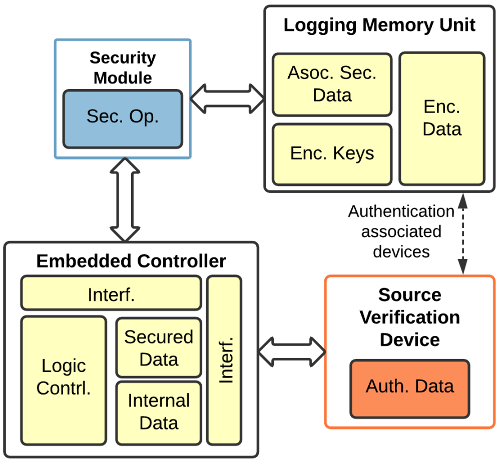


Figure 4: Design-based solution in task separation for handling security logging by providing secure operations and device authentication.

storing it as part of the logging. It is necessary to keep the SM cheap in design. As a minimum security requirement when storing the logging data, we propose to use encryption and authentication for the stored data. These can either be achieved by using separate security functions or applying a suite like Authenticated Encryption (AE) to handle this process. The data integrity check (additional AE data or a separate operation) would be saved in a separate memory block inside the LMU. These, however, do not need to be secured, but they do need to be checked by the EC from the SM each time a new LMU is authenticated. They also need to be periodically updated from the SM after new data is written. The SM should also offer the functionality of storing and handling the key data used by the security operations. Additionally, an EC together with its SM could also provide a Key Derivation Function (KDF). The basic principle of deriving and delivering the keys between the parties is left to the designers. The keys are generally securely encrypted and stored in the LMU secure section. The authentication operations can either be managed using symmetric-based authentication, e.g., AES challenge/response mechanism or by using asymmetric authentication, e.g., Public Key Infrastructure (PKI). The security operations can be handled entirely through software or be hardware-derived, where the hardware operations usually offer better performance, e.g., hardware implementation of the Advanced Encryption Standard (AES). While we consider using the integrated security engine through a dedicated SM as the most cost-effective solution, other dedicated hardware security components can also be examined. These include Secure Elements (SE) and Trusted Platform Module (TPM). However, unlike the integrated secure engine, SE and TPM are more complex to incorporate and much more costly.

The pattern is additionally aimed at providing an affordable and secure solution when transporting and then replacing an LMU.

This process is depicted in Figure 5. Here, a user would receive the LMU together with the SVD from a previous socket. When integrating it into the new system, it might be necessary to verify this memory unit alongside the newly installed SVD, which has been formerly taken out from the older system. This is achieved by using the previously explained security verification functions that the new EC, through its design with SM, would possess as well. The verification process needs to be successfully completed for the LMU to be further used, be it just for the analytic or for continuing operations.

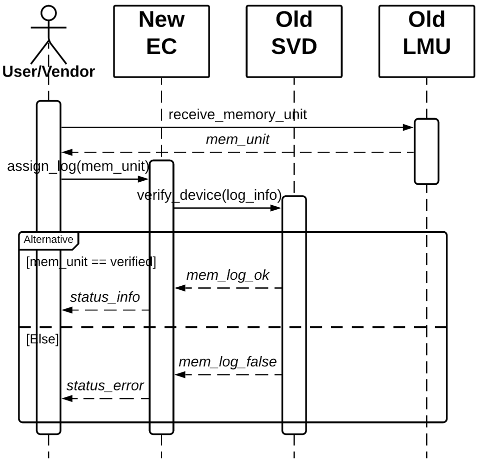


Figure 5: Sequence diagram describing the verification process during the porting of LMU and a SVD from a previous to a new embedded device platform.

3.6 Consequences

This section lists the benefits and liabilities found when applying the SEL architectural pattern based on the Forces from Section 3.4.1. The benefits of the SEL pattern are:

F1 As the pattern suggests using a dedicated security module and predefined security functions per EC, the general production design can be applied on a larger scale.

F4 The pattern proposes the use of a dedicated SM that should allow, at a minimum, encryption and integrity check for handling the security of the stored data.

F5 Additionally, the SM needs to allow for a method of authentication and verification of individual memory modules that were previously tied to a specific pair of EC and MED.

The liabilities when using the SEL pattern are:

F1 As long as the security logging is only handled in a closed local embedded platform, further system updates and configurations are not handled with the proposed pattern.

F2 Each device in the suggested embedded platform is handled as a separate unit, meaning that each embedded controller comes with their own security module. This advantage at flexibility comes also with a drawback, and that is the increase of the general production cost.

F3 Many embedded devices today are limited in terms of the extension capabilities, i.e., either not containing their own security modules or not providing additional interfaces.

3.7 Realized Example

Based on the open design question presented in Section 3.3, we present a solution in form of a module extension. Here, security is applied to guarantee: (i) confidentiality - protecting necessary system data by only providing data associated to the BMS operational cycle, (ii) repudiation – an action can be tied to the entity that caused it, and (iii) integrity – data has not been modified.

The resulted block design is shown in Figure 6. The security functionality is controlled by an internal SM service engine. The SM communicates directly with the EC and the memory interface.

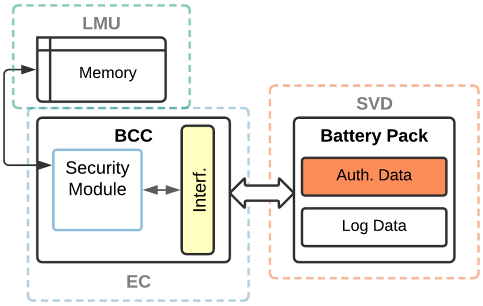


Figure 6: Realized example based on the Secure Embedded Logging pattern. Applied on a BCC of a BMS by extending its applicability for the secure logging process.

When the need arises for a battery pack to be replaced, using this design, it is possible to also easily port the whole BCC or just the memory module with the logged data, as indicated in the pattern solution. Based on the implementation, the BMS would verify the newly added BCCs, while a BCC can independently run the verification operation on the connected battery pack and memory module.

3.8 Related Patterns

A pattern that is similar in design but different in intent and context would be the Security Logger and Auditor [6]. It is focused on logging security-sensitive actions from different users. Hence, this pattern offers a security solution in tying recorded information with the particular users of a system on an architectural level. Another similar pattern would be the Secure Logger which is traditionally used for capturing targeted application events [15]. It is an implementation design pattern that can be applied on the software level in situations where otherwise system constraints are of no concern and are not taken into the design consideration.

4 RELATED WORK

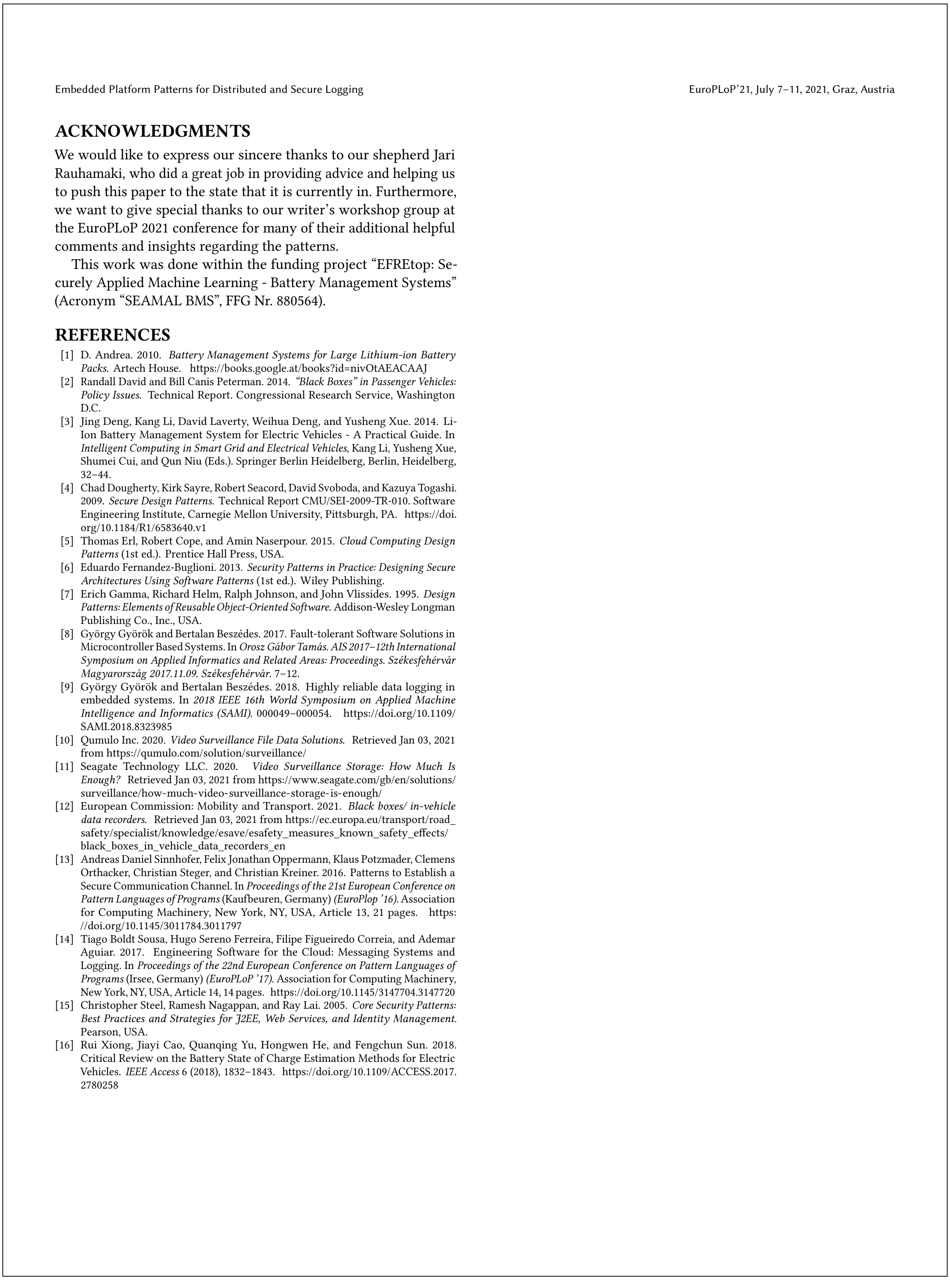
The patterns presented in this work are focused on delivering a design-level solution when processing data logging by providing task separation and secure handling in embedded devices. To successfully use the secure logging functions, it is necessary to implement them. This process can be done by expanding their use through one of the software-focused patterns. Several logging, and even secure logging, design patterns were already previously researched and published. Among them we have:

- The Secure Logger [15], which is prominent, as it provides a simple solution for handling logging in different systems on an implementation level.
- Security Logger and Auditor [6] is an another pattern that provides a conceptual solution for the logging and auditing with protection mechanisms for the logged information. It deals with the repudiation aspects of information by tracking and linking the users with the logged actions.
- Traditionally, Chain of Responsibility pattern [7] is also often used for logging implementations as well. To make sure that the condition is met which guarantees that logging will be made in a secure manner, the secure variant of the Chain of Responsibility pattern can be applied [4].

During the deployment, the communication interface between MPU, EC, and MED might remain insecure. Work presented in [13] presents an answer, where the Symmetric Key Cryptography pattern can be used for establishing a secure communication channel. Based on the system use-case, it might also be necessary to supervise and store the secured logging data online rather than locally. Collaborative Monitoring and Logging can be used as a template to handle the remote side of service, with an additional pattern like Secure External Cloud Connection used to establish the now necessary secure connection [5]. Furthermore, the work presented in [14] lists three distinct but related design patterns that can be utilized to build a remote messaging interface. These can be applied to the presented EC2M pattern to extend the abstraction on the memory unit, which in this case would mean replacing the extensible memory module with a cloud service.

5 CONCLUSION

In this paper, we have demonstrated how it is possible to implement a secure and efficient logging solution even in closed and constrained embedded systems through a careful design and separation of tasks and modules. Furthermore, necessary steps are proposed where, in case of faults or unauthorized actions, the security side would be adequately handled by giving a guide on the placement and use of a security module. This becomes increasingly important when the need for replacement and update of the active embedded devices arises, as it is necessary to also port history log data of the device's lifetime. Lastly, the work presented in this paper is meant to encourage different vendors to consider the implementation of the secure logging functionality in local devices by not breaking the initial cost and size limitations and to help users in employing and maintaining the provided logging services for the continuing device utilization. More importantly, it streamlines the availability of the logged data to the users through simplification of transfer and security verification of memory units.



ACKNOWLEDGMENTS

We would like to express our sincere thanks to our shepherd Jari Rauhamaki, who did a great job in providing advice and helping us to push this paper to the state that it is currently in. Furthermore, we want to give special thanks to our writer's workshop group at the EuroPLoP 2021 conference for many of their additional helpful comments and insights regarding the patterns.

This work was done within the funding project "EFREtop: Securely Applied Machine Learning - Battery Management Systems" (Acronym "SEAMAL BMS", FFG Nr. 880564).

REFERENCES

[1] D. Andrea. 2010. *Battery Management Systems for Large Lithium-ion Battery Packs*. Artech House. https://books.google.at/books?id=nivOtAEACAAJ

[2] Randall David and Bill Canis Peterman. 2014. *"Black Boxes" in Passenger Vehicles: Policy Issues*. Technical Report. Congressional Research Service, Washington D.C.

[3] Jing Deng, Kang Li, David Laverty, Weihua Deng, and Yusheng Xue. 2014. Li-Ion Battery Management System for Electric Vehicles - A Practical Guide. In *Intelligent Computing in Smart Grid and Electrical Vehicles*, Kang Li, Yusheng Xue, Shumei Cui, and Qun Niu (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 32–44.

[4] Chad Dougherty, Kirk Sayre, Robert Seacord, David Svoboda, and Kazuya Togashi. 2009. *Secure Design Patterns*. Technical Report CMU/SEI-2009-TR-010. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. https://doi.org/10.1184/R1/6583640.v1

[5] Thomas Erl, Robert Cope, and Amin Naserpour. 2015. *Cloud Computing Design Patterns* (1st ed.). Prentice Hall Press, USA.

[6] Eduardo Fernandez-Buglioni. 2013. *Security Patterns in Practice: Designing Secure Architectures Using Software Patterns* (1st ed.). Wiley Publishing.

[7] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Longman Publishing Co., Inc., USA.

[8] György Györök and Bertalan Beszédes. 2017. Fault-tolerant Software Solutions in Microcontroller Based Systems. In *Orosz Gábor Tamás. AIS 2017–12th International Symposium on Applied Informatics and Related Areas: Proceedings. Székesfehérvár Magyarország 2017.11.09. Székesfehérvár*. 7–12.

[9] György Györök and Bertalan Beszédes. 2018. Highly reliable data logging in embedded systems. In *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. 000049–000054. https://doi.org/10.1109/SAMI.2018.8323985

[10] Qumulo Inc. 2020. *Video Surveillance File Data Solutions*. Retrieved Jan 03, 2021 from https://qumulo.com/solution/surveillance/

[11] Seagate Technology LLC. 2020. *Video Surveillance Storage: How Much Is Enough?* Retrieved Jan 03, 2021 from https://www.seagate.com/gb/en/solutions/surveillance/how-much-video-surveillance-storage-is-enough/

[12] European Commission: Mobility and Transport. 2021. *Black boxes/ in-vehicle data recorders*. Retrieved Jan 03, 2021 from https://ec.europa.eu/transport/road_safety/specialist/knowledge/esave/esafety_measures_known_safety_effects/black_boxes_in_vehicle_data_recorders_en

[13] Andreas Daniel Sinnhofer, Felix Jonathan Oppermann, Klaus Potzmader, Clemens Orthacker, Christian Steger, and Christian Kreiner. 2016. Patterns to Establish a Secure Communication Channel. In *Proceedings of the 21st European Conference on Pattern Languages of Programs* (Kaufbeuren, Germany) *(EuroPlop '16)*. Association for Computing Machinery, New York, NY, USA, Article 13, 21 pages. https://doi.org/10.1145/3011784.3011797

[14] Tiago Boldt Sousa, Hugo Sereno Ferreira, Filipe Figueiredo Correia, and Ademar Aguiar. 2017. Engineering Software for the Cloud: Messaging Systems and Logging. In *Proceedings of the 22nd European Conference on Pattern Languages of Programs* (Irsee, Germany) *(EuroPLoP '17)*. Association for Computing Machinery, New York, NY, USA, Article 14, 14 pages. https://doi.org/10.1145/3147704.3147720

[15] Christopher Steel, Ramesh Nagappan, and Ray Lai. 2005. *Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management*. Pearson, USA.

[16] Rui Xiong, Jiayi Cao, Quanqing Yu, Hongwen He, and Fengchun Sun. 2018. Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles. *IEEE Access* 6 (2018), 1832–1843. https://doi.org/10.1109/ACCESS.2017.2780258

A.2 [B] Towards Trustworthy NFC-based Sensor Readout for Battery Packs in Battery Management Systems

F. Basic, M. Gaertner and C. Steger, "Towards Trustworthy NFC-based Sensor Readout for Battery Packs in Battery Management Systems," in *2021 IEEE International Conference on RFID Technology and Applications (RFID-TA)*, pp. 285-288, IEEE, 2021.

Abstract. In the last several years, wireless Battery Management Systems (BMS) have slowly become a topic of interest from both academia and industry. It came from a necessity derived from the increased production and use in different systems, including electric vehicles. Wireless communication allows for a more flexible and cost-efficient sensor installation in battery packs. However, many wireless technologies, such as those that use the 2.4 GHz frequency band, suffer from interference limitations that need to be addressed. In this paper, we present an alternative approach to communication in BMS that relies on the use of Near Field Communication (NFC) technology for battery sensor readouts. Due to a vital concern over the counterfeited battery pack products, security measures are also considered. To this end, we propose the use of an effective and easy to integrate authentication schema that is supported by dedicated NFC devices. To test the usability of our design, a demonstrator using the targeted devices was implemented and evaluated.

My Contribution. My contribution to this paper was in specifying the main design, defining the core motivation and research challenges, and providing related work. As the main author, I also contributed to the majority of the written text. Martin Gaertner supported the implementation and evaluation of the project results. Christian Steger provided mentoring guidance, challenges specification, and inputs on the paper organisation. I also provided the security evaluation.

Towards Trustworthy NFC-based Sensor Readout for Battery Packs in Battery Management Systems

Fikret Basic, Martin Gaertner, Christian Steger
Institute for Technical Informatics
Graz University of Technology
Graz, Austria
{basic, gaertner, steger}@tugraz.at

***Abstract*—In the last several years, wireless Battery Management Systems (BMS) have slowly become a topic of interest from both academia and industry. It came from a necessity derived from the increased production and use in different systems, including electric vehicles. Wireless communication allows for a more flexible and cost-efficient sensor installation in battery packs. However, many wireless technologies, such as those that use the 2.4 GHz frequency band, suffer from interference limitations that need to be addressed. In this paper, we present an alternative approach to communication in BMS that relies on the use of Near Field Communication (NFC) technology for battery sensor readouts. Due to a vital concern over the counterfeited battery pack products, security measures are also considered. To this end, we propose the use of an effective and easy to integrate authentication schema that is supported by dedicated NFC devices. To test the usability of our design, a demonstrator using the targeted devices was implemented and evaluated.**

***Index Terms*—Battery Management System, Security, Sensor, Near Field Communication, Anti Counterfeiting.**

I. Introduction

Over the years, Battery Management Systems (BMS) have seen an increased interest in the research community, primarily due to the higher digitization and use in different applications. They play an important role in many systems but are often mentioned today as battery control devices used in smart power grids and electric or hybrid vehicles [1]. BMS are mainly used to handle the balancing of large battery cells during charging & discharging cycles, as well as to offer diagnostic services to track their lifetime usage [2].

A BMS can be deployed in different topologies and usually consists of various devices. They generally contain a central BMS controller, which in a modulated setting, communicates with individual Battery Cell Controllers (BCC). The BCCs help in relaying diagnostic data back to the BMS controller through monitoring and control of individual battery cells. Data received from these sensors is critical in preventing dangerous incidents like the thermal runaway, which happens due to the rapid increase in battery temperature [3]. However, a BMS controller can only act as long as it has the correct information on the current state inside the battery pack, i.e., it is dependant on the sensor readouts. Two main factors influence the accuracy of these readings: (i) the number of sensors used, and (ii) the relative position of the sensors to their target of interest.

Commonly, BMS use wired connection to handle the communication with battery cell sensors. This imposes three main limitations however:

1) *Assembly cost:* Each connection to and from the cell and sensor source needs to be physically soldered, and it needs to account for materials used.
2) *Scalability:* The complexity of design, deployment, and afterwards maintenance, boosts with the increase in the number of battery cell sensors.
3) *Area coverage:* Due to the physical wires used, sensor placement is often limited. This can make the use of certain areas impossible, e.g., inner housing of the cells.

In order to alleviate the aforementioned wired limitations, it is possible to replace wired with wireless technology networks. However, we see several challenges that need to addressed when choosing an appropriate wireless technology:

- *Restricted data throughput*: For a safe and continuous execution of BMS operations, it is necessary to maintain a steady and fast flow of data from the battery cell sensors to the BCCs and afterwards to the BMS controller.
- *Interference*: The use of the same frequency band, even through different communication technologies, will cause interference. This is especially true with the 2.4 GHz band, which is used by several technologies, most prominent being LR-WPAN, Bluetooth, ZigBee, and WiFi.
- *Multipath propagation*: Maintaining communication sight and reliability under strict and obstructive environments.
- *Security concerns:* Unless placed in an enclosed case, wireless networks are prone to eavesdropping, remote attacks, and other malicious incursions [4].

As an answer to the mentioned obstacles, we propose the realization of the communication between battery cell sensors and BCCs to be done using NFC. By employing NFC, we are not only able to answer to the design restrictions imposed through the use of the wired communication but also address the challenges introduced when using wireless communication. Furthermore, we address a general safety BMS requirement centred around the battery cells source validity [5], [6]. It is important that only battery cells that come from valid and approved manufactures are installed, as inadequate battery cells could potentially lead to hazards that cannot be otherwise mitigated by the BMS controller.

Contributions. Summarized, our main contributions contained in this paper are: (i) We present an NFC-based approach that can be used for BMS sensor readout, specifically the communication between the battery cell sensors and the battery cell controllers. (ii) To counter potential malicious and counterfeiting attempts, we provide a security solution that can be easily integrated. (iii) The implementation of the proposed design and evaluation of its utility using a BMS test system.

II. Background and Related Work

With the increase of the number of battery cells in modern BMS, new topologies and architectures had to be introduced. A focus was set in using derivations of modular and distributed BMS [2]. This all further lead to an increase in expenses and complexity in cable installation. Different models have been proposed based on the wireless technology used, such as the use of Bluetooth [7], [8], and ZigBee [9]. They primarily focus on the communication between the BCCs and the main BMS controller using these wireless technologies. We extend the wireless usage by focusing on the BCC and sensor communication through NFC utilization.

The use of the NFC technology in more extensive system infrastructures has already been investigated before. Specifically, research presented by Ulz et al. [10] proposes the use of NFC-based communication for robot-machine interaction in an Industry 4.0 setting. Additionally, work by Chen et al. [11] investigates secure authentication and anti-counterfeiting methods using RFID. Alzahrani et al. [12] proposes an NFC-focused anti-counterfeiting system. Despite a large amount of research being done both for the general wireless BMS and the integration of NFC in similar environments, not much specific work has yet been done that combines these two fields of interest. Work done by Schneider et al. [13] focuses largely on this field by also proposing a design approach for wireless BMS battery sensors utilizing the same RFID technology. However, one of the main focal points in that paper is placed on the issues caused by galvanic isolation. Moreover, due to the date when the paper was published, it does not account for the newer BMS modular architectures and modern NFC derivations, alongside the security aspects. In this work, we try to bridge that gap and show the potential of using NFC in hard-to-reach sensor environments while at the same time giving attention to the security requirements.

III. Design of the Novel BMS NFC Sensor Readout

For our targeted design architecture we divide the entire system into three main modules: (i) a *BMS controller*, (ii) a *Cell control board*, (iii) and a *Battery module*. The BMS controller can either contain one or multiple different *Microcontroller Units* (MCUs) set for the overall BMS control and status monitoring. It communicates with the cell control board that contains a BCC, the NFC reader as the communication interface, and optionally an additional control MCU for the protocol handling. The battery module contains battery cells, sensors, and NFC communication interface to the cell board. In our case, this interface is a *NFC-Tag* (NTAG). This is illustrated in Figure 1. For charging & discharging cycles, as well as related voltage readings, we still rely on the hardwired measurements from conventional modulated BMS designs.

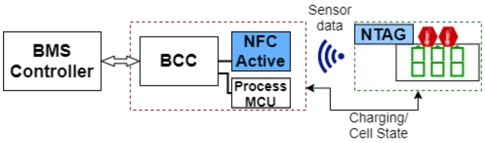


Fig. 1. Proposed BMS modular design architecture utilizing NFC components.

A. NFC Communication

For the BCCs and battery cells to be able to communicate using the wireless NFC technology, appropriate devices and communication mode need to be chosen. In the presented design, this is done over the *Reader/Writer mode*. The NFC reader is the active device that is connected to a specialized controller BCC, as well as to an MCU for pre-processing and security operations. Before the communication begins, the NFC reader needs to have discovered the necessary NTAG(s) using a discovery loop. Afterwards, the authentication process starts. Following it, the NTAG proceeds to initiate self-configuration and prepares to communicate with both the sensor and the NFC reader. Since in a standard environment, the same devices are going to be also used for the subsequent measurement readings, the initialization and configuration steps can be cached and therefore omitted.

B. Energy Harvesting and Positioning

A disadvantage that NFC has over most other wireless technologies is its relatively short range. This is of no issue in the presented design, as the BCCs and battery cells are usually tightly packed and installed together. The NFC in our design uses the energy harvesting feature to power up the NTAG from the reader. This feature limits the distance between the antennas. Depending on the environment, the distance peaks approximately at $5.4\,cm$. For a feasible communication and optimal initialization time, we opted to use a distance of $2\,cm$. As both the sensor and the NTAG reside on the battery module, it would be possible for them to be directly powered as it is done in a conventional design. However, this characteristic is not present in our design, as using the wiring to the battery modules would violate one of the design requirements set on reducing the extent of the necessary wires.

C. Authentication Protocol

In terms of security, NFC's advantage over the use of other wireless technologies is in both its short range and frequency band. This property limits the list of technologies that a potential attacker could use to attack the system. Since battery modules are usually enclosed in a protective case together with a BCC, this means that the main potential attack vector on these modules would be one through counterfeiting.

To be able to securely verify that the battery modules are valid, we integrate the use of an authentication protocol in our

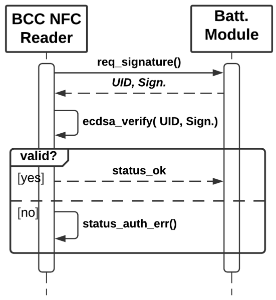


Fig. 2. Sequence diagram of the authentication protocol.

design. This process is achieved by verifying a value that needs to be unique to each device. Since NTAGs are usually shipped with a Unique Identifier (UID) value, we can use it as an input for an Elliptic Curve Digital Signature Algorithm (ECDSA). In our design, we use the *secp128r1* protocol as the Elliptic Curve (EC) function, having a good balance between the performance and output sizes. The signature value, which is calculated with a private key during the manufacturing process or subsequently updated, is then stored in a protected memory space located on the NTAG chip. The BCC needs to have access to the public key, either it being pre-embedded or accessed through other secure channels. The authentication protocol is shown in Figure 2. Before the signature verification takes place, the UID validity is first checked against the list of valid devices.

IV. Evaluation

A. Test System Implementation

For the purpose of testing our design approach, we implemented a test suite that contains the necessary BMS modules, as well as the additional NFC equipment. We aimed to use the NFC modules, which support the latest *NFC Type 5 Tag* technology. Furthermore, the used components are automotive-graded where applicable for the purpose of replicating a real-world use-case as closely as possible. To that end, all devices used, except for the temperature sensor, come from the NXP Semiconductors lineup of products.

As the main BMS controller, we use an S32K144 MCU board. It communicates with the cell control board via the FRDMDUAL33664 shield. It is further connected to an RD33771CDST that houses an MC33771C which functions as a BCC. The cell control board contains an automotive NFC Reader for handling the NFC transmissions and another S32K144 as the MCU for programming and testing. The battery module consists of a BATT-14CEMULATOR that serves as a battery emulator, an NTAG component as the passive NFC device, and a BMP180 temperature sensor. The BCC is able to receive the emulated cell voltage data from the battery emulator, while the temperature sensor data is sent through the NFC interface. Both the temperature and the cell voltage data are first received by the BCC and then transmitted to the BMS controller. For the authentication protocol, we base our implementation on the *originality signature* feature found on the NXP's RFID devices. Signature calculation and verification are handled via the *ecc-nano* library. Elements of the development and evaluation were handled in a recent master's thesis [14]. The system is shown in Figure 3.

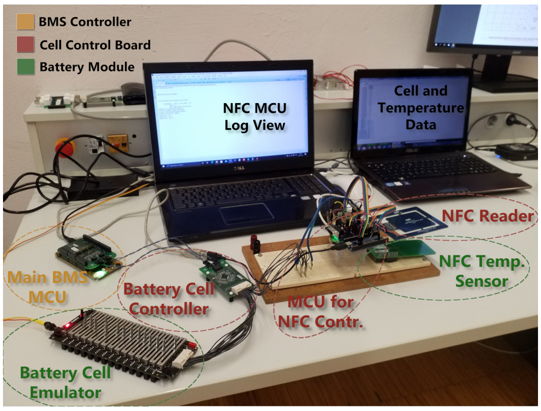


Fig. 3. Test setup for the BMS NFC sensor readout.

B. Time Measurements

We divide the entire BMS monitoring process into two phases: (i) *Initialization phase:* executed only once for device preparation and configuration, and (ii) *Monitoring phase:* continuous action that is called on every sample step to measure and retrieve sensor and cell data. Individual steps, as well as their time measurements, are shown in Figure 4.

We start the process after the NTAGs have already been discovered. As the first step the authentication protocol is run. This protocol run includes both sending an authentication request from the NFC reader, the response from the NTAG, and the verification calculation on the MCU that is connected with the NFC reader. The authentication step showed an average time of $369.3\,ms$, with majority of it being spent on the verification process. With the NTAG verified, the energy harvesting check is handled which lasts for $19.64\,ms$. Finally, the NTAG operation initialization is run which measured $29.16\,ms$, followed with the sensor initialization that took $116.1\,ms$.

After the initialization phase is finished, there is no need to reconfigure the devices during the system run. For the monitoring phase, sensor measurements are read and transmitted to the BCC using NFC communication. This phase is repeatable, with each action showing a time of $27.2\,ms$.

C. Security Threat Analysis

To evaluate the achieved security protection, a threat analysis was conducted. We have used *Data Flow Diagram* (DFD) to illustrate the system model as seen in Figure 5. Here, we demonstrate a summarized representation of the analysis based on *Threats* (T), *Assets* (A), and *Countermeasures* (C), as well as the points of their potential impact. In our security model, we argue the following assumptions: (i) A battery module can only be communicated with via an adequate BCC, (ii) Both the cell control board and the battery module are enclosed in a chassis and the external communication can only be achieved through the BMS controller, (iii) Every newly added and unknown battery module is considered untrustworthy.

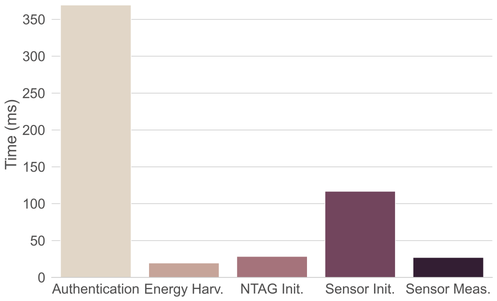


Fig. 4. Time measurement results for the Initialization phase (Authentication, Energy Harv., NTAG Init., Sensor Init.) and Monitoring phase (Sensor Meas.).

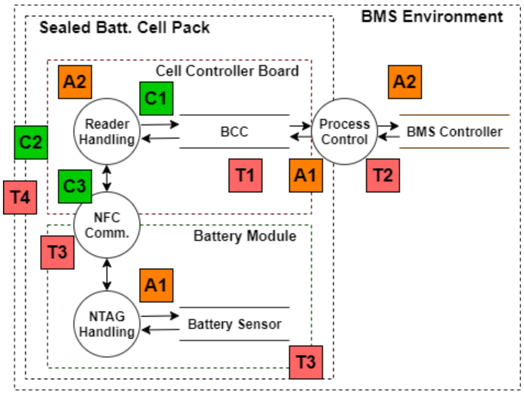


Fig. 5. Data Flow Diagram representation of the security analysis.

In our use-case, we have two assets that we want to protect: (A1) *Sensor (status) data*: data retrieved from the cell sensors, and (A2) *System integrity*: both hardware and software integrity. To disturb the cell balancing control, we indicate (T1) *Battery control obstruction* as a potential threat. Further, an attacker might, through data modification, try to (T2) *Tamper with BMS status messages*. Both of these threats are mitigated with the implemented (C1) *Authentication through signature validation* countermeasure. Here, BCCs check and validate every individual battery module, ensuring that the BMS controller only receives authorized status messages. Another possibility for an attacker would be to gain a (T3) *Backdoor access* through either the NFC interface or a counterfeited battery module. This is again protected with (C1), but also through reducing the attack proximity by relying on the (C3) *NFC physical layer characteristics*. Lastly, a (T4) *Remote attack* could also be launched from the outside using wireless communication. In this case, (C2) *Cell pack sealing* protects against remote attacks by isolating interfaces via material shielding. Also, (C3) would hamper the possibility of such an attack through frequency spectrum and range limitations.

V. Conclusion and Future Work

In this paper, we present the idea of using NFC as a wireless communication interface for battery sensor readouts in BMS. An authentication model was proposed and evaluated to alleviate the risk of the counterfeited battery cells and prevent safety and security threats. Experimental results using real components showed the feasibility of our approach but also design challenges related to the antenna and sensor placement. For the future work, we plan to evaluate the design using different antenna orientations, optimize execution time, and also consider a security protocol extension.

Acknowledgment

This project has received funding from the “EFREtop: Securely Applied Machine Learning - Battery Management Systems” (Acronym “SEAMAL BMS”, FFG Nr. 880564).

References

[1] R. Xiong, J. Cao, Q. Yu, H. He, and F. Sun, “Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles,” *IEEE Access*, vol. 6, pp. 1832–1843, 2018.

[2] D. Andrea, *Battery Management Systems for Large Lithium-ion Battery Packs*. EBL-Schweitzer, Artech House, 2010.

[3] P. Sun, R. Bisschop, H. Niu, and X. Huang, “A Review of Battery Fires in Electric Vehicles,” *Fire Technology*, pp. 1–50, 01 2020.

[4] K. Lounis and M. Zulkernine, “Attacks and Defenses in Short-Range Wireless Technologies for IoT,” *IEEE Access*, vol. 8, 2020.

[5] A. Khalid, A. Sundararajan, A. Hernandez, and A. I. Sarwat, “FACTS Approach to Address Cybersecurity Issues in Electric Vehicle Battery Systems,” in *2019 IEEE TEMSCON*, pp. 1–6, 2019.

[6] S. Engels, “Counterfeiting and piracy: the industry perspective,” *Journal of Intellectual Property Law & Practice*, vol. 5, 05 2010.

[7] C. Shell, J. Henderson, H. Verra, and J. Dyer, “Implementation of a Wireless Battery Management System (WBMS),” in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pp. 1954–1959, 2015.

[8] G. De Maso-Gentile, A. Bacà, L. Ambrosini, S. Orcioni, and M. Conti, “Design of CAN to Bluetooth gateway for a Battery Management System,” in *2015 12th International Workshop on Intelligent Solutions in Embedded Systems (WISES)*, pp. 171–175, 2015.

[9] A. Rahman, M. Rahman, and M. Rashid, “Wireless Battery Management System of Electric Transport,” *IOP Conference Series: Materials Science and Engineering*, vol. 260, p. 012029, nov 2017.

[10] T. Ulz, T. Pieber, C. Steger, S. Haas, and R. Matischek, “Sneakernet on Wheels: Trustworthy NFC-based Robot to Machine Communication,” in *2017 IEEE International Conference on RFID Technology Application (RFID-TA)*, pp. 260–265, 2017.

[11] C.-L. Chen, Y.-Y. Chen, T.-F. Shih, and T.-M. Kuo, “An RFID Authentication and Anti-counterfeit Transaction Protocol,” in *2012 International Symposium on Computer, Consumer and Control*, pp. 419–422, 2012.

[12] B. A. Alzahrani, K. Mahmood, and S. Kumari, “Lightweight Authentication Protocol for NFC Based Anti-Counterfeiting System in IoT Infrastructure,” *IEEE Access*, vol. 8, pp. 76357–76367, 2020.

[13] M. Schneider, S. Ilgin, N. Jegenhorst, R. Kube, S. Püttjer, K.-R. Riemschneider, and J. Vollmer, “Automotive Battery Monitoring by Wireless Cell Sensors,” in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 816–820, 2012.

[14] M. Gaertner, “Design and Implementation of a NFC-based Solution for Secure Battery Management Systems,” Master’s thesis, Graz University of Technology, 2021.

A.3 [C] Secure and Trustworthy NFC-Based Sensor Readout for Battery Packs in Battery Management Systems

F. Basic, M. Gaertner and C. Steger, “Secure and Trustworthy NFC-Based Sensor Readout for Battery Packs in Battery Management Systems,” in *IEEE Journal of Radio Frequency Identification*, vol. 6, pp. 637-648, IEEE, 2022.

Abstract. Wireless battery management systems (BMSs) are increasingly being considered for modern applications. The ever-increasing complexity and production costs of BMS modules and wired connections resulted in a necessity for new ideas and approaches. Despite this growing trend, there is a lack of generic solutions focused on battery cells’ sensor readout, where wireless communication allows for a more flexible and cost-efficient sensor installation in battery packs. Many wireless technologies, such as those that use the 2.4 GHz frequency band, suffer from interference and other limitations. In this article, we present an alternative approach to communication in BMS that relies on the use of near field communication (NFC) technology for battery sensor readouts. As an answer to the rising concern over the counterfeited battery packs, we consider an authentication schema for battery pack validation. We further consider security measures for the processed and stored BMS status data. To show that a general BMS application can make use of our design, we implement a BMS demonstrator using the targeted components. We further test the demonstrator on the technical and functional level, by also performing evaluation on its performance, energy usage, and a security threat model.

My Contribution. As the extended journal paper to the *Publication B*, I contributed by the written text, vastly extending the original concepts and design provided in the original paper. Additionally, I provided the main design chapters concerning the security processing and data structuring between the main BMS controller and the BPC. I also conducted and provided additional evaluations concerning data overhead, data throughput, and energy measurements. The extended design, implementation, and evaluation tests have been conducted on the original prototype and concept also provided and supported by Martin Gaertner. Christian Steger provided inputs on the paper extension, and evaluation guidance.

Secure and Trustworthy NFC-based Sensor Readout for Battery Packs in Battery Management Systems

Fikret Basic, Martin Gaertner, Christian Steger
Institute for Technical Informatics
Graz University of Technology
Graz, Austria
{basic, steger}@tugraz.at

***Abstract*—Wireless Battery Management Systems (BMS) are increasingly being considered for modern applications. The ever-increasing complexity and production costs of BMS modules and wired connections resulted in a necessity for new ideas and approaches. Despite this growing trend, there is a lack of generic solutions focused on battery cells' sensor readout, where wireless communication allows for a more flexible and cost-efficient sensor installation in battery packs. Many wireless technologies, such as those that use the 2.4 GHz frequency band, suffer from interference and other limitations. In this article, we present an alternative approach to communication in BMS that relies on the use of Near Field Communication (NFC) technology for battery sensor readouts. As an answer to the rising concern over the counterfeited battery packs, we consider an authentication schema for battery pack validation. We further consider security measures for the processed and stored BMS status data. To show that a general BMS application can make use of our design, we implement a BMS demonstrator using the targeted components. We further test the demonstrator on the technical and functional level, by also performing evaluation on its performance, energy usage, and a security threat model.**

***Index Terms*—Battery Management System, Security, Sensor, Wireless, Near Field Communication, Anti Counterfeiting.**

I. Introduction

GREEN energy and sustainability are becoming more important than ever before, with Battery Management Systems (BMS) also seeing an increased interest from the industry and the research community. This resulted in a higher digitization and use-case expansion that required additional attention to the already complex systems that utilize BMS [1]. BMS play an important role in many systems today that rely on the use of large battery packs. They are often mentioned as one of the main critical controller components being part of smart power grids and electric or hybrid vehicles [2], [3]. They are used as control devices in such systems, where they regulate the usage of individual battery cells by offering monitoring and diagnostic services, as well as the possibility to track the lifetime usage of each individual cell [4]. To offer safer and more efficient energy usage, a BMS also handles cell balancing control during the charging and discharging cycles. A BMS can be deployed in different topologies and usually consists of various devices. They generally contain a central BMS controller, which in a modulated setting, communicates with individual Battery Cell Controllers (BCCs). The BCCs help in relaying diagnostic data back to the BMS controller through monitoring and control of individual battery cells. These cells are packed together in parallel or serial connections inside battery modules, with accompanying temperature, pressure, or other sensors. Data received from these sensors is critical in preventing dangerous incidents like the thermal runaway, which happens due to the rapid increase in battery temperature [5]. A BMS controller is able to derive diagnostic data based on the monitored and measured data from the battery cells, e.g., State of Charge (SoC), State of Health (SoH), etc., [1]–[3]. However, these controllers can only act as long as they have the correct information on the current state inside the battery pack, i.e., they are dependent on the sensor readouts. Two main factors influence the accuracy of these readings: (i) the number of sensors used, and (ii) the relative position of a sensor to its measurement target.

Traditionally, BMS use wired connection to handle the communication between individual modules, i.e., between the BCCs and the battery cell sensors. This, however, imposes several limitations, as shown in Table I. In order to alleviate the aforementioned wired limitations, it is possible to replace wired with wireless technology networks. We see several challenges that need to addressed when choosing an appropriate wireless technology, as indicated in Table II.

Different communications have already been tested in an attempt to solve the mentioned limitations. Research with Bluetooth [6], [7] and ZigBee [8] have been tested and evaluated within the BMS domain. However, while they give promising results for the data throughput, these studies fail to address the main challenge of the mentioned technologies, that being the interference. Security is also only partially covered, mostly under the given technologies' security stack, with ZigBee being especially subjected to limited throughput and security concerns [9]. Schneider et al. [10] address most of the concerned challenges but does not focus on the security aspects and newer modulated BMS considerations. We further discuss the BMS wireless and security findings, and their relevance to our work, in Section II.

To address the BMS wired limitations and wireless requirements, we propose a system architecture for modular BMS that offers the NFC technology for battery module cells' sensor readout. This includes the extension with the conventional BCC by adding a connection to an active NFC reader. The battery cell's sensors would, hence, only connect to the provided passive NFC device per battery pack module,

TABLE I
LIMITATIONS OF THE WIRED BMS [9], [11]–[14]

Limitation	Description
Assembly cost	Each connection to and from the cell and sensor source needs to be physically soldered, and it needs to account for materials used.
Scalability	The complexity of design, deployment, and afterwards maintenance, boosts with the increase in the number of battery cell sensors.
Area coverage	Due to the physical wires used, sensor placement is often limited. This can make the use of certain areas impossible, e.g., inner housing of the cells.

TABLE II
WIRELESS TECHNOLOGY CHALLENGES FOR BMS

Challenge	Description
Restricted throughput	For a safe and continuous execution of BMS operations, it is necessary to maintain a steady and fast flow of data from the battery cell sensors to the BCCs and afterwards to the BMS controller [9], [14].
Interference	The use of the same frequency band, even through different wireless technologies, can cause interference. This is especially true with the 2.4 GHz band, which is used by several technologies, most prominent being LR-WPAN, Bluetooth, ZigBee, and WiFi [6]–[8], [15].
Multipath propagation	System's resilience in maintaining communication sight and reliability under obstructive environments [16].
Security concerns	Unless placed in an enclosed case, wireless networks are prone to eavesdropping, remote attacks, and other malicious incursions [17]–[19].

not requiring additional connection or power draw for their functionality. Security and data processing are handled via an additional Microcontroller Unit (MCU). These additional components would form a new overall control block together with the BCC. This block would still remain modulated, i.e., it would maintain the same input and output connections. A BMS with our presented architecture is able to perform security operations on the logged status data with a minimal overhead increase, while retaining its original functionality.

Contributions: In this work, we present an answer to the listed challenges, by proposing a design model that utilizes NFC as the chosen technology for the wireless communication between the battery cell sensors and the BCCs. By introducing NFC, we are not only able to answer to the design restrictions imposed through the use of wired communication, but also address the challenges introduced when using wireless communication technologies.

After providing the relevant background information and presenting related work in Section II, we make the following contributions in this article:

- We present a novel approach for the NFC-based BMS battery cells' sensor readout by indicating design points for the device use and placement, as well as the exchange protocol between the modules (Section III).
- We address the battery cell source validity [20], [21] question by proposing an authentication model for verifying individual battery cell modules. BCCs should communicate only with the trusted battery cells, for the reasons of both security and safety concerns (Section IV-A).
- We investigate the security protocol and design requirements for the purpose of storing and securely handling the derived BMS status data as the next operational step after the sensor readout (Section IV-B & IV-C).
- We experimentally show the feasibility of our design by realizing a BMS system prototype and implementing NFC and security control functionalities. The system is further evaluated on its (i) security dependability by a threat model analysis, (ii) time measurements for individual BMS NFC readout phases, (iii) protocol overhead analysis for the secure BMS monitoring and diagnostic data logging (Section V), (iv) system energy consumption, and (v) potential NFC sensor readout throughput.

This article presents an extended version of the published paper [22] that includes a more detailed analysis and investigation of the proposed design specifications related to the NFC integration for the BMS inter-module communication and sensor data readout. On top of the design and authentication approach presented in that paper, an additional security investigation and evaluation for the purpose of securely logging sensor monitoring and diagnostic BMS data were conducted. Additionally, in relation to the NFC system design, a throughput and energy consumption analysis have been done as well.

II. BACKGROUND AND RELATED WORK

A. Wireless Battery Management Systems (WBMS)

The increase of the number of battery cells in modern BMS resulted in an increase in the number of used component devices, especially regarding intermediate control components. This all further lead to an increase in expenses and complexity in cable installation. New topologies and architectures had to be introduced focused on using wireless technologies. Primarily, they were seen as an extension to already different derivations of modular and distributed BMS [4]. Some of the pioneering research includes the realization of a WBMS under custom chips and protocols by M. Lee et al. [12]. In this work, they introduce a WiBaAN protocol that works under the 900 MHz band with a data rate of up to 1 Mbit/s, allowing for direct communication between a large set of battery cells and the main BMS controller. However, while novel for the time of publishing, the relatively low data throughput rate, used frequency band, manufacturing costs, and no newer research updates regarding the modulated BMS topologies could present a limitation for the modern BMS derivations.

Several design models have been proposed and investigated in the domain of the 2.4 GHz frequency band. Shell et al. [7] presents a Bluetooth-based BMS design approach. They show its feasibility under the standard BMS environment and commercial applications. De Maso-Gentile et al. [6] presents a different design approach, that is more focused on applying Bluetooth gateway access to already conventional BMS CAN infrastructures. However, most of the proposals based on Bluetooth technology are primarily centred on intra-module communication and do not account for direct battery sensor readout. Bluetooth, specifically the newer BLE, has a limited throughput rate which can often fluctuate due to noisy channels even in the newer 5.x standards [23]. This can make it difficult to fulfill the necessary standard requirements for data transfer under the conventional BLE topologies. Research has been also conducted using the ZigBee technology by Rahman et al. [8]. While it showed potential in its applicability, ZigBee would suffer from restrictions due to its low-data rates and unstable channels. Wi-Fi was also considered under specialized BMS investigations. Gherman et al. [15] propose a WBMS build on a single chip that used Wi-Fi as its communication technology for their demonstrator. A different kind of research, more focused on smart cells, was proposed by Huang et al. [24]. Here, the communication between the individual cells and the main controller is done over a Wi-Fi channel, with the BMS controller using the channel for the cell balancing control. The focus of this research was on cell balancing and smart cells, with Wi-Fi being mostly used as a demonstrative wireless technology with no significant focus on the wireless aspects and challenges. The presented research gives an insight into the communication between the BCCs or similar modules and the main BMS controller using the prescribed wireless technologies. Also, as mentioned in Section I, 2.4 GHz technologies generally suffer from an increased chance of interference under complex environments, e.g., Electric Vehicles (EVs), where many devices and modules could compete over the use of the bandwidth channels. This work extends the wireless usage regarding the BMS components by also focusing on the BCC and sensor communication, which is often overlooked, through NFC utilization.

BMS today are also often considered for the cloud service extensions [25], [26]. These solutions offer the distribution of BMS modules over a wider area, and hence, further reduce the use of wires and deployment complexity. They also provide functionality extensions. Cloud services aim to cover the calculation of important State of Health (SoH) and State of Charge (SoC) BMS functions on a more efficient cloud base, by using different data sources and even resource-demanding machine learning algorithms, which otherwise would not be possible on resource and process constrained BMS MCU field controllers. These services, however, are outside of the scope of this work, as they focus mainly on the external, rather than on the internal BMS communication.

B. Security in Battery Management Systems (BMS)

The current research work related to BMS security is limited, due to it being a relatively novel topic that first started sparking interest in the recent time. Nonetheless, there has already been some research done focused on different aspects of the BMS security design. Sripad et al. [18] present an investigation of the cybersecurity threats of BMS, particularly of EVs, especially related to their interaction with battery packs and to overcharging and discharging manipulation concerns. A FACTS approach proposed by Khalid et al. in [20] deals with a formal threat analysis of BMS by investigating and comparing different existing frameworks. It also goes into a detailed analysis, points out and classifies important general security threats found under a BMS. Further BMS threat analysis models have also been proposed by Kumbhar et al. [19]. This work also goes in a direction of a wider topic and includes some security overview of BMS Internet-of-Things (IoT) solutions. A similar work that looks at the IoT security perspective with BMS and their related environments is by Lopez et al. [27]. While most of the mentioned publications present a broad BMS security analysis topic, they still serve as a good starting ground to complement the presented work.

C. Near Field Communication (NFC) Applications

NFC is a high frequency (HF) communication standard based on the Radio Frequency Identification (RFID) which operates on a frequency band of 13.56 MHz, has a typical range of up to $10\,cm$, and depending on the standard, supports data rates of up to 848 kbit/s [28], [29]. It handles different modes of communication, among them being communication between an active reader and a passive tag device. Like the RFID, it supports the energy harvesting features from the active to the passive device during the data exchange. The use of the NFC technology in more extensive system infrastructures has already been investigated before. Specifically, research presented by Ulz et al. [30] proposes the use of NFC-based communication for robot-machine interaction in an Industry 4.0 setting. Additionally, work by Chen et al. [31] investigates secure authentication and anti-counterfeiting methods using RFID. Alzahrani et al. [32] propose an NFC-focused anti-counterfeiting system. Despite a large amount of research being done both for the general wireless BMS and the integration of NFC in similar environments, not much specific work has yet been done that combines these two fields of interest, which is also indicated by the recent survey research paper by A. Samanta and S. S. Williamson [9]. Work done by Schneider et al. [10] focuses largely on this field by also proposing a design approach for wireless BMS battery sensors utilizing the same RFID technology. However, one of the main focal points in that paper is placed on the issues caused by galvanic isolation. Moreover, due to the date when the paper was published, it does not account for the newer BMS modular architectures and modern NFC derivations, alongside the security aspects. In this work, we try to bridge that gap and show the potential of using NFC in hard-to-reach sensor environments while at the same time giving attention to the security requirements.

III. Design of the Novel BMS NFC Sensor Readout

For the targeted design architecture we divide the entire system into three main modules:

- *BMS controller*
- *Cell control board* (CCB)
- *Battery module*

Modules, as well as their placement and connections, are illustrated in Fig. 1. Here, the BMS controller plays the role of the main control unit responsible for receiving and interpreting diagnostic data and conducting necessary safety control actions. It can contain one or multiple operational

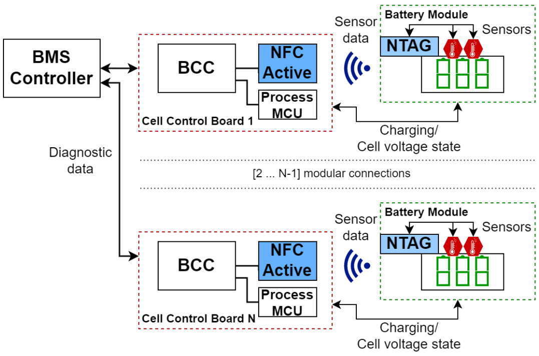


Fig. 1. Proposed BMS modular design architecture utilizing NFC components.

MCUs. The BMS controller communicates with the CCB that contains a BCC, an NFC reader as the communication interface, and optionally, an additional control MCU for the protocol handling connected via a supported communication bus protocol, e.g., Serial Peripheral Interface (SPI). In a traditional design, the CCB usually only contains a simple BCC aimed primarily only for the BMS functional support. To supplement the communication handling requirements for the NFC reader, and also preceding and subsequent data and security processing, we introduce another process MCU that works either along with the BCC chip or on top of the BCC functional block. In most scenarios, the communication between the CCBs and the BMS controller is established using either a Controller Area Network (CAN) protocol, or some other form of network connection, like Ethernet, Transformer Physical Layer (TPL), or SPI [33].One BMS controller can communicate with multiple CCBs, depending on the system and protocol limitations [1], [4].

The battery module contains battery cells, sensors, and an NFC communication interface to the CCB. In the presented design, this interface is an *NFC-Tag* (NTAG). The communication for the NTAG and sensors is primarily done with the Inter-Integrated Circuit (I2C) protocol. For charging and discharging cycles, as well as related voltage readings, we still rely on the hardwired measurements from conventional modulated BMS designs, with them being usually less demanding in terms of placement and installation compared to the investigated sensor connections and also requiring an otherwise special handling.

A. NFC Communication

To make the communication between the BCCs and battery cells using the wireless NFC technology possible, appropriate devices and communication modes need to be chosen. In the presented design, *Reader/Writer mode* is opted as the chosen mode of communication. The NFC reader plays the role of the active device that is connected to a specialized controller BCC, as well as to an MCU for pre-processing and security operations. In traditional designs of the modulated BMS, this MCU can also be already found as an integral part of the BCC. It is, however, vital, that the main functionality conditions are fulfilled and contained which entail that the communication over the NFC can be accurately processed and handled, as well as to be able to handle security operations. Before the communication begins, the NFC reader needs to have discovered the targeted NTAG(s) using the discovery loop process. Immediately afterwards, the authentication process starts. It is important to make sure that no formal communication can begin before the battery modules have been authenticated, as to avoid any potential vulnerabilities that might arise afterwards. Following the successful authentication, the NTAG proceeds to initiate self-configuration and prepares to communicate with both the sensors and the NFC reader. Since in a standard environment, the same devices are going to be also used for the subsequent measurement readings, the initialization and configuration steps can be cached and therefore omitted.

B. Energy Harvesting and Positioning

A disadvantage that NFC has over most other wireless technologies is its relatively short range. This is of no issue in the presented design, as the BCCs and battery cells are usually tightly packed and installed together. The NFC in the presented design uses the energy harvesting feature to power up the NTAG from the reader. The energy harvesting is also additionally used to power the necessary readout of the adjacent sensor. This feature limits the distance between the antennas. Depending on the environment, the distance peaks approximately at $5.4\,cm$. For a feasible communication and optimal initialization time, we opted to use a distance of $2\,cm$. The NTAG is not powered right at the boot-up of the system. It first needs to check if enough energy can be received from the present NFC field. The energy harvesting needs also to match the internally pre-configured voltage level. A voltage level of up to $3\,\mathrm{V}$ can be supplied, which was also deemed sufficient for the sensor readout operation. As both the sensor and the NTAG reside on the battery module, it would be possible for them to be directly powered as it is done in a conventional design. However, this characteristic is not present in our design model, as using the wiring to the battery modules would violate one of the design requirements set on reducing the extent of the necessary wires.

C. Data Exchange Protocol

The NFC reader is intended to establish the wireless connection to the dedicated battery sensors of the battery module. It plays the role of the active device, meaning that it initiates the communication. The sensors are able to transmit their values to the passive NTAG using the I2C connection. In this scenario, the master mode is used and the NTAG takes the role of the adapter module. The data is passed directly between the sensors and the NFC interface. Static Random Access Memory (SRAM) storage is used for the intermediate data placement before the read operation takes place. Additional commands had to be provided for the interaction on the battery module's I2C bus, as well as for the data transmission. These include the: (i) I2C read & write commands, and (ii) content read; which allows direct content read from the intermediate SRAM storage. No MCU or any additional component is needed here, making the design relatively simple and cheap.

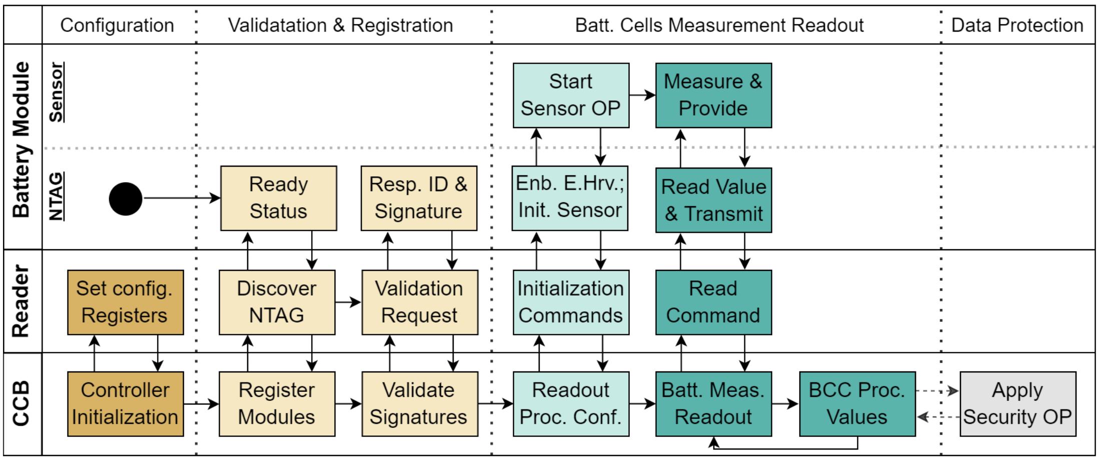


Fig. 2. General system design representation using a swimlane sequence diagram showing the communication flow between CCB with its NFC Reader, and the battery module with NTAG and sensor. The process follows three main operational steps, with the security operations appliance being an optional step.

Fig 2. shows a swimlane sequence diagram that encapsulates all main operation processes intended to be covered during a common data exchange run. It covers the following steps:

1) Configuration: initialization step at the session start, intended for loading up all the necessary configuration and operational material. It is expected to be run only once, usually at a start-up of a system (e.g., start of a vehicle). However, certain options could be cached, and hence pre-configured, with the aim of reducing the overall process execution time.
2) Validation & registration: CCB instructs its NFC reader to find and assign NTAGs first by using a discovery loop. Afterwards, validation takes place using the proposed signature authentication algorithm described more in detail in the Section IV-A.
3) Battery cells measurement readout: starts with the initialization step aimed for the measurement configuration. During this one-time procedure, the NTAG initializes its communication with the sensors, but also enables the energy harvesting feature covered in Section III-B. After the initialization is finished, a process loop is run that, based on the sampling time, periodically reads out and processes the battery cell measurement data. The cells' data are further covered by the conventional BCC monitoring and diagnostic operations.
4) Data protection: an optional step for the purpose of securing the read measurement, and BCC-derived, data. These operations are discussed in Sections IV-B & IV-C. If used, it is intended to be included together with the measurement process loop.

IV. Security Mechanisms

A. Battery Module Authentication Protocol

In terms of security, NFC's advantage over the use of other wireless technologies is in both its short range and frequency band. This property limits the list of technologies that a potential attacker could use to attack the system. Since battery modules are usually enclosed in a protective case together with a BCC, the main potential attack vectors on these modules would be the ones initiated through counterfeiting [27]. It is important that only battery cells that come from valid and approved manufactures are installed, as inadequate battery cells could potentially lead to hazards through compromising the BMS controller, or even going higher to the high-speed network outside the BMS environment [33].

To be able to securely verify that the battery modules are valid, we integrate the use of an authentication protocol in our design. This process is achieved by verifying a value that needs to be unique to each device. Since NTAGs are usually shipped with a Unique Identifier (UID) value, we can use it as an input for an Elliptic Curve Digital Signature Algorithm (ECDSA). In our design, we use the *secp128r1* protocol as the Elliptic Curve (EC) function, having a good balance between the performance and output sizes. The signature value, which is calculated with a private key during the manufacturing process or subsequently updated, is then stored in a protected memory space located on the NTAG chip. The BCC needs to have access to the public key, either it being pre-embedded or accessed through other secure channels. The authentication protocol is shown in Fig. 3. Before the signature verification takes place, the UID validity is first checked against the list of valid devices. Failure in either can lead to a warning message presented through the BMC controller, or a complete shutdown of the system, depending on the targeted use-case.

B. BMS Status Data Protection

To protect the transmitted battery sensor data and the derived diagnostic data, it is necessary to apply different security measures. These measures would present an answer to the aforementioned security requirements and would be handled as an extension to the current BMS communication design, but also to its data acquisition protocol. Primarily,

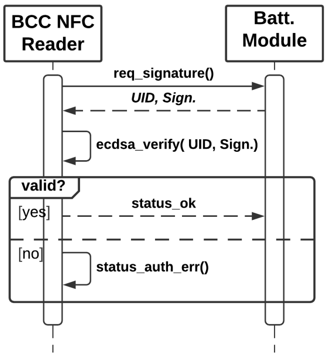


Fig. 3. Sequence diagram of the authentication protocol.

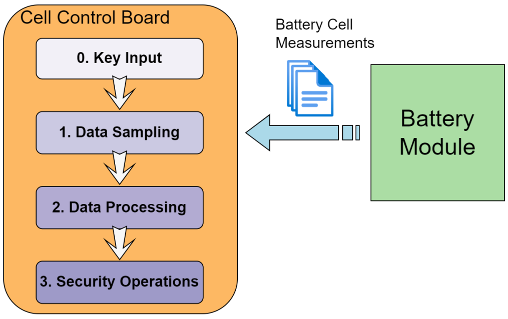


Fig. 4. BMS measurement data sampling and secure handling.

for the BMS use-case, it is important to fulfill the integrity and availability security requirements, since changes in the accuracy of the data and its sampling rate directly affects the output of the BMS control decisions. Data confidentiality also plays an important role, since the exposure of BMS data to unwanted third parties can also lead to the exposure of users' privacy, e.g., driver's behavior in electric vehicles.

Based on the design from Section III and Fig. 1, an extension in the view of a security module would be necessary as part of the CCB. This would free the design space of the battery module from the otherwise additional hardware modifications. It also means, however, that the transferred sensor data is not going to be encrypted or otherwise secured on the analogue connection between the battery module and the CCB, i.e., either through the proposed NFC interface or an adequate wired transfer. This is deemed to be acceptable, as the CCB and the battery module are usually tightly coupled and enclosed together, and attacks on those connections from the outside would be either difficult or even unfeasible. What is therefore important before the data transfer takes place between these modules, is that the authentication of the battery module was successful as described in Section IV-A. Fig. 4 shows the additional operational steps for secure data handling. The input of the key would take place at the start of the measurement session, and would be run only once for that session. Data sampling would contain the main functionality for receiving and applying monitoring and diagnostic operations from the standard BCC. Before the security operations can be applied, the data will first need to be structurally prepared, e.g., by using compression, or padding, during the data processing step. Finally, the designated security operations are run.

Placing the security operations on the CCB rather than onto the main BMS controller adds several benefits. Mainly, it presents an additional layer of security to otherwise different and uniform communication interfaces and standards used for the communication between the CCBs and the BMS controllers. It also frees the resources from the main BMS controller which would be necessary for secure storage in case of lifetime logging operations. Such data could then be stored onto the memory units connected to individual CCBs, encrypted and integrity protected against malicious modifications. This is especially important under the modulated topology where one BMS controller can communicate with multiple CCBs and would therefore reduce the computational and storage constraints on the main BMS controller. The CCB's controller needs to contain the necessary hardware and software components for the targeted security protocols.

C. Security Protocols

To protect data confidentiality, it would be necessary to employ encryption of the sampled sensor data. Embedded devices rely on the use of either Hardware Security Modules (HSM), Secure Elements (SE), Trusted Platform Modules (TPM), or processor extensions with security function implementations. Security modules under the BMS use-case should be able to provide encryption and decryption operations, and tag verification for integrity check. The security module also provides other security functions, like a Random Number Generator (RNG), secure boot, and secure key generation and storage among others. The integrated algorithms are also often hardware-implemented, meaning that they benefit from the accelerated operations and physical security considerations.

Advanced Encryption Standard (AES) is often employed for symmetric encryption operations due to its high-security profile and small footprint. AES also benefits from hardware implementations for a faster algorithm execution. During employment, AES would need to be in different modes to provide encryption operation across a larger set of data. Traditionally, CBC and CTR modes are used, with Authenticated Encryption with Associated Data (AEAD) also gaining prominence where available, with modes like EAX, GCM, or CCM.

To protect the data against modifications, i.e., to guarantee its integrity, it is recommended to apply Message Authentication Code (MAC) calculations. These can be done on an arbitrary length of sampled sensor or diagnostic data either before or after the encryption on them took place. The calculated MAC bytes would be used for the integrity check. The MAC calculation can be left out in case the affiliated encryption algorithm is from the AEAD group and hence includes an integrity tag check as part of its procedure. These functions are sufficient in providing necessary BMS sensor data protection intended for either its intermediate storage or further data propagation and processing.

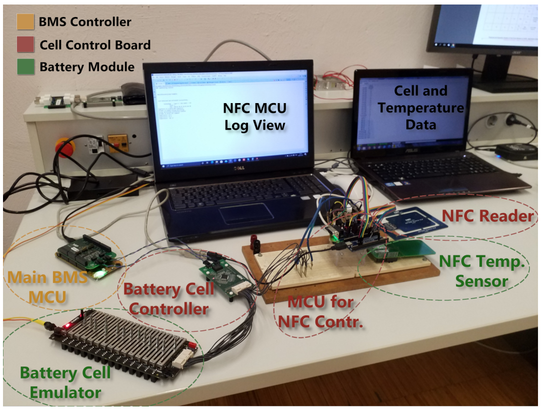


Fig. 5. Evaluation setup for the BMS NFC sensor readout.

V. Evaluation

A. Test System Implementation

To test the presented design model, we implemented a test suite that contains the necessary BMS modules, as well as the additional NFC equipment. We aimed to use the NFC modules, which support the latest *NFC Type 5 Tag* technology. Furthermore, the used components are automotive-graded where applicable for the purpose of replicating a real-world use-case as closely as possible. To that end, all devices used, except for the temperature sensor, come from the NXP Semiconductors lineup of products. The system is shown in Fig. 5.

As the main BMS controller, we use an S32K144 MCU board. It communicates with the CCB via the FRDMD-UAL33664 shield over the TPL protocol. It is further connected to an RD33771CDST that houses an MC33771C, which functions as a BCC. The CCB contains an automotive NFC Reader for handling the NFC transmissions and another S32K144 as the MCU for programming and testing. The MCU board is connected with the NFC reader via SPI. The battery module consists of a BATT-14CEMULATOR that serves as a battery emulator, an NTAG component as the passive NFC device, and a BMP180 temperature sensor. The NFC devices are of the NCF33xx product family. The antennas of the active NFC reader and the passive NTAG devices are placed in parallel to each other, with the reader placed at a short distance over the NTAG, corresponding to the positioning discussion in Section III-B. The sensor is placed in close proximity with the NTAG device. For the setup, the temperature sensor from the battery emulator was disabled from transferring the temperature data, being otherwise routed through the attached NTAG component and the added BMP180 temperature sensor that communicates with the NTAG via the I2C protocol. Hence, the BCC is able to receive the emulated cell voltage data from the battery emulator, while the temperature sensor data is sent through the NFC interface. Both the temperature and the cell voltage data are first received by the BCC and then transmitted to the BMS controller. For the authentication protocol, we base our implementation on the *originality signature* feature found on the NXP's RFID devices. Signature calculation and verification are handled via the *ecc-nano* library [34]. Elements of the project development and evaluation were handled in a recent master's thesis [35].

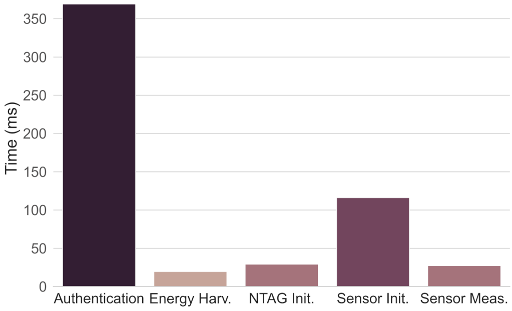


Fig. 6. Time measurement results for the Initialization phase (Authentication, Energy Harv., NTAG Init., Sensor Init.) and Monitoring phase (Sensor Meas.).

BMS status data protection: for this investigation, a security module was used to provide the necessary security operations, that comes integrated with the S32K144. The offered functionalities of this module are based on the Secure Hardware Extension (SHE) specification [36], and they included among others: a secure key derivation and storage, provided True Random Number Generator (TRNG), AES encryption algorithm with CBC mode, and Cipher-based MAC (CMAC) for data integrity and authentication.

B. NFC Sensor Readout Process Time Measurements

We divide the main BMS monitoring process into two phases: (i) *Initialization phase:* executed only once for device preparation and configuration, and (ii) *Monitoring phase:* continuous action that is called on every sample step to measure and retrieve cell sensor data. Individual steps, as well as their time measurements, are shown in Fig. 6. All represented time values are median values taken after multiple measurements.

The process starts after the NTAGs have already been discovered. As the first step the authentication protocol is run. This protocol run includes both sending an authentication request from the NFC reader, the response from the NTAG, and the verification calculation on the MCU that is connected with the NFC reader. The authentication step showed a median time of $369.30 \pm 0.37\,ms$, with majority of it being spent on the verification process. The relatively high execution time is attributed to this step being very hardware and software dependant, with optimizations being possible by using dedicated security components. With the NTAG verified, the energy harvesting check is handled which lasts for $19.64 {\pm} 0.25\,ms$. Finally, the NTAG operation initialization is run which measured $29.16 \pm 2.44\,ms$, followed with the sensor initialization that took $116.1 \pm 1.19\,ms$. After the initialization phase is finished, there is no need to reconfigure the devices during the system run. For the monitoring phase, sensor measurements are read and transmitted to the BCC using NFC communication. This phase is repeatable, with each action showing a time of $27.2 \pm 0.54\,ms$.

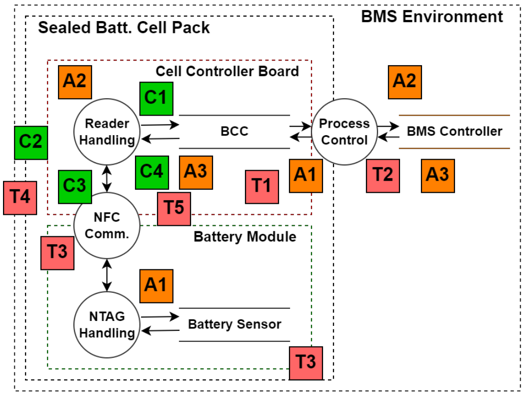


Fig. 7. Security threat analysis visual overview using Data Flow Diagram.

C. Security Threat Analysis

The proposed design has been subjected to a security threat analysis, for the purpose of evaluating the achieved security protection [37]. This has been conducted by listing individual *Assets* (A), *Threats* (T), and *Countermeasures* (C). To better illustrate the carried out process, a visual representation of the targeted use-case system model was made using Data Flow Diagram (DFD) which can be seen in Fig. 7. Here, a demonstration is made with the indicated threats, their influenced assets, and answered countermeasures, also illustrating their potential points of impact. Threats are derived based on the carried security requirements analysis, as well as the basis security threats found in common BMS models done in prior research works [18], [19], [38].

In our security model, we argue the following assumptions: (i) a battery module can only be communicated with via an adequate BCC, (ii) both the CCB and the battery module are enclosed in a chassis and the external communication can only be achieved through the BMS controller, (iii) every newly added and unknown battery module is considered untrustworthy, (iv) the CCB is deemed to contain adequate hardware and software components for security protection and calculations.

We indicate three important assets that need to be protected:

- (A1) *Sensor data*: data retrieved from the cell sensors.
- (A2) *System integrity*: hardware and software integrity.
- (A3) *Diagnostic data*: status data derived from the monitored battery readings.

An attacker would look to exploit a vulnerability of the system, i.e., the potential to conduct a successful attack. Each attack is tied to a threat and assets that are targeted by it. In the following, each separate threat is listed with a given short description, the assets that it impacts, and the countermeasures:

- **(T1)** *Battery control obstruction* $\mapsto (A1), (A3)$
  A potential threat that disturbs the cell balancing control through a fake source of sensor and diagnostic data.
  Mitigated through **(C1)** *Authentication through signature validation* by the proposed design. Here, BCCs validate every individual battery module, ensuring that the BMS controller only receives authorized status messages.
- **(T2)** *Tamper with BMS status messages* $\mapsto (A2), (A3)$
  A similar threat like (T1), but that is more covered and tries to tamper with the data rather than obstruct it.
  Also mitigated via the **(C1)** countermeasure.
- **(T3)** *Backdoor access* $\mapsto (A1), (A2)$
  An attacker might try to gain system's access through either the NFC interface or a counterfeited battery module. Protected through the **(C1)** countermeasure, but also by reducing the attack proximity by relying on the **(C3)** *NFC physical layer characteristics*.
- **(T4)** *Remote attack* $\mapsto (A1), (A2), (A3)$
  Various attacks can be launched from outside of the system on unprotected channels by using wireless communication. Under this context, we primarily consider the probing attacks that target the NFC channels.
  (C2) *Cell pack sealing* protects against remote attacks by isolating interfaces via material shielding. Also, **(C3)** would hamper the possibility of such an attack through frequency spectrum and range limitations.
- **(T5)** *BMS log data compromise* $\mapsto (A1), (A3)$
  Such an attack can take place on CCB, both from the local or possible backdoor access via exposed (T3), or through (T4). These include both the privacy leak of the associated system through the compromise of the read raw data, but also any kind of unauthorized data changes which would be intermediately stored on the CCB.
  The data can be protected by **(C5)** *Data security measures* which include the prescribed encryption, authentication and integrity validations.

TABLE III
FULL BMS DATA SAMPLING, PROCESSING AND SECURITY OPERATION

Iterations	1 sample	5 samples	100 samples
Time	$114.85 \pm 0.73\,ms$	$580.56 \pm 1.55\,ms$	$11.64 \pm 0.02\,s$

D. Data Security Overhead Analysis

An evaluation was conducted for the purpose of testing the BMS data security handling. This evaluation includes a model that was built to depict a real-world representation of the BMS data structure that includes both the monitoring and diagnostic data components. The evaluation follows the design principles described in Section IV-B and security protocol considerations in Section IV-C. To fulfill the security conditions, we employ the use of a security module as stated in Section V-A.

The BMS test system uses a battery emulator that emulates 14 cell voltages together with a sensor temperature value derived from the extended NFC measurement components. The software presents each measurement with an identifier and the measured value. These values are considered monitoring values. The BCC is further capable of deriving diagnostic values for the active status report. One-time reading from one battery module is considered a sample. In our testing case, one such sample has a length of 162 bytes. For security purposes, padding is added to round up the total size to be 176 bytes, a multiple value of 16, since the security algorithms used are of the 128-bit block length.

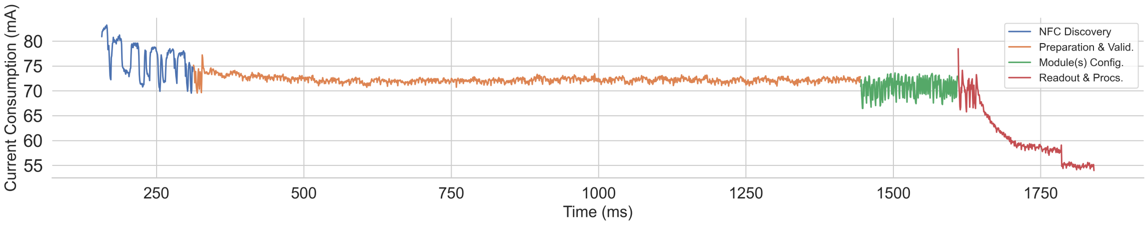


Fig. 8. Current consumption over time for the process MCU in CCB during the start-up configuration period for the active battery sensor readout showcasing: discovery process from the NFC reader, internal config. & secure NTAG validation, batt. module, sensor and NTAG configuration, and NFC readout process.

The evaluation was divided into three phases following the design given in Fig. 4. Values are shown as mean values derived from multiple measurements. The initial key insertion step was measured at a constant $20\,ms$. ① *Data sampling* was measured at $112.98 \pm 0.54\,ms$. Among this, the measurement step only required $3.5\,ms$, with the remaining $109.5\,ms$ being used for the diagnostic derivations. ② *Data processing* was shown to have little impact and be very fast with a resulting time of $1.0 \pm 0.1\,ms$. ③ *Security operations* include the AES-CBC and CMAC calculations for the data confidentiality, integrity and authenticity security coverage. The execution was relatively fast, resulting in a total time of $992 \pm 8.75\,\mu s$.

As it can be concluded from the evaluation, the main operational overhead comes from the data sampling step. This shows that the integration of security operations along with the traditional BMS data sampling results in minimal overhead change, having an increase of $1.7\,\%$, and therefore would not intervene with the standard BMS time-critical safety operations. Even if the security data logging procedure would only be limited to the measurement steps, either due to performance of infrequent diagnostic checks, it still would result in an acceptable overhead range, adding additional $\approx 1.5\,ms$ to the $3.5\,ms$ of measurements sampling. The main challenges would come from determining how often the security logging should take place, and defining what would be the necessary memory capacity for the long-term administration. The measurements were also done over a longer operation run, but no significant changes between the measurements have been detected. The results follow a linear time increase. Total times for 1, 5 and 100 sample runs are shown in Table III.

E. Energy Consumption

We measured the energy consumption of our BMS implementation to investigate how much additional energy would be required with the added CCB components, energy overhead for the added sensors and the NTAG, and the overall energy consumption for the BMS controller when considering the added security operations. For conducting the measurements, we used a Nordic Semiconductor Power Profiler Kit.

The **CCB** was evaluated on two added elements: on the extended process MCU, and the active NFC reader. Fig. 8 shows the current consumption for the CCB's MCU, which is responsible for the control of the NFC reader. The same operational segments were also considered in parallel when measuring the consumption from the CCB's NFC reader board.

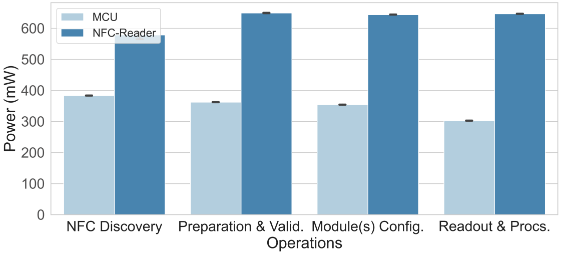


Fig. 9. Power consumption over the CCB's MCU and NFC reader.

From Fig. 8, we can observe four operational segments:

1) *NFC Discovery*: mainly considers the discovery loop for the battery module's passive NTAG component; shows the highest peak in current consumption, but the average remains consistent with other operations.
2) *Preparation & Validation*: board configuration and start-up steps; also includes the signature authentication step (Section IV-A). Shows a constant and stable consumption over most of its period.
3) *Module(s) Configuration*: configuration command exchange for NFC and sensor devices on the battery module. More oscillating consumption due to a more intensive NFC reader interaction.
4) *Readout & Processing*: one iteration of the battery sensor readout and data handling. The drop in current consumption indicates inactivity on the MCU part after the operation ends, wherein the beginning a higher consumption can be observed from the NFC data exchange.

Fig. 9 shows the graphical comparison for average power consumption between the CCB's MCU and NFC reader after five different measurement runs. The operational voltage for both components was set at $5\,V$. Overall, the added devices resulted in an increase of up to $1\,W$ of power consumption, without optimization considerations. This means that for the repeatable monitoring phase (described in Section V-B), without any other additional computational overhead, the energy consumption amounts to $25.82\,mJ$.

The power consumption shown considers the consumption of the whole NFC reader board during the active period. This means that it accounts for all regulators, communication interfaces, the NFC chip controller, and most importantly, the active RF transceiver. Since for most of the operational run, the communication interaction between the active and passive NFC devices were taking place, the RF field remained also mainly active. In a general environment, NFC readers are indented to offer a polling feature, i.e., periodical wake-up from the stand-by state for the purpose of detecting present passive NFC devices. This feature greatly reduces the average current consumption over time. However, in the presented design, the communication remains active for most of the time during the monitoring phase since the positioning condition of the devices would not change and, depending on the sampling rate, the next measurement might occur soon after the last one finished. Optimisation and adjustment of the standby mode and RF activation is largely dependent on the targeted system implementation goals and is left open for the developers.

Battery pack sensor consumption was negligible compared to other energy consumption of the system, with current consumption of $40\,nA$ for the standby state, and peeking for a short time of up to $22\,uA$ during initialization and active period. The NTAG relies on the energy harvesting feature for the operation and control of the sensors (see Section III-B). In our test case, this results in additional current consumption draw of the NFC reader, with an average rise of $5\,mA$, when in the range of the NTAG.

We analyzed the **BMS controller** on the total power and energy consumption for one full diagnostic sampling cycle. The average drawn power resulted in $122.16\,mW$, with an average energy consumption of $13.80\,mJ$, after ten different system runs. Additionally, the security operations, and its preceding data processing, resulted only in a slight increase of energy consumption with $0.28\,mJ$ for one sampling cycle, i.e., $2.66\,mJ$ for one-time key-insertion operation. We can observe that the added security operations result only in a minimal increase of up to $2\,\%$ of energy consumption per sample run.

F. Battery Sensor Throughput Analysis

The throughput of battery module sensor data largely depends on several factors. Primarily, it is dependent on: (i) the number of the total sensors used per module, (ii) the number of total battery modules used per CCB, the number of CCBs used per the main central BMS controller, and in this case of using the NFC components, (iv) the total number of communicating NFC components (active and passive) and the number of sensors per passive components (communication chains). As indicated in Section V-A, for the experimental setup a battery emulator was used that offers the reading of fourteen battery cells and one temperature sensor. As such, under our setup, we represent a system that has: one sensor per battery module, one passive NTAG device per battery module, connected directly with the battery sensor, and a CCB with one active NFC reader per assigned battery module. More points on the realization and potential future work based on the aforementioned throughput factors are discussed in Section VI.

The readout of the NTAG is done through the provided SRAM. The SRAM in our test environment offers 256 bytes of data transfer, with data being divided into blocks of 4 bytes. In our setup, reading the whole SRAM would take $82.28\,ms$. However, in a real setting, this readout would probably require much less data. As indicated, each sensor would need 1-2 blocks containing 4 bytes each to process and send its derived data. The amount of sensors is also usually limited per pack, and it is very unlikely, that with current battery modules the data requirements would exceed one SRAM read request.

TABLE IV
EXPECTED SAMPLING THROUGHPUT PER ONE CCB AND BATT. MODULE

Iterations	100	1,000	10,000
Time & **Data Size**	$2.96\,s$ $0.76\,kB$	$29.24\,s$ $7.54\,kB$	$297.45\,s$ $76.68\,kB$

Each measurement requires three actions to take place:

1) CCB (NFC Read.) → (NTAG) Batt. module; write command to enable and start the sensor measurement.
2) CCB (NFC Read.) → (NTAG) Batt. module; read command to read out from the specific block of the SRAM of the saved sensor values.
3) CCB (NFC Read.) ← (NTAG) Batt. module; transmitting the sensor values from the NTAG's SRAM.

The request and response frames contain additional data in the form of flags, IDs, commands, address, and the Cycle-Redundancy-Check (CRC) appendices. Thus, the first two write and read SRAM commands take additional 15 bytes of the header, which can be reduced to 7 bytes if the ID component is removed (if the communication is 1-to-1, it is not necessary). The response SRAM read frame only has 3 header bytes (flags and CRC). The remaining payload depends on the sensor data, which in our case is two blocks, i.e., 8 bytes. Out of those, the measured value is contained in 2 bytes.

Based on the experiments from the implementation setup, the CCB was able to conduct 33 measurements per second, i.e, for reading 8 bytes, 264 bytes/s of the pure measurement data. As noted, each measurement is a three-stage process with always the same repeating overhead. The time required for processing the received data by the CCB is negligible compared to the transfer time. This accounted for $1-2\,ms$ between each read request used for handling the processing of the received sensor data, but measurements otherwise correspond to running a single sensor measurement during the "Monitoring phase" as indicated in Section V-B. The total amount of time and handled sensor payload data when running repeatable measurements for a different number of iterations is shown in Table IV. As expected, linear growth of time compared to the number of repeated iterations is observed.

Compared to the measurements conducted for the overall BMS process after applying security operations in Section V-D, it can be concluded that the amount of data processed would suffice for the current setup, even when using multiple sensors per battery module. Furthermore, for a modulated topology that is presented here, the measurements and sampling would be conducted in parallel, and are independent of the number of used CCB, being only limited by the processing power of the assigned BMS controller. However, for BMS that have requirements for a faster sampling rate, in this case, that being $< 30\,ms$, additional optimization aspects would need to be considered.

VI. Discussion and Future Work

As we see from Section V-B, the initialization phase is very time-demanding. The primary reason for this is the long execution time for the signature validation, which took around 69% of the total initialization phase time in case all four initialization phase steps would be executed sequentially. Based on our evaluation tests and findings, we note the following important points that can be handled during the implementation of the proposed design to alleviate the time and help in phase delivery: (a) signature validation hardware and software need to be optimized for the target system to reduce the overall initial execution time, (b) process parallelization for steps reduction in the execution, (c) configuration and status caching for the targeted devices.

The proposed system design solution can also be used on different BMS settings regardless of the use case, which should fit the needs of automobile and industrial environments. In this context, the applicability of the presented solution is not only limited to conventional temperature sensors but also other sensors as part of a battery module. The battery cell's sensor placement and the target of measurement play an important role and could benefit from using the means of NFC transfer. The closer the sensor is to the core of the battery, the more accurate and time-punctual results are going to be. To this end, it would be possible to utilize NFC to transfer the data from the inside to the outside of the battery from these sensors. These measurements would add an additional layer to the safety precautions of the BMS, and hence, would influence the increased safety of the overall system. Separate research would need to be conducted which would investigate the optimal placement and usability of using the NFC communication for the data transfer in regards to the actual sensor placement in a battery module.

Next to the handling of the sensor, antenna positioning should be further investigated as well [16]. For the current setup, a parallel placement is used with no physical considerations. Future work should also include research on the limits of the NFC range when considering the obstructing environment of the enclosed BMS modules. Additionally, an analysis should be made on the possible range and performance when not considering the energy harvesting feature. As mentioned in this work under Section III-B, the proposed design uses the energy harvesting feature of the NFC technology to allow for less reliance on the wired connections with the batteries. However, by disabling this feature and basing the use-case on using the source of power from the underline batteries, the range can be greatly exceeded, but at a higher cost, as additional wiring would also need to be provided.

Concerning the system design, another important point to consider for future work is the analysis of the number of used NFC elements. The current design proposes the use of one active NFC device per CCB and one passive device for the battery module. Considerations should be made on the adequate distribution of active and passive NFC devices. This is especially important for the passive NFC devices, i.e., the NTAGs, since an adequate hardware solution should be provided that considers the potential of multiple sensors placed in one battery module, or multiple NTAGs being handled by one active NFC component. Optimisations in the system design could lead to a reduction in the overall production cost.

In this section, we have primarily discussed the hardware aspects of future work and improvement, but an investigation should also be made into the optimization methods for the purpose of improving the time execution and reliability of the connection during the NFC sensor readout process. This can be realized on different software layers, targeting both the lower driver control and application stack. Among others, this investigation may include the consideration of different communication protocol extensions, but also the improved security realization. For future work, we also plan to further extend the investigation of the data security control within the BMS environment. Security attention should be given to the extension of the authentication algorithm, but also in adding an extra security layer for communication with the external components and services. Additional threat aspects need to be considered when the attack surface is extended [33], [38].

VII. Conclusion

In this work, we have presented the idea of using NFC as a wireless communication interface for battery sensor readouts in BMS. A system design has been proposed that considers the construction of a modulated BMS with NFC components with special regard to the data exchange protocol and NFC requirements. To alleviate the risk of the counterfeited battery cells and prevent safety and security threats that can arise from them, an authentication model has been proposed and evaluated. A further study has been conducted that investigates the security handling and control of the derived sensor and diagnostic data once they are logged on a cell control board. Experimental results using real components show the feasibility of our approach, but also design challenges that open the possibilities of various further research in this field.

Acknowledgments

We would like to express our sincere thanks to the BMS team from the NXP Semiconductors Austria GmbH Co & KG for the support and cooperation during the design and evaluation phases of this work, as well as for providing the necessary equipment for conducting our experimental tests.

This project has received funding from the "EFREtop: Securely Applied Machine Learning - Battery Management Systems" (Acronym "SEAMAL BMS", FFG Nr. 880564).

References

[1] X. Hu, F. Feng, K. Liu, L. Zhang, J. Xie, and B. Liu, "State estimation for advanced battery management: Key challenges and future trends," *Renewable and Sustainable Energy Reviews*, vol. 114, 2019.

[2] R. Xiong, J. Cao, Q. Yu, H. He, and F. Sun, "Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles," *IEEE Access*, vol. 6, pp. 1832–1843, 2018.

[3] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles," *IEEE Industrial Electronics Magazine*, vol. 7, 2013.

[4] D. Andrea, *Battery Management Systems for Large Lithium-ion Battery Packs*. EBL-Schweitzer, Artech House, 2010.

[5] P. Sun, R. Bisschop, H. Niu, and X. Huang, "A Review of Battery Fires in Electric Vehicles," *Fire Technology*, pp. 1–50, 01 2020.

[6] G. De Maso-Gentile, A. Bacà, L. Ambrosini, S. Orcioni, and M. Conti, “Design of CAN to Bluetooth gateway for a Battery Management System,” in *2015 12th International Workshop on Intelligent Solutions in Embedded Systems (WISES)*, pp. 171–175, 2015.

[7] C. Shell, J. Henderson, H. Verra, and J. Dyer, “Implementation of a Wireless Battery Management System (WBMS),” in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pp. 1954–1959, 2015.

[8] A. Rahman, M. Rahman, and M. Rashid, “Wireless Battery Management System of Electric Transport,” *IOP Conference Series: Materials Science and Engineering*, vol. 260, p. 012029, nov 2017.

[9] A. Samanta and S. S. Williamson, “A Survey of Wireless Battery Management System: Topology, Emerging Trends, and Challenges,” *Electronics*, vol. 10, no. 18, 2021.

[10] M. Schneider, S. Ilgin, N. Jegenhorst, R. Kube, S. Püttjer, K.-R. Riemschneider, and J. Vollmer, “Automotive Battery Monitoring by Wireless Cell Sensors,” in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 816–820, 2012.

[11] D. Alonso, O. Opalko, and K. Dostert, “Physical Layer Performance Analysis of a Wireless Data Transmission Approach for Automotive Lithium-Ion Batteries,” in *IEEE VNC*, pp. 235–242, 2015.

[12] M. Lee, J. Lee, I. Lee, J. Lee, and A. Chon, “Wireless Battery Management System,” in *World Electric Vehicle Symposium and Exhibition (EVS27)*, pp. 1–5, 2013.

[13] J. Farmer, J. Chang, J. Zumstein, J. Kotovsky, E. Zhang, A. Dobley, G. Moore, F. Puglia, S. Osswald, K. Wolf, J. Kaschmitter, S. Eaves, and T. Bandhauer, “Wireless Battery Management System for Safe High-Capacity Li-Ion Energy Storage,” tech. rep., Lawrence Livermore National Laboratory, 01 2013.

[14] T. Vogt, “Wired vs. Wireless Communications in EV Battery Management,” tech. rep., Texas Instruments, 10 2020.

[15] T. Gherman, M. Ricco, J. Meng, R. Teodorescu, and D. Petreus, “Smart Integrated Charger with Wireless BMS for EVs,” in *IECON - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018.

[16] D. Alonso, O. Opalko, M. Sigle, and K. Dostert, “Towards a wireless battery management system: Evaluation of antennas and radio channel measurements inside a battery emulator,” in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pp. 1–5, 2014.

[17] K. Lounis and M. Zulkernine, “Attacks and Defenses in Short-Range Wireless Technologies for IoT,” *IEEE Access*, vol. 8, 2020.

[18] S. Sripad, S. Kulandaivel, V. Pande, V. Sekar, and V. Viswanathan, “Vulnerabilities of Electric Vehicle Battery Packs to Cyberattacks,” *ArXiv*, 2017.

[19] S. Kumbhar, T. Faika, D. Makwana, T. Kim, and Y. Lee, “Cybersecurity for Battery Management Systems in Cyber-Physical Environments,” *ITEC 2018*, pp. 761–766, 2018.

[20] A. Khalid, A. Sundararajan, A. Hernandez, and A. I. Sarwat, “FACTS Approach to Address Cybersecurity Issues in Electric Vehicle Battery Systems,” in *2019 IEEE TEMSCON*, pp. 1–6, 2019.

[21] S. Engels, “Counterfeiting and piracy: the industry perspective,” *Journal of Intellectual Property Law & Practice*, vol. 5, 05 2010.

[22] F. Basic, M. Gaertner, and C. Steger, “Towards Trustworthy NFC-based Sensor Readout for Battery Packs in Battery Management Systems,” in *2021 IEEE International Conference on RFID Technology and Applications (RFID-TA)*, pp. 285–288, 2021.

[23] M. Spörk, C. A. Boano, and K. Römer, “Performance and Trade-Offs of the New PHY Modes of BLE 5,” in *Proceedings of the ACM MobiHoc Workshop on Pervasive Systems in the IoT Era*, PERSIST-IoT ’19, (New York, NY, USA), p. 7–12, Association for Computing Machinery, 2019.

[24] X. Huang, A. B. Acharya, J. Meng, X. Sui, D.-I. Stroe, and R. Teodorescu, “Wireless Smart Battery Management System for Electric Vehicles,” in *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 5620–5625, 2020.

[25] T. Kim, D. Makwana, A. Adhikaree, J. S. Vagdoda, and Y. Lee, “Cloud-Based Battery Condition Monitoring and Fault Diagnosis Platform for Large-Scale Lithium-Ion Battery Energy Storage Systems,” *Energies*, vol. 11, no. 1, 2018.

[26] W. Li, M. Rentemeister, J. Badeda, D. Jöst, D. Schulte, and D. U. Sauer, “Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation,” *Journal of Energy Storage*, vol. 30, p. 101557, 2020.

[27] A. B. Lopez, K. Vatanparvar, A. P. Deb Nath, S. Yang, S. Bhunia, and M. A. Al Faruque, “A Security Perspective on Battery Systems of the Internet of Things,” *Journal of Hardware and Systems Security*, 2017.

[28] ISO/IEC 18092:2013, “Information Technology—Telecommunications and Information Exchange Between Systems—Near Field Communication—Interface and Protocol (NFCIP-1),” Standard, International Organization for Standardization, 2013.

[29] ISO/IEC 21481:2021, “Information technology — Telecommunications and information exchange between systems — Near field communication interface and protocol 2 (NFCIP-2),” Standard, International Organization for Standardization, 2021.

[30] T. Ulz, T. Pieber, C. Steger, S. Haas, and R. Matischek, “Sneakernet on Wheels: Trustworthy NFC-based Robot to Machine Communication,” in *2017 IEEE International Conference on RFID Technology Application (RFID-TA)*, pp. 260–265, 2017.

[31] C.-L. Chen, Y.-Y. Chen, T.-F. Shih, and T.-M. Kuo, “An RFID Authentication and Anti-counterfeit Transaction Protocol,” in *2012 International Symposium on Computer, Consumer and Control*, pp. 419–422, 2012.

[32] B. A. Alzahrani, K. Mahmood, and S. Kumari, “Lightweight Authentication Protocol for NFC Based Anti-Counterfeiting System in IoT Infrastructure,” *IEEE Access*, vol. 8, pp. 76357–76367, 2020.

[33] N. Druml, A. Genser, A. Krieg, M. Menghin, and A. Höller, *Solutions for Cyber-Physical Systems Ubiquity*. IGI Global, 08 2017.

[34] “ECC-Nano.” https://github.com/iSECPartners/nano-ecc, 2013. Accessed: 15.01.2022.

[35] M. Gaertner, “Design and Implementation of a NFC-based Solution for Secure Battery Management Systems,” Master’s thesis, Graz University of Technology, 2021.

[36] AUTOSAR, *Specification of Secure Hardware Extensions*, 2019.

[37] S. Myagmar, A. Lee J., and W. Yurcik, “Threat Modeling as a Basis for Security Requirements,” *in SREIS*, 2005.

[38] M. Cheah and R. Stoker, “Cybersecurity of Battery Management Systems,” *HM TR series*, vol. 10, no. 3, p. 8, 2019.

Fikret Basic received the Dipl.-Ing. (M.Sc.) degree in computer science from the Graz University of Technology in 2019, with the main focus of his studies being on the pervasive computing, information systems, secure system design and HW/SW codesign. From 2019 until 2020 he worked as a design engineer in CISC Semiconductors. From 2020 he is employed at the Institute of Technical Informatics of Graz University of Technology, where he is currently also pursuing the Ph.D. degree in information and computer engineering. His current area of research focuses on the security in battery management systems.

Martin Gaertner received the Dipl.-Ing. (M.Sc.) degree in information and computer engineering from the Graz University of Technology in 2021, focussing on secure and correct systems and embedded automotive systems. His research focuses on secure systems in embedded automotive environments.

Christian Steger received the Dipl.-Ing. (M.Sc.) degree in 1990, and the Dr.Techn. (Ph.D.) degree in electrical engineering from the Graz University of Technology, Austria, in 1995. He graduated from the Export, International Management and Marketing course with the Karl-Franzens-University of Graz in 1993 and completed the Entrepreneurship Development Program at MIT Sloan School of Management, Boston, in 2010. He is a Strategy Board Member of the Virtual Vehicle Competence Center (ViF, COMET K2) in Graz, Austria. Since 1992, he has been an Assistant Professor with the Institute of Technical Informatics, Graz University of Technology, where he heads the HW/SW codesign group with the Institute of Technical Informatics.

A.4 [D] A Novel Secure NFC-based Approach for BMS Monitoring and Diagnostic Readout

F. Basic, C. R. Laube, C. Steger and R. Kofler, "A Novel Secure NFC-based Approach for BMS Monitoring and Diagnostic Readout," in *2022 IEEE International Conference on RFID (RFID)*, pp. 23-28, IEEE, 2022.

Abstract. In modern systems that rely on the use of Battery Management Systems (BMS), longevity and the re-use of battery packs have always been important topics of discussion. These battery packs would be stored inside warehouses where they would need to be properly monitored and configured before their re-integration into the new systems. Traditional use of wired connections can be very cumbersome, and sometimes even impossible, due to the outer layers and packaging. To circumvent these issues, we propose an extension to the conventional BMS design that incorporates the use of Near Field Communication (NFC) for the purpose of wireless battery pack status readout. Additionally, to ensure that these packs are only managed by authenticated devices and that the data that is communicated with is protected against outside eavesdropping and tampering, we present a solution in the form of a lightweight security layer on top of the NFC protocol. To show the feasibility of our design, an accompanying prototype has been implemented and evaluated.

My Contribution. As the main author of the publication, I contributed by structuring and writing the majority of the paper. I provided the main basis design for the BMS and NFC external diagnostic readout, especially analyzing and specifying requirements for the security model and protocol based on symmetric cryptography. Claudia Laube provided the extensions to the design, worked on the prototype and implementation, and conducted the performance evaluation. I supported the evaluation concerning the performance measurements and provided the security evaluation. Christian Steger provided mentoring guidance, support by the design specification and relevant research questions. The use case specification of the current industrial SotA on the BMS and NFC requirements was provided by Robert Kofler and his team from NXP Semiconductors. We also relied on the use of their devices for implementation and testing, as was the case for the majority of other publications as well.

A Novel Secure NFC-based Approach for BMS Monitoring and Diagnostic Readout

Fikret Basic, Claudia Rosina Laube, Christian Steger
Institute of Technical Informatics
Graz University of Technology
Graz, Austria
{basic, laube, steger}@tugraz.at

Robert Kofler
R&D Battery Management Systems
NXP Semiconductors Austria GmbH Co & KG
Gratkorn, Austria
robert.kofler@nxp.com

***Abstract*—In modern systems that rely on the use of Battery Management Systems (BMS), longevity and the re-use of battery packs have always been important topics of discussion. These battery packs would be stored inside warehouses where they would need to be properly monitored and configured before their re-integration into the new systems. Traditional use of wired connections can be very cumbersome, and sometimes even impossible, due to the outer layers and packaging. To circumvent these issues, we propose an extension to the conventional BMS design that incorporates the use of Near Field Communication (NFC) for the purpose of wireless battery pack status readout. Additionally, to ensure that these packs are only managed by authenticated devices and that the data that is communicated with is protected against outside eavesdropping and tampering, we present a solution in the form of a lightweight security layer on top of the NFC protocol. To show the feasibility of our design, an accompanying prototype has been implemented and evaluated.**

***Index Terms*—Battery Management System; Security; Cyber-physical; Authentication; Near field Communication; Mobile.**

I. Introduction

With the rise of general awareness for green sustainability and environmental protection, Electric Vehicles (EV) are becoming ever more prevalent. The most valued components that they contain are the battery packs. These packs lose their power over time, with many manufacturers suggesting that the battery cell packs should be replaced when the battery capacity drops to around 70% - 80% of their maximum capacity [1]. While it varies, these values are expected to be reached after just ten years of active usage. To reduce the load on the living environment, reusable battery packs are almost certainly going to become important in the upcoming market, as they can be recycled for other purposes, such as for energy harvesting, or for systems with moderated safety requirements [2].

A battery pack usually contains a set of battery cells and sensors connected to a Battery Cell Controller (BCC). The safety control and charging handling of battery packs are further managed through Battery Management Systems (BMS) [3]. These are specialized devices that handle the main data processing and system control from one or several battery packs, connected in a central, modular or distributed topology [4]. BMS components are traditionally coupled in an enclosed environment, and hence, work as a closed system. Therefore, when a battery pack is withdrawn to a storehouse, an external communication interface would need to be provided for the purpose of obtaining diagnostic information. The usability of this external readout is generally seen under two potential use-cases: (i) warehouse stored battery cells with their respective BCC, and (ii) active usage in systems (e.g. EVs) for faults and communication breakouts analysis. In both cases, it is of importance that the abnormal behaviour of battery cells is detected early by the BMS. Changes in temperature and storage conditions can affect the life of a battery cell [5], [6]. Outside of the battery status readout, external communication can also be used for firmware and configuration updates [7].

Extending the functionality also extends the portfolio of potential malicious attacks. A capable attacker could fake a single temperature value to initiate a fake thermal runaway in the BMS. Further manipulations could even allow the attackers to completely mask the real damage that is done to the battery pack or leave an exploit that could be used to hide a fake malicious battery pack by replicating the behaviour of a real one. It is therefore important that the communicating devices are mutually authenticated and their data adequately protected.

The readout of battery packs can be achieved by using a conventional wired interface, e.g., Controller Area Network (CAN), or other serial interfaces. However, these come with several limitations as the device handling would need to be done on an individual basis. Wireless technologies allow for more efficient handling of a larger number of battery packs. It also circumvents the limitations of packaged battery packs and allows for an external readout, with its integration into an automated environment also allowing for faster processing by employing contact-less readers and assembly lines. But even with these advantages, choosing an appropriate wireless technology under the presented conditions is difficult, as we see several requirements that need to be fulfilled:

- *Widespread availability*: the protocol needs to be supported across multiple devices and be simple to integrate.
- *"Wake-up" functionality*: to reduce the reliance on the use of the battery cells and additional connection points, it is desirable that the control units function independently and are powered-up from an external interface.
- *Security considerations*: the system needs to be secure against common threats and to support an integration of the extended security communication.

To fulfil the mentioned criteria, we have decided to use Near Field Communication (NFC) as the proposed wireless technology. NFC allows for easy integration into the existing BMS architectures, offers a wide range of supported NFC readers (incl. mobile phones), and has a fast readout process. To reduce the reliance on the battery cells, NFC also offers the energy harvesting feature, being able to power up an NFC-tag device from an outside NFC reader. While the NFC protocol itself does not offer a full security suite, it does offer some security features that are of an advantage when compared to other wireless technologies [8]. Furthermore, the communication usually has smaller latency and less interference when compared to other wireless technologies used with BMS [9]. We extend on the notion of the protocol security designs, by proposing a low-overhead security solution that can be used under the specified industrial application settings.

Contributions. Summarized, the main contributions of this paper include: (i) a design proposal for establishing external NFC readout between a configuration reader and a BMS and its battery packs, (ii) a lightweight security solution built on top of the NFC layer that is able to provide mutual authentication and secure session establishment, (iii) testing and evaluation of the presented methods on a real hardware test-suite. To the authors' best knowledge, this is the first publication that describes an NFC-design proposal with an integrated security protocol for a BMS status readout.

II. Background and Related Work

A. Wireless Battery Management System (BMS)

Recently, wireless BMS have become a topic of discussion since replacing wired with wireless interfaces would help in reducing production cost and complexity. Many of the recent works look for solutions using the 2.4 GHz frequency band technologies such as Bluetooth [10], [11], ZigBee [12], and WiFi [13]. However, most of these publications primarily focus on the inter-communication between modular BMS components, and only partially on the requirements derived for external access, which we investigate in this work.

Combining NFC applications with BMS is a relatively novel topic, as not much work has yet been done by the research community as mentioned in a current survey study of wireless BMS [14]. A recent paper published by Basic et al. [15] proposes a solution for wireless sensor readouts from battery cells to BCCs by using NFC technology, and also presents an anti-counterfeiting authentication measure, but only for closed active systems. In this work, we further try to bridge the gap of some of the open questions in respect to design requirements between NFC and BMS by also extending the security application for external communication interactions.

B. Near Field Communication (NFC) Security

NFC is a high-frequency Radio Frequency Identification (RFID) wireless technology operating in the 13.56MHz frequency band with a range up to 10 cm. NFC-based tags and smart cards are typically compliant with ISO/IEC 14443 or ISO/IEC 15693. The passive tags are capable to be powered by the active readers for the duration of the data exchange. NDEF record is a widely accepted approach for data encapsulation in NFC, as it provides a relatively low message overhead. NFC relies on different security approaches to provide additional protection for data handling. A common approach would be to use Signature Record Type Definition (RTD). The original 1.0 version was proven to be vulnerable to attacks [16], [17], with the 2.0 version being the one that is often deployed instead. It uses signatures through certificate chaining to provide data authenticity and integrity. However, it does not provide data confidentiality. Additionally, the employed schema relies on asymmetric cryptography, which can prove to be demanding on constrained devices, requiring a dedicated infrastructure.

Extended solutions, like the QSNFC proposed by Ulz et al. [18], provide a full security suite. QSNFC uses Diffie-Hellman key exchange and certificates for device authentication. However, this approach would not be suitable for the presented BMS use-case, as only the QSNFC's server authenticity is checked, but not the client's, leaving the possibility for configuration updates from unauthorized readers towards the BMS MCU. It also relies on 128-bit public keys, which is less than the current NIST recommendation for legacy applications. Regarding other certificate-based approaches, Urien and Piramuthu [19] propose a TLS schema adaptation for NFC. It, however, would be very resource-demanding for the current BMS applications and therefore not applicable.

III. Design of a Novel Secure BMS NFC Readout

For the configuration and status readout of BMS, a system architecture is proposed containing the following components:

1) *Processing unit*: e.g. an MCU for process handling, attached either through BCC or the main BMS controller.
2) *NFC-Tag (NTAG)*: for communication and data transfer.
3) *Secure Module (SM)*: provides security functionality.
4) *Mobile reader*: a mobile device or a different NFC reader-equipped device that is also capable of the necessary processing and security operations.

The mobile reader needs to be appropriately configured to be able to communicate with the dedicated NTAGs and BMS hardware. For the context of this work, an assumption is made which entails that the devices have been correctly pre-configured and embedded with the correct security material. NTAG, which is used to transmit information between the BMS processing unit and the external mobile reader, is primarily used as a bridge device to pass and handle the data. This is done with the intention that the security functions and the secure data would be stored inside a trusted environment, which in this case would be an SM that resides on the battery pack together with the MCU. The NTAG can also be boosted with additional device authorization mechanisms [15].

As mentioned in Section I, we have focused on deriving a design solution for two specific use-case scenarios:

- *Active scenario*: active usage within a BMS system; capable of extracting current operational diagnostic data.
- *On-Rest scenario*: for stored and inactive battery packs; capable of extracting lifetime and present status data.

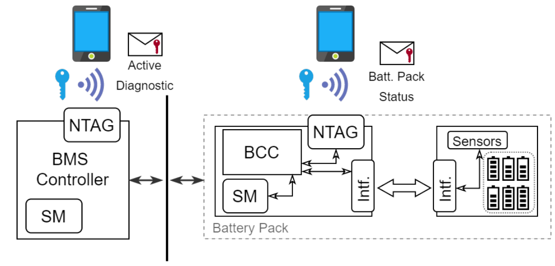


Fig. 1. Proposed system architecture: The Active scenario is shown on the left hand side with the communication going through the main BMS controller; On the right hand side the On-Rest scenario is shown for a passive readout.

The communication design in both cases does not change, as the security protocol stays the same. The main difference comes from the use of the NTAG component related to energy harvesting and the wake-up procedure. Namely, the wake-up procedure needs to be initiated for a stored battery pack to conserve the used energy. Here, the NTAG plays a part of the event trigger and energy supplier for the initial wake up of the connected MCU. Based on the power draw, the MCU can either be powered directly from the NTAG, or it needs to power itself up by re-diverting the energy from the connected battery cells. As mentioned, in both cases, the actual event action does not change, and it results in a read-out of the predefined information. The system architecture and its applicable use-cases are shown in Fig. 1. The line indicates separation since for the on-rest warehouse scenario, battery packs are usually detached from their main BMS controllers. For the rest of this work, we will refer to the processing unit as the BMS for both use-cases.

A. Security Threats and Prerequisites

The communication design needs to adhere to security requirements drawn from research concerning common threats in BMS [20], and otherwise similar industrial systems [21], as well as Confidentiality, Integrity, and Availability (CIA) principles from a general security design. Specifically, the design needs to be able to also protect transferable sensitive BMS data from being spied on or tampered with. Other attacks can include a variety of replay or Man-in-the-Middle (MitM) attacks, denial of service, and malicious actions against the hardware and software integrity of a BMS device [22]–[24]. NFC provides a low range communication, that limits the range from where attacks can be conducted. However, there still exist a variety of possible remote attacks that could take the advantage of an unprotected channel [21], [25]. These can range from sniffing attacks that can compromise the confidentiality of transferred data using eavesdropping equipment with a range of up to 10m as demonstrated by Haselsteiner and Breitfuß [8], up to attacks that directly target the authentication identity [26]. With these threats in mind, authentication will also need to be provided via a mutual authentication procedure that takes place before the data exchange starts.

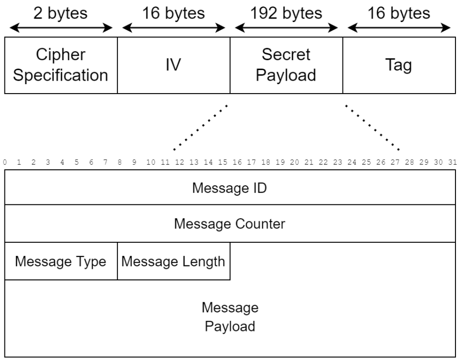


Fig. 2. SNDEF record structure.

An additional aim of the security design is to keep the overall structure lightweight, both in its implementation complexity, processing time, and extra data size. To achieve these properties, we opted to use a symmetric cryptography architecture approach. Security needs to be guaranteed based on Kerckhoffs's principle, i.e., the master key material needs to be unique and securely stored on the devices.

B. Secure Near Field Communication Structure

To enable the secure message exchange, a message structure has been proposed in the form of an NDEF record named Secure-NDEF (SNDEF). These records are intended to be short NDEF records, build as an extension to the proposed records from Ulz et al. [7], but adapted to be more flexible in use among different cipher protocols, such as the Authenticated Encryption with Associated Data (AEAD) schemes or the traditional AES+MAC protocols. The record structure can be seen in Fig. 2. It consists of: (i) a cipher specification (e.g., AES-CBC+CMAC, AES-GCM, AES-CCM), (ii) an Initialization Vector (IV), (iii) a secret payload, which is the encrypted data, and (iv) a tag, a piece of additional information for integrity check, e.g., a Message Authentication Code (MAC).

The computations are done in the Encrypt-then-MAC approach, meaning that the data is first encrypted and then the tag is calculated on the data, i.e., including the secret payload and the IV. The secret payload contains a 4-bytes message ID, which for application purposes can also be adapted to be a, e.g., sensor ID. Message type holds the purpose of the action, such as READ_STATUS or UPDATE_CONFIG. The structure uses the message counter field to keep track of a larger chain of messages. It needs to be unique for each message in a communication session (for each key) as a guard against replay attacks from rogue messages. The current design allows up message length of up to 182 bytes, being sufficient for the application's needs, with possible extensions.

C. Security Measures

Based on the security and system analysis from Section III-A, a security architecture is presented consisting of the following security protocols and operations, as seen in Fig. 3.

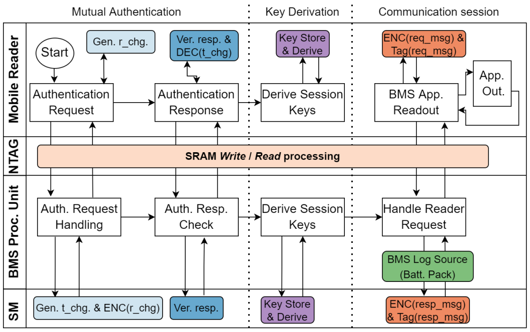


Fig. 3. Diagram showcasing sequence steps between the participating devices.

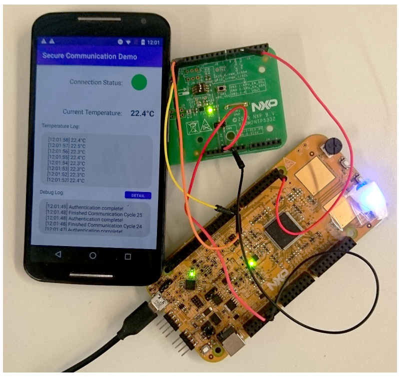


Fig. 4. Prototype showcasing the secure NFC BMS readout.

Mutual authentication. Before the communication starts, both the mobile reader and the BMS module need to authenticate each other, i.e., mutually prove that they come from valid sources. The architecture uses a symmetric challenge/response mechanism with pre-embedded keys. The protocol starts with the mobile reader sending a 128-bit randomly generated challenge, to which the BMS replies with its 128-bit challenge and encryption of the reader's challenge. The reader verifies the received data and responds with the decrypted tag challenge. The BMS verifies the reader's response.

Keys derivation. After the authentication, a secure channel is established. First, session keys need to be generated and derived using a Key Derivation Function (KDF) with $K_d = KDF(K_M \,||\, dev\_add\_data \,||\, seed \,||\, padding)$, where K_M is the stored master key, dev_add_data is optional and can be production data, the $seed$ is made from concatenating the nonces from the authentication step, with $padding$ being used for rounding up. In the case of an AEAD schema, only one key is necessary. Otherwise, a MAC key different from the encryption key is also derived from $K_d = (K_{Enc} \,||\, K_{MAC})$. An important detail to the design is to provide enough entropy between the authentication and key derivation procedure as not to allow the attackers to exploit it through a replay attack. Since the authentication uses a symmetric encryption operation, the KDF function should not have blocks in its derivation that use the same keys and procedures. An example would be the Cipher-MAC (CMAC) which if used for the KDF, would also use the encryption operation possibly based on the same original key. To circumvent this, it is advisable to do either one of the following: (i) adding a guard against specialized particular nonces, i.e., not allowing re-usable challenge nonces, all zeros, etc., and using double encryption operations during the authentication step to hide the single encryption values, or (ii) to use a KDF with completely separate operations from the authentication step such as Hash-MAC (HMAC).

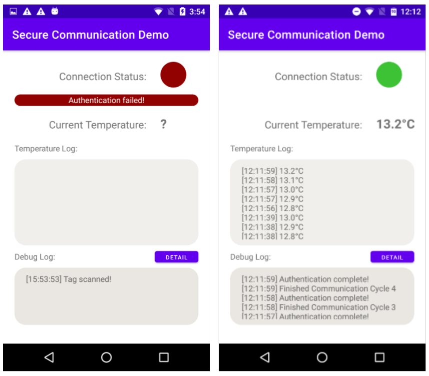
Communication session. The mobile reader and the BMS module communicate over the NFC using the SNDEF structure. The security is provided through the use of encryption, data integrity and authentication checks. The underlying protocols are either AES+MAC or the AEAD algorithms based on the system's availability. The Encrypt-then-MAC method is used for the best security mode protection. The design also uses native security mechanisms found within NFC devices. This includes the limitations of the *Write* command to the BMS device in case only the *Read* process has been called.

IV. Evaluation

A. Prototype Implementation

To evaluate the proposed architecture and test its feasibility in an applicable scenario, an adequate prototype was implemented. It consists of a mobile phone with an integrated NFC functionality that fulfils the role of a mobile reader. For this purpose, a Motorola Moto X running Android 6.0 was used. The BMS setup consists of NXP Semiconductor components that mimic real-world usage. An S32K144 MCU was used as the main BMS controller. It communicates with a battery pack consisting of MC33771C as the BCC and a battery cell emulator module. An NTAG Type 5 was used for the NFC interface of the BMS as an NFC-enhanced module communicating via an I2C connection.

The security capabilities are provided through the native Android SDK for the mobile phone, while the BMS MCU relies on an integrated Cryptographic Services Engine compressed (CSEc) [27] which implements the Secure Hardware Extension (SHE) specification [28]. It provides basic security functions such as the Random Number Generator (RNG), secure keys storage, and AES-CBC+CMAC cipher suite, while the testing of the AE functions was done using the BearSSL security library [29]. The main prototype components can be seen in Fig 4. Appropriate software extensions were implemented into the BMS monitoring and diagnostic firmware to handle the added protocol extensions while still allowing for the normal workflow of the basis system. Furthermore, a graphical application was developed for the mobile phone to test the usability of the main functions. Some of its application outputs can be seen in Fig 5, displaying the results after a failed and a successful authentication procedure. This setup was used for the further security and performance analysis.



3:54

Secure Communication Demo

Connection Status:

Authentication failed!

Current Temperature: ?

Temperature Log:

Debug Log: DETAIL

[15:53:53] Tag scanned!

12:12

Secure Communication Demo

Connection Status:

Current Temperature: 13.2°C

Temperature Log:

```
[12:11:59] 13.2°C
[12:11:58] 13.1°C
[12:11:57] 13.0°C
[12:11:57] 12.9°C
[12:11:56] 12.8°C
[12:11:39] 13.0°C
[12:11:38] 12.9°C
[12:11:38] 12.8°C
```

Debug Log: DETAIL

```
[12:11:59] Authentication complete!
[12:11:59] Finished Communication Cycle 4
[12:11:58] Authentication complete!
[12:11:58] Finished Communication Cycle 3
[12:11:57] Authentication complete!
```

Fig. 5. Developed prototype mobile application used for the evaluation.

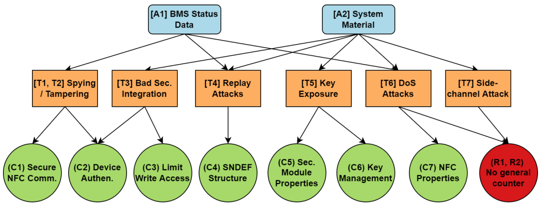


Fig. 6. GSN visual model representation of the Threat analysis.

The conducted evaluation is applicable to both scenarios discussed in Section III since the used security protocols remain the same.

B. Security Threat Analysis

A comprehensive security investigation has been conducted to evaluate the applicability of the proposed design [30]. The analysis has been summarized through the specification of Assets (A), Threats (T), Countermeasures (C), and Residual Risks (R), that are derived based on the specifics of our design and investigated BMS threat concerns [20], [22]–[24]. An illustrative representation of the threat analysis was done using Goal Structuring Notation (GSN) modelling shown in Fig. 6.

The system assets that need to be protected are:

- [A1] *BMS status data*: functional data, i.e., diagnostic or sensor measured data.
- [A2] *System configuration material*: considers general configuration data, firmware, and security material.

Each potential threat is listed followed with a short description of countermeasures, or possible residual risks, i.e., in case the threat cannot be mitigated, along with their target assets.

- [T1] Eavesdropping on the RF channel.
  $\rightarrow (A1)$. (C1) Securing the RF channel via the proposed design using encrypted session channel with MAC check.
- [T2] Channel data tampering and malicious configs.
  $\rightarrow (A1)$. (C2) Authenticating the involved parties, but also employing (C1) with MAC validation.

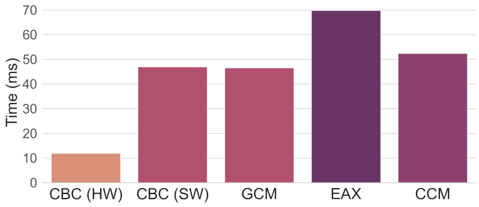


Fig. 7. Crypto-algorithm operations on 192-bytes of secret data.

- [T3] Faulty crypto. software implementation and bugs.
  $\rightarrow (A2)$. (C2) Entity authentication, and (C3) limiting write access to only allowed memory space.
- [T4] Replay attacks through MitM manipulation.
  $\rightarrow (A1, A2)$. (C4) SNDEF counter message field in combination with the unique key for each session.
- [T5] Security material, (master, session) keys exposure.
  $\rightarrow (A2)$. (C5) Security module storage properties, (C6) key management which involves KDF & key exchange.
- [T6] Denial of Service (DoS).
  $\rightarrow (A1, A2)$. Certain attacks partially mitigated via the (C7) NFC properties, with (R1) no general counteract.
- [T7] Side-channel attacks concerning extra ports.
  $\rightarrow (A2)$. (R2) No direct countermeasures;

C. Performance Analysis

The first point of focus was set on comparing the performance of different AES-based encryption algorithms usable under the proposed environment. We focused on comparing the traditional AES-CBC scheme alongside AES-GCM, AES-EAX, and AES-CCM of the AEAD package. The evaluation was done on a 192-bytes application payload also including the CMAC, i.e., tag calculations. The result of the analysis can be seen in Fig. 7. Additionally to the software implementation of the AES-CBC, we have also compared the hardware implementation using the CSEc SM. As it can be concluded, the AES-CBC hardware execution results in the fastest time, followed by the AES-GCM for the AEAD solutions. AES-GCM generally also has a better implementation support compared to other AEAD algorithms. Therefore, for the rest of the analysis, we will focus only on these two algorithms.

The performance analysis of the implemented prototype is set in a loop running enclosed process cycles. Each cycle consists out of the period for the device authentication between the BMS and the mobile phone, followed with session key derivation, encryption and exchange of 192 bytes of test data, and waiting for the readout from the mobile phone. Time measurements are derived as averages for each important cycle step after multiple runs. Measurements were split between the authentication and the secure transmission phase, shown in Fig. 8 respectively, with the transmission step considering the AES-CBC+CMAC functions. Each step shown considers the time of the respective data and security handling functions, as well as the NFC reading and writing operations which were the main contributors to the total execution time.

TABLE I
MEASURED TOTAL EXECUTION TIMES FOR RESPECTIVE PROCESS PHASES

Authentication	Secure Transmission AES-CBC+CMAC	Secure Transmission AES-GCM
$78.61\,ms \pm 1.53\,ms$	$114.34\,ms \pm 1.98\,ms$	$145.64\,ms \pm 2.09\,ms$

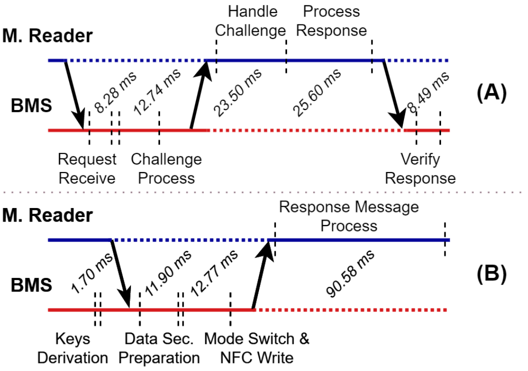


Fig. 8. Timeline diagram illustrating sequence steps and presenting time measurements for: (A) Authentication phase, (B) Secure Transmission phase.

It has been observed that repeatedly throughout the communication loop, each cycle takes roughly a constant amount of time. The time variations relied mainly on the used underlying security protocols, as well as the reliability of the NFC connection, which greatly depends on the positioning of the NFC components, i.e., the position of the NFC reader's relative to the NTAG. The total execution times are shown in Table I. The resulting performance time is deemed sufficient for the intended use-case.

V. Conclusion and Future Work

In the course of this paper, a novel system design approach was presented for a secure interaction and data exchange between a BMS and a mobile control reader. The proposed design is based on a wireless communication concept utilizing NFC technology. It is intended to be suitable for different active and passive BMS use-cases, regardless of whether the data acquisition is handled through an actual BMS controller or a modulated battery pack. An NFC security record SNDEF was presented along with lightweight symmetric cryptography measures. These security enchantments provide entity authentication and a secure channel for data confidentiality and integrity protection during the mobile readout process. The SNDEF accounts both for the traditional, as well as AEAD cryptography schemes. A demonstrative prototype was implemented for the purpose of functional verification, and security and performance evaluation. For future work, we consider an alternative design with asymmetric cryptography schemes for devices that support them. These could benefit in an expanded security architecture by offering forward secrecy and potential remote cloud support.

Acknowledgment

This project has received funding from the "EFREtop: Securely Applied Machine Learning - Battery Management Systems" (Acronym "SEAMAL BMS", FFG Nr. 880564).

References

[1] W.-C. Lih, J.-H. Yen, F.-H. Shieh, and Y.-M. Liao, "Second Use of Retired Lithium-ion Battery Packs from Electric Vehicles: Technological Challenges, Cost Analysis and Optimal Business Model," in *IS3C*, 2012.

[2] N. Nicholas, "End-of-life Electric vehicle batteries: Recycling or second-life?." https://www.smart-energy.com/features-analysis/end-of-life-electric-vehicle-batteries-recycling-or-second-life/, 2020. Accessed: 26.11.2021.

[3] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles," *IEEE Industrial Electronics Magazine*, vol. 7, 2013.

[4] A. Reindl *et al.*, "Scalable, Decentralized Battery Management System Based on Self-organizing Nodes," in *ARCS*, pp. 171–184, 2020.

[5] M. K. Hasan *et al.*, "Review of electric vehicle energy storage and management system: Standards, issues, and challenges," *Journal of Energy Storage*, vol. 41, 2021.

[6] P. Sun, R. Bisschop, H. Niu, and X. Huang, "A Review of Battery Fires in Electric Vehicles," *Fire Technology*, pp. 1–50, 01 2020.

[7] T. Ulz *et al.*, "SECURECONFIG: NFC and QR-code based Hybrid Approach for Smart Sensor Configuration," in *IEEE RFID*, 2017.

[8] E. Haselsteiner *et al.*, "Security in Near Field Communication (NFC). Strengths and Weaknesses," in *Workshop on RFID Security*, 2006.

[9] P. Bansal and P. Nagaraj, "Wireless Battery Management System for Electric Vehicles," in *IEEE ITEC-India*, pp. 1–5, 2019.

[10] C. Shell *et al.*, "Implementation of a Wireless Battery Management System (WBMS)," in *IEEE I2MTC*, pp. 1954–1959, 2015.

[11] G. De Maso-Gentile *et al.*, "Design of CAN to Bluetooth gateway for a Battery Management System," in *12th WISES*, pp. 171–175, 2015.

[12] A. Rahman, M. Rahman, and M. Rashid, "Wireless Battery Management System of Electric Transport," *IOP Conference Series: Materials Science and Engineering*, vol. 260, nov 2017.

[13] T. Gherman *et al.*, "Smart Integrated Charger with Wireless BMS for EVs," in *44th IECON*, pp. 2151–2156, 2018.

[14] A. Samanta and S. S. Williamson, "A Survey of Wireless Battery Management System: Topology, Emerging Trends, and Challenges," *Electronics*, vol. 10, no. 18, 2021.

[15] F. Basic, M. Gaertner, and C. Steger, "Towards Trustworthy NFC-based Sensor Readout for Battery Packs in Battery Management Systems," in *IEEE RFID-TA*, pp. 285–288, 2021.

[16] M. Roland *et al.*, "Security Vulnerabilities of the NDEF Signature Record Type," in *Third International Workshop on NFC*, 2011.

[17] M. Q. Saeed *et al.*, "A Record Composition/Decomposition attack on the NDEF Signature Record Type Definition," in *ICITST*, 2011.

[18] T. Ulz *et al.*, "QSNFC: Quick and Secured Near Field Communication for the Internet of Things," in *IEEE RFID*, pp. 1–8, 2018.

[19] P. Urien *et al.*, "LLCPS and SISO: A TLS-based Framework with RFID for NFC P2P Retail Transaction Processing," in *IEEE RFID*, 2013.

[20] M. Cheah and R. Stoker, "Cybersecurity of Battery Management Systems," *HM TR series*, vol. 10, no. 3, p. 8, 2019.

[21] S. Plosz *et al.*, "Security Vulnerabilities and Risks in Industrial Usage of Wireless Communication," in *IEEE ETFA*, pp. 1–8, 2014.

[22] S. Sripad *et al.*, "Vulnerabilities of Electric Vehicle Battery Packs to Cyberattacks," *ArXiv*, 2017.

[23] A. Khalid *et al.*, "FACTS Approach to Address Cybersecurity Issues in Electric Vehicle Battery Systems," in *IEEE TEMSCON*, pp. 1–6, 2019.

[24] S. Kumbhar *et al.*, "Cybersecurity for Battery Management Systems in Cyber-Physical Environments," *ITEC 2018*, pp. 761–766, 2018.

[25] N. A. Chattha, "NFC — Vulnerabilities and Defense," in *Conference on Information Assurance and Cyber Security (CIACS)*, pp. 35–38, 2014.

[26] C. H. Chen, I. C. Lin, and C. C. Yang, "NFC Attacks Analysis and Survey," in *8th IMIS*, pp. 458–462, 2014.

[27] NXP Semiconductors, *Application Note: Getting Started with CSEc Security Module*, AN5401 Rev. 1 ed., 03 2018.

[28] AUTOSAR, *Specification of Secure Hardware Extensions*, 2019.

[29] "BearSSL." https://bearssl.org/, 2021. Accessed: 22.12.2021.

[30] S. Myagmar, A. Lee J., and W. Yurcik, "Threat Modeling as a Basis for Security Requirements," *in SREIS*, 2005.

A.5 [E] Trust your BMS: Designing a Lightweight Authentication Architecture for Industrial Networks

F. Basic, C. Steger, C. Seifert and R. Kofler, "Trust your BMS: Designing a Lightweight Authentication Architecture for Industrial Networks," in *2022 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1-6, IEEE, 2022.

Abstract. With the advent of clean energy awareness and systems that rely on extensive battery usage, the community has seen an increased interest in the development of more complex and secure Battery Management Systems (BMS). In particular, the inclusion of BMS in modern complex systems like electric vehicles and power grids has presented a new set of security-related challenges. A concern is shown when BMS are intended to extend their communication with external system networks, as their interaction can leave many backdoors open that potential attackers could exploit. Hence, it is highly desirable to find a general design that can be used for BMS and its system inclusion. In this work, a security architecture solution is proposed intended for the communication between BMS and other system devices. The aim of the proposed architecture is to be easily applicable in different industrial settings and systems, while at the same time keeping the design lightweight in nature.

My Contribution. For this publication, I provided the main design, which includes system challenges and requirements, system design, and security protocols based on symmetric cryptography, and also asymmetric cryptography using implicit certificates. I developed and expanded on the previous security models and protocols that rely on the use of the ECQV scheme. I provided the main design for the BMS security architecture. To show the design's feasibility, I also contributed to realizing the implementation of the protocols on the real BMS sub-system, and evaluating it on the performance and provided security. I also wrote the majority of the text. Christian Seifert provided the implementation of the security modules and serial interface drivers used in the evaluation. Christian Steger provided input on the paper structure, figure specification, and insight in relation to the SotA.

Trust your BMS: Designing a Lightweight Authentication Architecture for Industrial Networks

Fikret Basic, Christian Steger, Christian Seifert
Institute of Technical Informatics
Graz University of Technology
Graz, Austria
{basic, steger, christian.seifert}@tugraz.at

Robert Kofler
R&D Battery Management Systems
NXP Semiconductors Austria GmbH Co & KG
Gratkorn, Austria
robert.kofler@nxp.com

***Abstract*—With the advent of clean energy awareness and systems that rely on extensive battery usage, the community has seen an increased interest in the development of more complex and secure Battery Management Systems (BMS). In particular, the inclusion of BMS in modern complex systems like electric vehicles and power grids has presented a new set of security-related challenges. A concern is shown when BMS are intended to extend their communication with external system networks, as their interaction can leave many backdoors open that potential attackers could exploit. Hence, it is highly desirable to find a general design that can be used for BMS and its system inclusion. In this work, a security architecture solution is proposed intended for the communication between BMS and other system devices. The aim of the proposed architecture is to be easily applicable in different industrial settings and systems, while at the same time keeping the design lightweight in nature.**

***Index Terms*—Battery Management System; Security; Keys; Implicit Certificates; ECQV; Authentication; Networks.**

I. Introduction

Many systems today rely on large sets of battery cells as power sources. These battery cells are usually packed together in serial or parallel connections. As the number of these battery cells increases, so does the need for systems that are able to control and automatically respond to different conditions and situations [1]. This control is handled through Battery Management Systems (BMS). Today, their usage is rapidly expanding as they are found as part of many different smaller and larger systems. With the increase of the importance of clean energy, BMS are slowly becoming a topic in a broad variety of fields. Prominent use-cases include hybrid and electric vehicles, and smart power grids, where BMS integration is of critical importance for a safe and efficient energy control [2]–[4]. BMS helps in preventing incidents like the thermal runaway that occurs during the expeditious increase of the battery cell temperature, which would otherwise be difficult to detect [5].

Each BMS usually consists of a main BMS controller, individual Battery Cell Controllers (BCC), and a battery module which contains battery cells, corresponding sensors and interfaces. Traditionally, BMS were deployed as relatively simple sub-systems with limited interaction with the outside components and services. However, when transitioning to larger networks and systems, a special attention needs to also be given in the form of protection against malicious attacks [6]. If a device is compromised that is either part of the BMS or the general network, it would give the possibility for a malicious user to mount different attacks. Specifically, an attacker might try to gain a direct access to the system, manipulate system data, or even compromise privacy of an user profile [7]–[9].

BMS in industrial environments need to be carefully administrated and often require configuration and status updates. These are often done today through external services, such as cloud [10] or remote configuration approaches [11], and a gateway device. However, in internal networks that connect the BMS to the gateway and other components, security is often neglected due to its complexity and design demands. A similar concern has also been addressed in larger smart power grid systems [12], [13]. Based on our analysis, we see the following security matters that needs to be addressed: (i) configuration data manipulation via exposed interfaces, (ii) industry espionage through Man-in-the-Middle (MitM) attacks, (iii) physical compromise through unauthorized access with a counterfeited or malicious devices.

To address the previously mentioned challenges and security issues, we consider a design which takes into the account the following conditions: consider the following requirements:

- *Portability*: the design needs to allow the exchange and validation of modules between different systems.
- *Small footprint*: the implemented security blocks need to be lightweight and not interfere with other operations.
- *Accessibility*: usable between different vendors.
- *Security*: secure under the given operational conditions.

We consider the use of the implicit certificates, specifically the Elliptic Curve Qu-Vanstone (ECQV) schema, for establishing fast and efficient network authentication. The use of implicit certificates for in-vehicle authentication has already been previously investigated [14]. However, no specific analysis has yet been conducted related to the use of BMS and its connected services. In this work, we propose an efficient and lightweight design approach for establishing authentication and secure channel communication for BMS and related communication devices. To the best of our knowledge, no other work that investigates this security architectural approach in BMS has been previously proposed.

Contributions. Summarized, our main contributions contained in this paper are following: (i) proposing a BMS secure design architecture for communication with external devices in closed networks, (ii) presenting an authentication protocol based on the implicit certificates, and session key derivation, (iii) using a BMS test device and controllers, we implement the proposed solution and evaluate the process.

II. Background and Related Work

A. BMS Security Concepts

A BMS usually consists of several distinct units. A main BMS controller can communicate with one or many BCCs which in turn can also be connected to one or many battery cell packs. This results in two main security environments that need to be addressed: internal component security, and external service communication. As a relatively new topic that slowly gains interest, research has been mainly focused on the theoretical BMS security models based on the general threat analysis methods [6], [9]. While researchers primarily concentrate on the general BMS security models, Fuchs et al. [15] shows a design that uses a Trusted Platform Module (TPM) for establishing a secure communication between BMS and Electric Vehicle Charging Controllers (EVCC). On the other hand, researchers have also been interested in the BMS cloud environment, proposing design solutions with limited security design considerations [16], [17]. In this work, we try to bridge the gap between the end point of the BMS controller and direct communication devices to present a design that can be applied for general BMS authentication questions.

B. Authentication Approaches in the Automotive Industry

Since BMS today play a vital role in the vehicles domain, we have also investigated the State-of-the-Art (SOTA) security architectures inside the vehicle communication environment. Hazem et al. [18] present a protocol for incorporating authentication with the traditional CAN communication protocol. Research conducted by Mundhenk et al. [19] showed an earlier design proposal that includes both the device authentication and secure session establishment between Electronic Control Units (ECUs) in a vehicle. Device authentication is based on combining both asymmetrical and symmetrical crypto approaches and relies on a central security module for control. Similarly, work described in [14] extends on the lightweight notion and introduces a general design for in-vehicle authentication of ECUs utilizing Physical Unclonable Functions (PUFs) for the initial device authentication and furthermore implicit certificates for subsequent authentication and key derivation. We do not consider using PUFs for several reason. Mainly, our target features are portability and ease of use of the already established security architectures found in industrial systems and vehicles, especially those that can be established with the verified manufacturers. Furthermore, the PUFs are still largely experimental and based on the recent studies, current implementations have shown vulnerabilities to various threats including machine learning related attacks [20]–[22].

C. Implicit Certificates

In most modern architectures and networks, systems rely on the use of the explicit certificates usually coupled together with the TLS/SSL for the purpose of authentication and secure communication. However, while secure, this approach might result in a considerable overhead. Research work by Pullen et al. [14] proposed the use of implicit certificates for establishing entity authentication after the initial device authentication. Several other works have also already been conducted handling the implicit certificate implementation, specifically with IoT-related devices [23], [24]. Other work includes research conducted in [25], which focuses on the Certificate Transparency (CT) specially aimed to fit the constrained implicit certificate schematic use-cases. Implicit certificates allow for a lightweight schema without security compromise.

III. Design of a Novel BMS Security Architecture

A. Security Requirements

In an enclosed local network, authentication is an important step usually carried out before other main operations to verify devices that are interconnected. A BMS might need to communicate with additional devices, often to extend the services offered, such as logging and monitoring purposes [2]. Before this communication can take place, the BMS needs to be certain that the device it speaks to is valid and authenticated. Additionally, even if not directly communicated with, every other device inside the network needs to be already authenticed to prevent any kind of sniffing or MitM attacks that could potentially take place [9]. A potential attacker might either try to attack a BMS for the purpose of reverse engineering and technology exploitation, or data compromise for ransom, frauds, or simply vandalism.

B. System Architecture

Our solution is aimed at the modulated BMS topology that uses a central main controller to handle the control of battery packs through BCCs [26]. The proposed architecture can also be used for distributed BMS topologies, as each main BMS controller is seen as a separate unit. Through our proposed design the communication to the outside world from the enclosed BMS is only performed through the main control device. This ensures that the main threats, and with that the protection, would be focused on the connection point that the BMS has with external devices.

The proposed architecture consists of (Fig. 1):

- *BMS sub-system*: complete modules that include battery controllers and battery packs.
- *Secure Edge Device* (SED): a device which is used both for the device authentication and certificate creation, and represents the Central Authority (CA) for the local network in this case. It needs to securely handle credential data and fulfill the Common Criteria (CC) conditions.
- *Control Units*: ad-hoc devices attached to the system network, either internally or externally, that want to authenticate a BMS, and need to be authenticated itself.

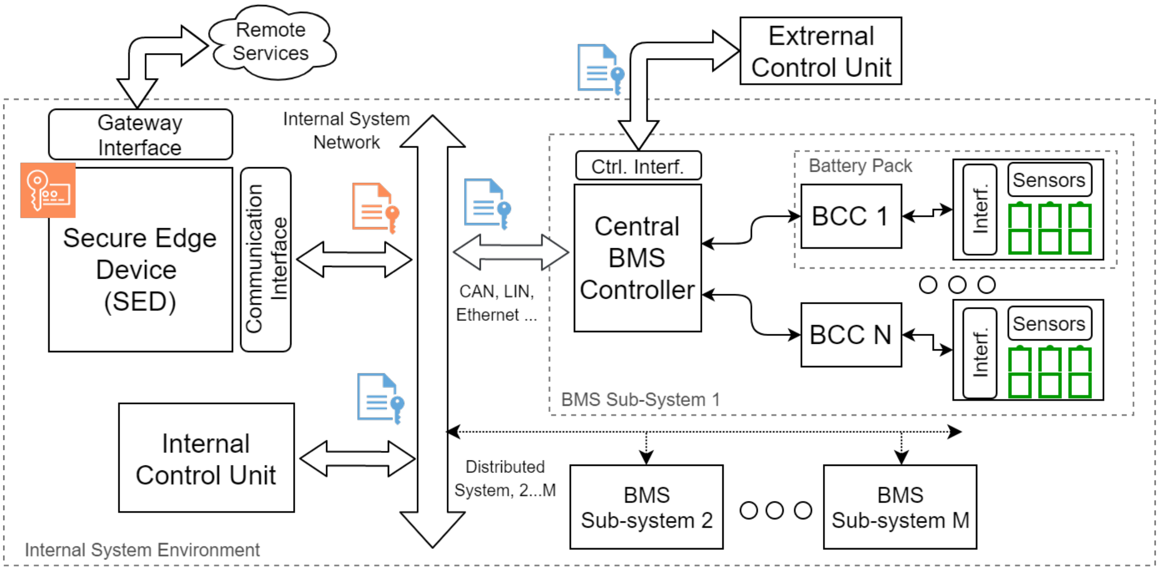


Fig. 1. Block demonstration of the proposed security architecture, suggested modules and connections points for the industry systems that contain one or several BMS sub-systems and control devices that interact with them. It showcases potential points of placement regarding SED, BMS and the Control units.

We assume that the targeted network is closed, i.e., only the SED has access to the outside services (e.g., cloud, monitoring devices). Additionally, any other external communication access (e.g., diagnostic tools) would as well need to be verified first as a trusted source by the SED before establishing connection with other devices in the network.

C. Security Model

To establish a secure authentication and communication procedure between the BMS and the corresponding devices, a security model was established consisting out of four consecutive steps: (1) fabrication; (2) device authentication; (3) certificate derivation, (4) session communication. Notations used for figures and algorithms are shown in Table I.

The device authentication is proceeded with the **fabrication** step during which devices are pre-embedded with the necessary security material. This phase is performed only once during the manufacturing stage.

Device authentication step (Fig. 2) uses the Message Authentication Code (MAC) operation for the purpose of handling the authentication procedure. With this, both the BMS and the SED are able to authenticate each other. This process is intended to be run only once when a new device is detected on the network to avoid performance and timing constraints. The handling is based on the challenge and response mechanism with a *pre-shared key*. Both the SED and the BMS should have a pre-installed secret identifier that can be configured through other secure means [11], with the initial one being established during the fabrication step and used for further key-derivations. Dynamic nonce handling is added for extra protection which includes nonce generation on both entity sides, and the nonce summation and encryption validation [14]. The challenge issued by the SED are concatenated with the random nonces on the BMS side, which is then encrypted and handled with MAC. The extra encryption process helps in preventing potential MitM attacks, particularly replay attacks.

TABLE I
NOTATIONS ABBREVIATION LIST

Symbol	Description
N	Field key size
C	Random auth. challenge
key_{auth}	Key used for the device auth.
key_{enc}, key_{mac}	Auth. encryption & MAC keys
N_{SED}, N_{BMS}, N_{SUM}	Auth. random nonces
ID_{BMS}	BMS unique identif. number
R	Response auth. message
t_{BMS}, k_{BMS}	Random private int. values
P_{BMS}	Cert. req. EC point
U_{BMS}, S_{BMS}	Keys contribution recon. data
$Cert$	Encoded device certificate
prk_i, pub_i	Private & public key of device ‘i’
ID_{Sess}	Device unique session ID
chg_i, $resp_i$	Auth. challenge & response
k_s	Symmetric session key

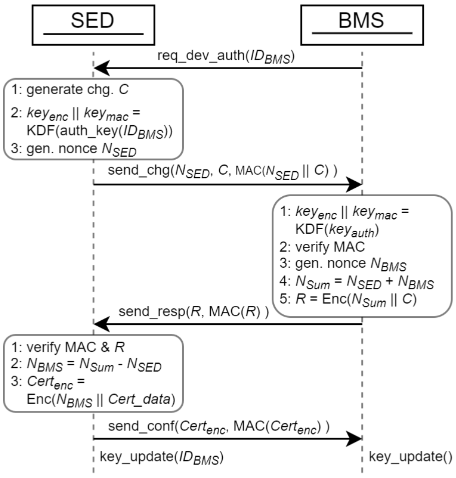

SED
BMS
req_dev_auth(ID_{BMS})
1: generate chg. C
2: key_{enc} || key_{mac} = KDF(auth_key(ID_{BMS}))
3: gen. nonce N_{SED}
send_chg(N_{SED}, C, MAC(N_{SED} || C))
1: key_{enc} || key_{mac} = KDF(key_{auth})
2: verify MAC
3: gen. nonce N_{BMS}
4: N_{Sum} = N_{SED} + N_{BMS}
5: R = Enc(N_{Sum} || C)
send_resp(R, MAC(R))
1: verify MAC & R
2: N_{BMS} = N_{Sum} - N_{SED}
3: $Cert_{enc}$ = Enc(N_{BMS} || $Cert\_data$)
send_conf($Cert_{enc}$, MAC($Cert_{enc}$))
key_update(ID_{BMS})
key_update()

Fig. 2. Device authentication process.

Certificate derivation (Fig. 3) follows after the device authentication to complete the configuration process of the newly recognized device. This step is important since the certificates can be afterwards used for verification between the BMS and any other device that is part of the network based on the asymmetric cryptography principle. Certificate authentication data is derived and exchanged. To make this possible, during the device authentication, configuration data is sent from the SED to BMS, which contains: a session ID, algorithm identifier (curve, hash), SED's public key and ID.

The authentication algorithm uses the **implicit certificates** with the ECQV as the targeted schema for the purpose of deriving and exchanging certificates [27]. Based on the proposed ECQV documentation and the ANS.1 format, we decided to use the Minimal Encoding Scheme (MES) without additional extensions for our certificates. The main reason is the smaller certificate sizes, and therefore faster processing than the traditional X.509 format.

The BMS initiates the request for the certificate validation by calculating its necessary construction data, deriving a random nonce, and calculating the MAC value with the previously updated authentication key based on the pre-shared key. Session ID is used to confirm the request. A new session ID is derived on each new device authentication step and is unique for each system device. After verifying the request, the SED will derive the necessary certificate and key construction using Algorithm 1. Afterwards, a response will be generated and sent back to the BMS where it will first verify the authenticity of the messages based on its MAC and nonce and then proceed with calculating its private and public keys. This key derivation procedure is described by Algorithm 2.

Algorithm 1: SED implicit certificate formulation.

```
Input: $ID_{Sess}$, $P_{BMS}$
Output: $S_{BMS}$, $Cert$
Generate $k_{BMS} \in_R [1, ..., n-1]$
$U_{BMS} \leftarrow P_{BMS} + k_{BMS} * G$
$Cert \leftarrow Encode(ID_{Sess}, U_{BMS})$
$S_{BMS} \leftarrow (Hash(Cert) * k_{BMS} + prk_{SED} * G) \ mod \ n$
return $S_{BMS}$, $Cert$
```

Algorithm 2: BMS implicit certificate keys derivation.

```
Input: $S_{BMS}$, $Cert$
Output: $prk_{BMS}$, $pub_{BMS}$, status
$prk_{BMS} \leftarrow (Hash(Cert) * k_{BMS} + S_{BMS}) \ mod \ n$
$pub_{BMS} \leftarrow Hash(Cert) * Decode(Cert) + pub_{SED}$
if $pub_{BMS} == prk_{BMS} * G$ then
    return $prk_{BMS}$, $pub_{BMS}$
else
    return $false$
end
```

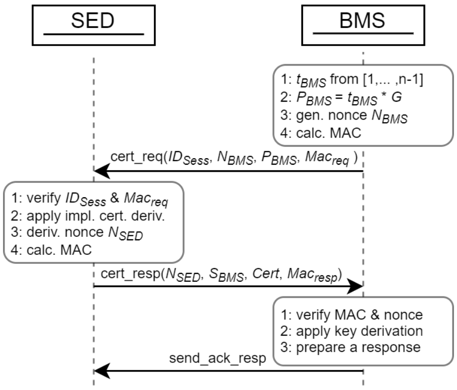

SED
BMS
1: t_{BMS} from [1,... ,n-1]
2: P_{BMS} = t_{BMS} * G
3: gen. nonce N_{BMS}
4: calc. MAC
cert_req(ID_{Sess}, N_{BMS}, P_{BMS}, Mac_{req})
1: verify ID_{Sess} & Mac_{req}
2: apply impl. cert. deriv.
3: deriv. nonce N_{SED}
4: calc. MAC
cert_resp(N_{SED}, S_{BMS}, $Cert$, Mac_{resp})
1: verify MAC & nonce
2: apply key derivation
3: prepare a response
send_ack_resp

Fig. 3. Certificate derivation process.

Session communication phase (Fig. 4), is lastly used during a defined session when two devices other than the SED want to mutually authenticate and derive session keys, e.g., the BMS sub-system with a control unit. This phase is coupled together with the certificate derivation for performance reasons since the derived session keys are based on the current public key value and the long-term device private keys [23].

D. Discussion on Security Material Updates

To guarantee a partial *forward secrecy*, i.e., in case older authentication keys are compromised, the keys used in the device authentication phase are updated after each authentication cycle. A Key Derivation Function (KDF) is used to derive new keys based on the previous key and the current request nonce. The initial authentication keys have to be pre-embedded during the fabrication step. With this procedure, even if earlier keys get compromised, the attacker needs to have caught all the previous authentication session interactions and the request nonces to be able to correctly derive the current valid authentication key.

For the certification derivation phase, an open question is made on when should the *re-certification* take place, i.e., when should the new certificates be generated and exchanged. It highly depends on the applications needs, but it is certain to

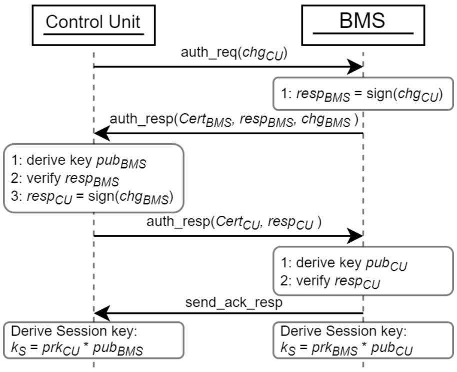


Fig. 4. Mutual authentication and session key establishment.

happen at least when certificates expire or during a new system start-up. Otherwise, we propose that the device authentication and re-configuration happen under the following conditions: (i) installation of a new device, (ii) configuration or firmware updates, (iii) changes in the certificate configuration.

IV. Evaluation

A. Prototype Implementation

To evaluate our proposed design approach and analyse its applicability and usability, a prototype test suite was implemented and tested. It was aimed to use higher grade industry-applicable components with the intention of more closely depicting the real-world systems. The test suite consists of a full BMS emulation equipment and a Raspberry Pi 4 functioning as a SED. The setup is shown in Fig. 5.

For the BMS setup, a S32K144 MCU board was used as a central BMS controller. This controller is connected to MC33771C which function as the BCC. Furthermore, a BATT-14CEMULATOR was used for the emulation of battery cells. The connection between the Raspberry Pi 4 and the BMS controller was established using serial communication with a protocol developed for message handling. SED functionalities have been implemented in Python, with appropriate security handlers using the cryptography library. Encryption is done with the AES-CBC algorithm, where hash (H)MAC is used for the MAC calculations. The lightweight *BearSSL* library was used for the elliptic curve and certificate-related operations. The security software implementation was carefully handled as to still allow the normal flow of the BMS safety control.

B. Threat Model Analysis

To test the security feasibility of our design as well as the achieved security level, we have conducted a comprehensive threat model analysis [28]. The analysis is based on the common attacks indicated by the investigated BMS threat models in [6]–[9]. We assume that the attacker has enough resources and knowledge to launch the potential attacks and that any communication outside of the system is deemed unsafe. We derive the involved Assets (A), Threats (T), Countermeasures (C), and for threats that are not able to be mitigated, the potential Residual Risks (R). Afterwards, each threat is classified based on the STRIDE threat categories [29], by indicating Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.

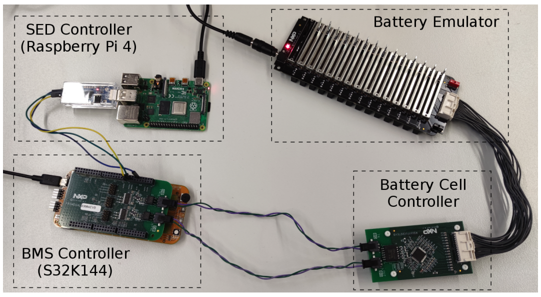


Fig. 5. Prototype demonstrator of the proposed security architecture design.

In terms of protection, the following assets need to be secured: **(A1)** *BMS operational process*: status alerts and adequate safety monitoring, **(A2)** *Status data*: configuration, raw sensor and derived safety status data, **(A3)** *Network integrity*: device connectivity, and port access.

The following threats and countermeasures are observed:

- **(T1)**⟨S,T,R,I,E⟩ *Malicious update*: attack through configuration data or even code injections. Mitigated by **(C1)** *Authentication procedure* as proposed in this paper.
- **(T2)**⟨I⟩ *Network eavesdrop*: if the attacker gains access to the internal system network. Protected through **(C1)**, but also **(C2)** *Encrypted channel*.
- **(T3)**⟨T,I⟩ *System data compromise*: affects vulnerable devices that are not properly configured. Either mitigated by **(C1)** & **(C2)**, or not by **(R1)** *No secure configuration*.
- **(T4)**⟨S,T,R,I,D⟩ *Node capturing attacks*: as described in [30]. Handled via **(C3)** *Frequent certificate update control*, and **(C4)** *Dynamic key updates*.
- **(T5)**⟨S,T,R,I,E⟩ *Previous key exposure*: vulnerability depends on the system design and configuration of the updates. Limited protection with **(C4)** *Forward secrecy*, or, depending on the configuration, **(R2)** *Updates neglect*.
- **(T6)**⟨S,T,R,I,E⟩ *Credentials exposure*: targets either the stored or communicated security material. Mitigated via SED and **(C5)** *Central access control*.
- **(T7)**⟨S,T,R,I⟩ *Counterfeited devices*: fake devices or devices with malicious intent. Protected with **(C1)**.

C. Performance Analysis

To evaluate the application of the design under operational conditions, an execution time analysis has been conducted for critical tasks and steps. Measurements have been run through multiple iterations on both the BMS controller (Table II) and the SED (Table III) noting an average value for each vital operation; each noted time includes reading the request, operation handling, and preparing and sending the response.

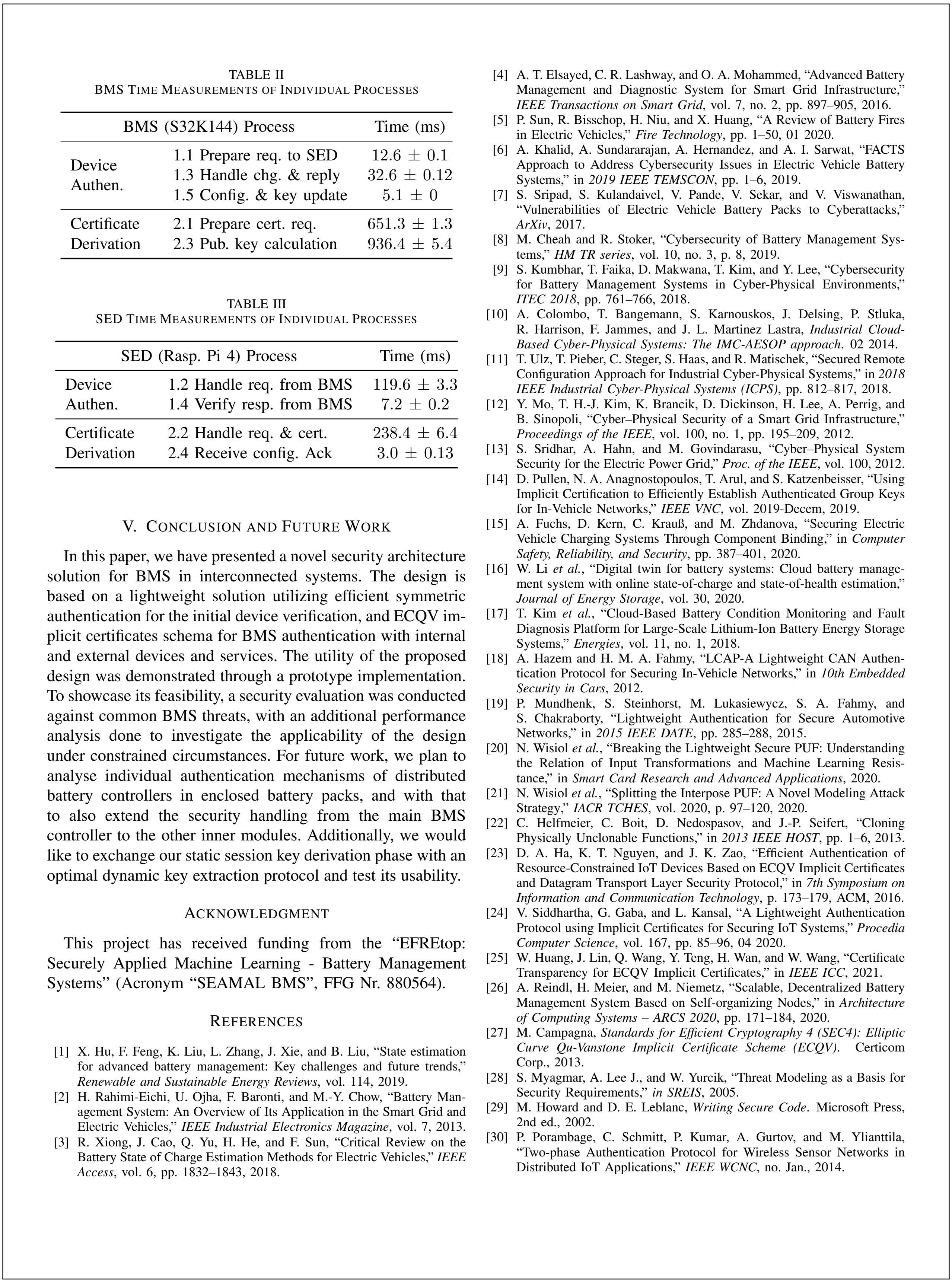


TABLE II
BMS Time Measurements of Individual Processes

BMS (S32K144) Process		Time (ms)
Device Authen.	1.1 Prepare req. to SED	12.6 ± 0.1
	1.3 Handle chg. & reply	32.6 ± 0.12
	1.5 Config. & key update	5.1 ± 0
Certificate Derivation	2.1 Prepare cert. req.	651.3 ± 1.3
	2.3 Pub. key calculation	936.4 ± 5.4

TABLE III
SED Time Measurements of Individual Processes

SED (Rasp. Pi 4) Process		Time (ms)
Device Authen.	1.2 Handle req. from BMS	119.6 ± 3.3
	1.4 Verify resp. from BMS	7.2 ± 0.2
Certificate Derivation	2.2 Handle req. & cert.	238.4 ± 6.4
	2.4 Receive config. Ack	3.0 ± 0.13

V. Conclusion and Future Work

In this paper, we have presented a novel security architecture solution for BMS in interconnected systems. The design is based on a lightweight solution utilizing efficient symmetric authentication for the initial device verification, and ECQV implicit certificates schema for BMS authentication with internal and external devices and services. The utility of the proposed design was demonstrated through a prototype implementation. To showcase its feasibility, a security evaluation was conducted against common BMS threats, with an additional performance analysis done to investigate the applicability of the design under constrained circumstances. For future work, we plan to analyse individual authentication mechanisms of distributed battery controllers in enclosed battery packs, and with that to also extend the security handling from the main BMS controller to the other inner modules. Additionally, we would like to exchange our static session key derivation phase with an optimal dynamic key extraction protocol and test its usability.

Acknowledgment

This project has received funding from the "EFREtop: Securely Applied Machine Learning - Battery Management Systems" (Acronym "SEAMAL BMS", FFG Nr. 880564).

References

[1] X. Hu, F. Feng, K. Liu, L. Zhang, J. Xie, and B. Liu, "State estimation for advanced battery management: Key challenges and future trends," *Renewable and Sustainable Energy Reviews*, vol. 114, 2019.

[2] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles," *IEEE Industrial Electronics Magazine*, vol. 7, 2013.

[3] R. Xiong, J. Cao, Q. Yu, H. He, and F. Sun, "Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles," *IEEE Access*, vol. 6, pp. 1832–1843, 2018.

[4] A. T. Elsayed, C. R. Lashway, and O. A. Mohammed, "Advanced Battery Management and Diagnostic System for Smart Grid Infrastructure," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 897–905, 2016.

[5] P. Sun, R. Bisschop, H. Niu, and X. Huang, "A Review of Battery Fires in Electric Vehicles," *Fire Technology*, pp. 1–50, 01 2020.

[6] A. Khalid, A. Sundararajan, A. Hernandez, and A. I. Sarwat, "FACTS Approach to Address Cybersecurity Issues in Electric Vehicle Battery Systems," in *2019 IEEE TEMSCON*, pp. 1–6, 2019.

[7] S. Sripad, S. Kulandaivel, V. Pande, V. Sekar, and V. Viswanathan, "Vulnerabilities of Electric Vehicle Battery Packs to Cyberattacks," *ArXiv*, 2017.

[8] M. Cheah and R. Stoker, "Cybersecurity of Battery Management Systems," *HM TR series*, vol. 10, no. 3, p. 8, 2019.

[9] S. Kumbhar, T. Faika, D. Makwana, T. Kim, and Y. Lee, "Cybersecurity for Battery Management Systems in Cyber-Physical Environments," *ITEC 2018*, pp. 761–766, 2018.

[10] A. Colombo, T. Bangemann, S. Karnouskos, J. Delsing, P. Stluka, R. Harrison, F. Jammes, and J. L. Martinez Lastra, *Industrial Cloud-Based Cyber-Physical Systems: The IMC-AESOP approach*. 02 2014.

[11] T. Ulz, T. Pieber, C. Steger, S. Haas, and R. Matischek, "Secured Remote Configuration Approach for Industrial Cyber-Physical Systems," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pp. 812–817, 2018.

[12] Y. Mo, T. H.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–Physical Security of a Smart Grid Infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, 2012.

[13] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber–Physical System Security for the Electric Power Grid," *Proc. of the IEEE*, vol. 100, 2012.

[14] D. Pullen, N. A. Anagnostopoulos, T. Arul, and S. Katzenbeisser, "Using Implicit Certification to Efficiently Establish Authenticated Group Keys for In-Vehicle Networks," *IEEE VNC*, vol. 2019-Decem, 2019.

[15] A. Fuchs, D. Kern, C. Krauß, and M. Zhdanova, "Securing Electric Vehicle Charging Systems Through Component Binding," in *Computer Safety, Reliability, and Security*, pp. 387–401, 2020.

[16] W. Li *et al.*, "Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation," *Journal of Energy Storage*, vol. 30, 2020.

[17] T. Kim *et al.*, "Cloud-Based Battery Condition Monitoring and Fault Diagnosis Platform for Large-Scale Lithium-Ion Battery Energy Storage Systems," *Energies*, vol. 11, no. 1, 2018.

[18] A. Hazem and H. M. A. Fahmy, "LCAP-A Lightweight CAN Authentication Protocol for Securing In-Vehicle Networks," in *10th Embedded Security in Cars*, 2012.

[19] P. Mundhenk, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, and S. Chakraborty, "Lightweight Authentication for Secure Automotive Networks," in *2015 IEEE DATE*, pp. 285–288, 2015.

[20] N. Wisiol *et al.*, "Breaking the Lightweight Secure PUF: Understanding the Relation of Input Transformations and Machine Learning Resistance," in *Smart Card Research and Advanced Applications*, 2020.

[21] N. Wisiol *et al.*, "Splitting the Interpose PUF: A Novel Modeling Attack Strategy," *IACR TCHES*, vol. 2020, p. 97–120, 2020.

[22] C. Helfmeier, C. Boit, D. Nedospasov, and J.-P. Seifert, "Cloning Physically Unclonable Functions," in *2013 IEEE HOST*, pp. 1–6, 2013.

[23] D. A. Ha, K. T. Nguyen, and J. K. Zao, "Efficient Authentication of Resource-Constrained IoT Devices Based on ECQV Implicit Certificates and Datagram Transport Layer Security Protocol," in *7th Symposium on Information and Communication Technology*, p. 173–179, ACM, 2016.

[24] V. Siddhartha, G. Gaba, and L. Kansal, "A Lightweight Authentication Protocol using Implicit Certificates for Securing IoT Systems," *Procedia Computer Science*, vol. 167, pp. 85–96, 04 2020.

[25] W. Huang, J. Lin, Q. Wang, Y. Teng, H. Wan, and W. Wang, "Certificate Transparency for ECQV Implicit Certificates," in *IEEE ICC*, 2021.

[26] A. Reindl, H. Meier, and M. Niemetz, "Scalable, Decentralized Battery Management System Based on Self-organizing Nodes," in *Architecture of Computing Systems – ARCS 2020*, pp. 171–184, 2020.

[27] M. Campagna, *Standards for Efficient Cryptography 4 (SEC4): Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)*. Certicom Corp., 2013.

[28] S. Myagmar, A. Lee J., and W. Yurcik, "Threat Modeling as a Basis for Security Requirements," *in SREIS*, 2005.

[29] M. Howard and D. E. Leblanc, *Writing Secure Code*. Microsoft Press, 2nd ed., 2002.

[30] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase Authentication Protocol for Wireless Sensor Networks in Distributed IoT Applications," *IEEE WCNC*, no. Jan., 2014.

A.6 [F] Establishing Dynamic Secure Sessions for ECQV Implicit Certificates in Embedded Systems

F. Basic, C. Steger and R. Kofler, “Establishing Dynamic Secure Sessions for ECQV Implicit Certificates in Embedded Systems,” in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1-6, 2023.

Abstract. Be it in the IoT or automotive domain, implicit certificates are gaining ever more prominence in constrained embedded devices. They present a resource-efficient security solution against common threat concerns. The computational requirements are not the main issue anymore. The focus is now placed on determining a good balance between the provided security level and the derived threat model. A security aspect that often gets overlooked is the establishment of secure communication sessions, as most design solutions are based only on the use of static key derivation, and therefore, lack the perfect forward secrecy. This leaves the transmitted data open for potential future exposures by having keys tied to the certificates rather than the communication sessions. We aim to patch this gap, by presenting a design that utilizes the Station to Station (STS) protocol with implicit certificates. In addition, we propose potential protocol optimization implementation steps and run a comprehensive study on the performance and security level between the proposed design and the state-of-the-art key derivation protocols. In our comparative study, we show that with a slight computational increase of 20% compared to a static ECDSA key derivation, we are able to mitigate many session-related security vulnerabilities that would otherwise remain open.

My Contribution. My contribution to this paper was manyfold. Firstly, in realizing the main concepts and design behind the proposed dynamic key derivation protocol. I research and compared the current SotA on the ECQV scheme and the use of key derivation protocols, as well as the use of this type of protocol in other networks and scheme. Based on the findings, I realized a novel protocol that uses the established and proven STS protocol with ECQV, benefiting from the implicit certificate design. I also implemented and tested the proposed protocol together with three other SotA ECQV key derivation protocols on different devices and evaluated their performance. To support the idea of using a dynamic key derivation protocol, I analyzed and defined an optimization design for possible parallel execution of related operations. The new optimized protocol derivation was also included in the evaluation analysis. Finally, to showcase the use of the provided protocols on the real automotive system, I implemented them on top of the already previously realized BMS security architecture. Furthermore, I implemented a CAN-FD communication with a hypothetical EVCC and tested it on its performance against a static key derivation protocol. To summarize the evaluation, I also provided a security analysis of the proposed protocol in relation to the other SotA protocols.

Copyright © 2023 IEEE. **DOI**: 10.23919/DATE56975.2023.10137025
Reprinted, with permission, from https://ieeexplore.ieee.org/document/10137025

Establishing Dynamic Secure Sessions for ECQV Implicit Certificates in Embedded Systems

Fikret Basic, Christian Steger
Institute of Technical Informatics
Graz University of Technology
Graz, Austria
{basic, steger}@tugraz.at

Robert Kofler
R&D Battery Management Systems
NXP Semiconductors Austria GmbH Co & KG
Gratkorn, Austria
robert.kofler@nxp.com

***Abstract*—Be it in the IoT or automotive domain, implicit certificates are gaining ever more prominence in constrained embedded devices. They present a resource-efficient security solution against common threat concerns. The computational requirements are not the main issue anymore. The focus is now placed on determining a good balance between the provided security level and the derived threat model. A security aspect that often gets overlooked is the establishment of secure communication sessions, as most design solutions are based only on the use of static key derivation, and therefore, lack the perfect forward secrecy. This leaves the transmitted data open for potential future exposures by having keys tied to the certificates rather than the communication sessions. We aim to patch this gap, by presenting a design that utilizes the Station to Station (STS) protocol with implicit certificates. In addition, we propose potential protocol optimization implementation steps and run a comprehensive study on the performance and security level between the proposed design and the state-of-the-art key derivation protocols. In our comparative study, we show that with a slight computational increase of 20% compared to a static ECDSA key derivation, we are able to mitigate many session-related security vulnerabilities that would otherwise remain open.**

***Index Terms*—ecqv, implicit, certificate, sts, dynamic, session, key derivation, embedded, security, constrained, automotive.**

I. Introduction

Security is becoming increasingly important in protecting the ever-expanding connections of modern embedded devices. The use of common schemes, e.g., Transport Layer Security (TLS), often proves to be difficult due to the constrained nature of the used devices, which can only allow for a limited performance overhead [1]. In contrast, implicit certificates are showing promise in replacing the traditional security architecture schemes. Implicit certificates offer a lightweight certificate format, and a flexible public key derivation and authentication mechanism that make the use of public key infrastructures more accessible for constrained embedded systems [2]–[6].

Different schemes exist based on the implicit certificates, with Elliptic Curve Qu-Vanstone (ECQV) still being the most popular and researched one [7]. While there has been numerous research done on ECQV and its use with embedded systems [2]–[6], [8]–[10], we noticed that certain security aspects are left out when considering the session key derivation process. The key derivation (KD) and session establishment solutions often neglect a very important key aspect, the *perfect forward secrecy*, specifically, the ephemeral key security characteristic. Forward secrecy allows for a dynamic KD and it considers the state where each newly derived key has a high-enough entropy and is independent of a previous one [11]. This is especially important in session communication, where interactions happen on a frequent basis. We believe that is characteristic often gets neglected due to a believed premise of the necessity for sacrificing the security strength for the performance gain with the limited embedded devices. Rather, what often gets deployed is a static KD where key computations are directly linked to their certificate material. These keys would, hence, only be changed by the change of the certificates and through re-initiating the authentication and session establishment steps. It is, therefore, called a static key exchange, since no other KD function or additional input data is used to mask the present session key which is fully dependent on the current certificate. This can be very problematic in situations where, implementation-wise, either due to the limitations in the system's architecture, constrained nature of the devices, or neglect from the developers, can lead to longer than the intended use of the same session key.

Regular key updates are important, as in unfortunate cases, where the session key might get compromised, e.g., via the node capturing attack by compromising a valid device that holds it, all the captured exchanged messages would also be able to be decrypted. Any attack that can compromise the stored device credentials would be able to exploit the statically derived keys. An especially dangerous attack, which is also prevalent in TLS, is the key compromise impersonation (KCI). It is a man-in-the-middle (MitM) attack where an attacker can impersonate the trusted server side to manipulate the key derivation process [12]. In 2018, OWASP rated for internet of things (IoT) weak, guessable and hard-coded passwords as the number one weakness for the IoT systems, which also considers the key credentials [13]. In fact, based on the study by the SEC Consult between 2015 and 2016, the number of exposed private keys by IoT devices grew by 40% [14]. The ENISA initiative, targeted at investigating automotive security vulnerabilities, listed remote attacks, theft and surveillance as one of the most potent attacks that can happen due to the lack of the required cryptographic functionality support. In their document, all three attacks are affiliated with the lack of forward secrecy for both the wide and local networks [15].

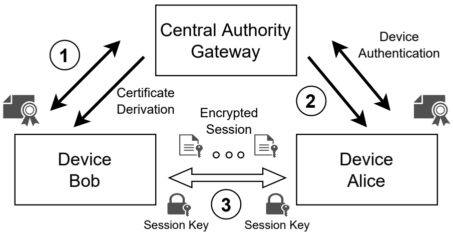


Fig. 1. Centralized implicit certificate architecture.

To mitigate these security vulnerabilities, we focus on providing a solution that is independent of the rate of the certificate updates, and which ensures that each new communication session would always yield a new key derivation. Additionally, we want to make sure that a session key compromise does not lead to exposure of previous or further keys, i.e., to guarantee perfect forward secrecy. To fulfil these constraints, we present a design based on the Station-to-Station (STS) protocol [11] for a dynamic KD for implicit certificate schemes and extend on the general lightweight ECQV implementation by Pollicino et al. [2]. Furthermore, we investigate the optimization steps for the STS KD protocol execution for the implicit certificates, analyze its applicability for the embedded hardware by implementing and evaluating it on different devices, and compare it with other related implicit certificate schemes. Summarized, our **main contributions** contained within this work are:

1) Design and implementation of a dynamic key derivation approach for implicit certificate architecture schemes using the STS protocol.
2) Performance and security evaluation of state-of-the-art (SotA) KD implicit certificate schemes by expanding on the existing work from the automotive and IoT domains.
3) Testing the protocol's feasibility in an automotive system by implementing it on top of a battery management system (BMS) to depict a real-world scenario.

II. Background on the Security Architecture

We consider three main stages when deploying implicit certificates in a network, as shown in Figure 1 [5], [8]: (1) device authentication and deployment, (2) certificate derivation, and (3) session establishment. The deployment phase primarily depends on the main system architecture, however, it generally contains a central, and a more powerful, certificate authority (CA) device. The certificate derivation phase is straightforward with ECQV and almost identical among different solutions [3], [5], [6], [8]. The session establishment process often differs and depends on the KD and node authentication algorithms.

A. Key derivation for secure sessions

We differentiate between two sessions, the certificate session and the communication session. The certificate session considers the validity duration of the currently issued certificates, e.g., in a vehicle during each new engine start, while the communication session considers the duration during one message exchange between two or multiple devices, e.g., monitoring, updates, status readout, etc.

We refer to static key derivation (SKD) as the calculation approach that relies on the traditional Diffie-Hellman KD, i.e., where the keys or the underlying secret are derived from the multiplication of the stored private key and the other device's public key as $S_k = Prk_a * Puk_b = Prk_b * Puk_a$. The SKD secret is tied to its current certificate session rather than the communication session. As long as the private and public key pairs are not updated, the underlying session key will also not change. Contrarily, the dynamic key derivation (DKD), as the one presented in this work, fulfils the condition that a new session key is derived on each new communication session start, regardless of the current certificate session. The DKD makes sure that each communication session remains independent from the other sessions and should, ideally, provide the perfect forward secrecy attribute. A key derived via this method is also known as the ephemeral secret key.

III. Related Work

Several research works have already been published on the use of the ECQV and the session KD, both under the general and embedded environments. Porambage et al. [3], [9] present one of the earlier session authentication and key exchange solutions for the wireless networks, where the communication between the nodes is done using an SKD. For authentication, the protocol uses Message Authentication Code (MAC) with pre-embedded keys, but it also requires that each node possesses from each other the authentication key. A different authentication scheme is presented by Siddhartha et al. [6], where an "authenticator" is used. It is made out of certificate-related data and signed by the CA. A hash function is also used for the additional integrity check. The session key calculation, however, is still based on the standard SKD.

D. Lee and I. Lee [16] present two approaches to KD in a constrained IoT environment. The first approach is based on the pure ECQV methodology with no additional authentication steps. It relies on validating the identification (ID) and correctness of the certificate calculation, but this does not guarantee the authenticity of the device itself. The certificates and the ID could be spoofed, resulting in a false identification by a malicious actor. Additionally, similar to the work presented by Sciancalepore et al. [4], the KD uses additional nonces to diversify the key. However, this does not add additional protection since the underlying secret is still calculated using an SKD, i.e., it only considers the multiplication of the private and public keys. The nonces used in the KD can be read from a monitored network. Their second method does provide DKD and ephemeral keys, nonetheless, both methods suffer from a central problem, and that is that the device authentication is not considered, rather only the public key validity.

Recently, Zi-Yuan Liu et al. [10] presented an extension of the ECQV, where devices might house multiple certificates and keys. While novel, the challenges presented in the paper are currently not relevant for this work's use cases, as the focus is placed on larger dynamic networks.

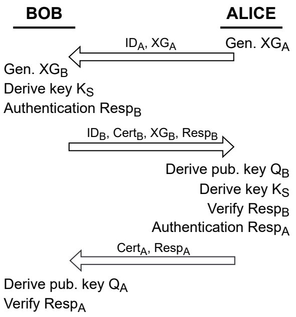


Fig. 2. Key derivation using STS protocol for ECQV architectures.

IV. A Novel Dynamic Key Derivation for ECQV

A. Security requirements

For the security requirements, it is intended to provide a design that can answer to the following threats: (T1) past data exposure, (T2) MitM attacks, (T3) node capturing attacks, (T4) key data reuse for further session calculations, (T5) key derivation exploitation; each unique key needs to have a high-enough entropy, and that is only stored, and being able to be stored, by the valid parties. We aim to protect two important system assets: session data, and security credentials. The design also needs to be lightweight in its implementation so as to be easily accessible for the embedded devices.

B. Protocol formalization

We base our design of the DKD on the use of the STS protocol [11], [17]. STS is a known protocol used in wide networks; however, it has not been previously investigated for use with the ECQV. The STS derivation should consider the ECQV implicit certificate calculation properties. The protocol steps are shown in Figure 2. It is assumed that the first two phases are correctly done as explained in Section II.

The protocol uses the implicit certificate with the elliptic curve digital signature algorithm (ECDSA) to provide authentication as shown with Algorithm 1, and verification with Algorithm 2. What makes it unique compared to other STS algorithm derivations, is that ECQV relies on the implicit derivation of the public key for the signature verification. The security of the ECDSA algorithm with the ECQV scheme has been proven secure against passive attacks [18]. The public key calculation used for verification is derived as:

$$Q_X = Hash(Cert_X) * Decode(Cert_X) + Q_{CA} \quad (1)$$

The STS provides ephemeral keys, by always deriving a new random elliptic curve (EC) point in the request as:

$$X \in_R [1, ..., n-1] \rightarrow XG = X * G \quad (2)$$

Derivation of session K_S keys is done by calculating:

$$K_{PM} = X_A * XG_B = X_B * XG_A \quad (3)$$

$$K_S = KDF(K_{PM}, salt) \quad (4)$$

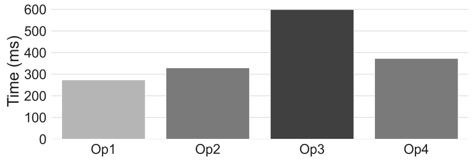


Fig. 3. Time duration of individual STS operation runs on an STM32F676.

Algorithm 1: STS implicit certificate auth. response.

```
Input: $XG_A$, $XG_B$, $K_S$
Output: $Resp$
if $device_A$ then
    $d_{sign} \leftarrow sign(Prk_A, (XG_A || XG_B))$
else
    $d_{sign} \leftarrow sign(Prk_B, (XG_B || XG_A))$
end
$Resp \leftarrow encrypt(K_S, d_{sign})$
return $Resp$
```

Algorithm 2: STS implicit certificate sign. verification.

```
Input: $Resp_X$, $Cert_X$
Output: $Status_{Ok}$, $Status_{Err}$
$d_{signX} \leftarrow decrypt(K_S, Resp_X)$
$Q_X \leftarrow hash(Cert_X) * decode(Cert_X) + Q_{CA}$
$Status \leftarrow verify(Q_X, d_{signX})$
return $Status$
```

C. STS protocol optimization

Even though the STS protocol might provide more security advantages compared to related KD implicit certificate protocols (see Section V-D), the main drawback is in its timely execution. As this is still an important aspect of modern constrained systems, we investigate potential optimizations. We divide the entire STS ECQV protocol into four operations:

- Op1 - Request phase; random XG point derivation
- Op2 - Public key and premaster session key generations
- Op3 - Auth. signature derivation and encryption
- Op4 - Auth. signature decryption and verification

In this analysis, we do not consider the transfer time. We derive two potential optimizations. Similar to the work presented by Sciancalepore et al. [4], the initial request can be made to contain both the certificate and the XG data, with the calculations of the public key and premaster secret data (see Op2) being done in parallel. Further optimization could be to also include the following Op3 to be executed parallel after Op2 as well. There is a drawback here, and that comes at the expense of the algorithm's flexibility. Failed authentication requests would only be checked after the calculations have been processed. This could open some doors for misuse by malicious users, either through denial-of-service, or similar attacks. But the actual implementation does not suffer in terms of general security since the calculations are still processed in the same manner. The main advantage would be from the system design perspective, which would allow additional operations to run in parallel. The sent data is identical to the original protocol, but the message and content order vary slightly. Figure 3. shows individual operation time requirements.

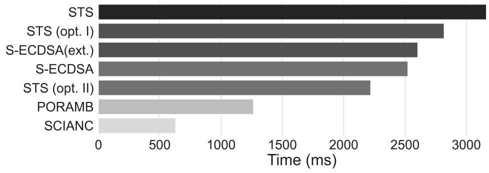


Fig. 4. Comparison of the total KD protocols processing time.

The total execution time with the conventional STS between two devices can be represented as:

$$\tau_T = \sum_{i=1}^{N_{Op}} T_{OpAi} + \sum_{i=1}^{N_{Op}} T_{OpBi} \;\text{,with } N_{Op} = 4 \quad (5)$$

As the optimization can be applied through the Op2 and Op3, we get the following derivation based on the time that each device takes to calculate the operations:

$$\forall x \in \{2,3\},\, T_{OpAx} = \begin{cases} 0, & \text{if } A = B \\ |T_{OpAx} - T_{OpBx}|, & \text{otherwise} \end{cases} \quad (6)$$

This means that no additional time is taken per device A (or B, as it is symmetrical) if they are identical, or if they are not, the extra amount of time depends on the difference in their execution time for Op2 and Op3.

If the devices are equal, ideally, optimization formulas for two different steps of optimizations based on the system requirements would bring the total run times to:

$$\text{Opt. I} \quad \tau_T^{'} = 2 * T_{Op1} + T_{Op2} + 2 * T_{Op3} + 2 * T_{Op4} \quad (7)$$

$$\text{Opt. II} \quad \tau_T^{''} = 2 * T_{Op1} + T_{Op2} + T_{Op3} + 2 * T_{Op4} \quad (8)$$

The primary advantage of the optimization is the clear reduction in the total execution time by maintaining a minimal change to the original STS protocol structure. In Section V-A, we compare different protocols for the implicit certificate KD and show the difference in time execution between the optimized and non-optimized STS on real embedded hardware.

V. Implementation and Evaluation

A. Protocol performance evaluation

To show the feasibility of the proposed STS protocol derivation in modern systems and compare it with other SotA KD protocols for implicit certificates, we implement and run the protocols under different embedded devices. We analyze the runs under three main hardware performance level groups:

- Low-end: Arduino, ATmega2560, 8-bit 16MHz
- Mid-tier: S32K144, ARM Cortex-M4F 32-bit 80MHz; and STM32F767, Cortex-M7 32-bit 216MHz
- High-end: Raspberry Pi 4, Cortex-A72 64-bit 1.5GHz

The implementations are done in *C* and make use of the functions provided by the micro-ecc, tiny-aes, and bear-ssl libraries, as well as the micro ECQV functions provided by Pollicino et al. [2]. All protocols have been tested with the secp256r1 256-bit EC, with 256-bit level for the SHA and HMAC, and 128-bits for the AES and CMAC.

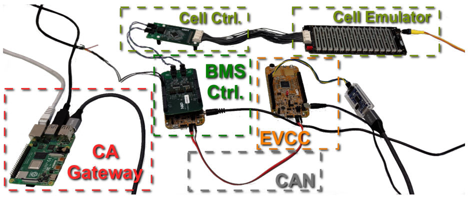


Fig. 5. Test suite for the ECQV and KD protocol evaluation.

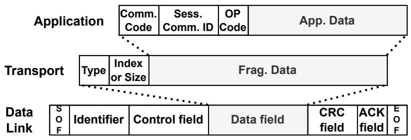


Fig. 6. CAN-FD network layers used for the session test communication.

In total, we test four different protocols derived from two groups based on the use of the authentication mechanism, i.e., on those that rely on the use of ECDSA: (i) static ECDSA by Basic et al. [5] as S-ECDSA, and (ii) STS from this work, and those that only use the symmetric cryptography authentication without the EC operations: (iii) from Porambage et al. [3] as PORAMB, and (iv) from Sciancalepore et al. [4] as SCIANC. We also consider the extension of the S-ECDSA protocol, specifically the additional authentication of the ack acknowledgement messages, based on the finished message handling as seen from Porambage et al. [3]. Furthermore, we also evaluate the STS protocol when considering the optimization steps explained in Section IV-C. Only STS is the true DKD, while the rest fall into the SKD category. The results of the evaluation are shown in Table I, with Figure 4. showing the graphical representation for the STM32F767. The times are averaged after ten runs. The measurements were done using system ticks and Nordic PPK2. The run time scalability is relatively consistent regarding the devices' performances. While STS shows the highest execution time, its optimization variants show the potential time similar to or faster than the S-ECDSA. The PORAMB and SCIANC show the fastest time as they use a different authentication mechanism and do not rely on the EC operations. However, these protocols lack some of the necessary security options as discussed in Section V-D.

B. Overhead examination

To give a clearer analysis of the algorithm processing time, it would be advantageous to consider the transmission overhead, however, that parameter is heavily dependent on the used communication protocol and its configuration. Here, we provide an overview of the overhead for each algorithm during the KD exchange protocol, independent of the communication technology in use. We consider only the protocol-affiliated transmission data on the application level. Security algorithms bit sizes are the same as the ones used in Sect. V-A. We assume IDs to be of 16 bytes and use the minimal certificate encoding with 101 total bytes [7]. The results are shown in Table II.

Both the S-ECDSA and STS protocols showed similar transmission sizes, with also the least communication steps when

TABLE I
EXECUTION TIME IN MILLISECONDS OF THE KD PROTOCOLS FOR ECQV FOR THE RESPECTIVE EMBEDDED HARDWARE.

Protocol / Device	ATMega2560	S32K144	STM32F767	RaspberryPi 4
S-ECDSA	36859.26 ± 0.18	2894.1 ± 9.83	2521.77 ± 5.87	18.76 ± 0.11
S-ECDSA (ext.)	36882.64 ± 0.23	2976.2 ± 11.56	2602.69 ± 8.61	18.68 ± 0.12
STS	46262.03 ± 0.13	3622.71 ± 7.034	3162.07 ± 7.52	23.26 ± 0.12
STS (opt. I)	41680.23 ± 1.2	3246.55 ± 12.97	2818.02 ± 11.26	20.87 ± 0.07
STS (opt. II)	32410.81 ± 1.14	2556.84 ± 13.13	2219.25 ± 11.3	16.31 ± 0.07
SCIANC	8990.49 ± 0.03	721.67 ± 0.28	628.1 ± 0.32	4.58 ± 0.02
PORAMB	17932.17 ± 0.05	1471.66 ± 0.63	1263.0 ± 0.42	8.98 ± 0.04

TABLE II
COMMUNICATION STEPS AND TRANSMISSION OVERHEAD OF THE KD PROTOCOLS FOR ECQV.

Protocol	S-ECDSA(+ext.)	STS	SCIANC	PORAMB
Step: Op. (X bytes)	A1: ID(16), Nonce(32)	A1: ID(16), XG(64)	A1: ID(16), Nonce(32), Cert(101)	A1: Hello(32), ID(16)
	B1: ID(16), Cert(101), Sign(64), Nonce(32)	B1: ID(16), Cert(101), XG(64), Resp(64)	B1: ID(16), Nonce(32), Cert(101)	B1: Hello(32), ID(16)
	A2: Cert(101), Sign(64)	A2: Cert(101), Resp(64)	A2: Auth_MAC(32)	A2: Cert(101), Nonce(32), MAC(32)
	B2: ACK(1), (+Ext_Fin(96))	B2: ACK(1)	B2: Auth_MAC(32)	B2: Cert(101), Nonce(32), MAC(32)
	A3: (+Ext_Fin(96))			A3 & B3: Finish(197)
Total	4(+1): 427(+192) B	4: 491 B	4: 362 B	6: 820 B

not considering the last ack message. The SCIANC protocol also requires only four transmissions, but with only 362 total bytes under the assumed setup. Contrarily, the PORAMB algorithm showed the largest overhead, with 6 total steps and 820 bytes. We did not include the optimized version of STS since it does not differ in terms of the transmitted data. Considering the fast data rates of most communication protocols and the presented data sizes, we can conclude that the influence of the transmission overhead would be minimal in comparison to the individual KD protocols. This is further complemented by the prototype evaluation results from Section V-C.

C. Prototype implementation evaluation

In order to evaluate the proposed protocol design on its technical use, we implemented a prototype system that depicts a common communication occurrence between two ECUs in an automotive network. It handles the secure communication between a BMS controller, and an electric vehicle charging controller (EVCC) [19]. Both devices are represented with an S32K144 microcontroller from the NXP Semiconductors to portray a real-world environment. The BMS is additionally connected to a battery cell controller and a battery emulator for emulating a functional unit. The setup is shown in Figure 5.

The session communication between the devices takes place over a Controller Area Network (CAN) interface. The test suite uses the CAN-FD derivation with an implemented CAN-TP layer for message fragmentation [20]. Figure 6 shows the message formats. The devices also communicate with a more powerful CA gateway (represented with a Raspberry Pi 4) to handle the initial device authentication and certificate distribution. The nominal phase CAN-FD bit rate was configured at 0.5 Mbit/s, with the data phase rate being set at 2 Mbit/s.

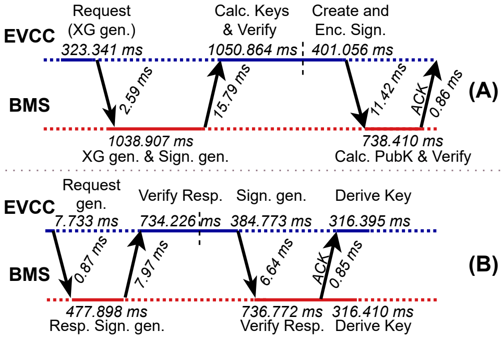


Fig. 7. Timeline model of the prototype session communication between a BMS and EVCC for: (A) STS & (B) S-ECDSA, ECQV KD protocols.

For the evaluation, we compare the proposed STS implementation against the common static ECDSA [2], [5]. For a fair comparison, as to account for the conventional deployment of these protocols in the field, we did not consider the optimization handling for the parallel operation runs argued in Section IV-C. The implemented security protocols use the same library sources as those mentioned in Section V-A. The timeline of both protocols is shown in Figure 7. The

TABLE III
SECURITY OVERVIEW OF THE KD PROTOCOLS FOR ECQV.

	S-ECDSA	STS	SCIANC	PORAMB
Data exposure	X	✓	X	X
Node capturing	Δ	Δ	X	X
Key data reuse	X	✓	Δ	X
Key der. exploit	Δ	✓	Δ	Δ
Auth. procedure	✓	✓	Δ	Δ

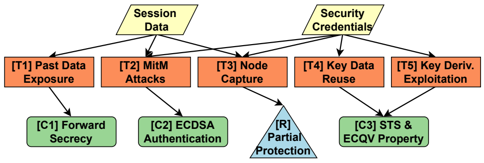


Fig. 8. Block diagram representation for the STS-ECQV KD threat model.

STS implementation showed only a slight difference in the total run time with $3.257\,s$ compared to S-ECDSA's $2.677\,s$, i.e., an increase of $21.67\,\%$. The CAN-FD transfer time over the physical link was negligible ($< 1\,ms$). The majority of the communication time from Figure 7. was for the data processing on the remaining layers.

D. Security analysis

We concern ourselves with the listed threats from Section IV-A and compare the previous KD algorithms on the provided security level. We also specially look at the mutual authentication procedure, as an important feature against MitM attacks. The analysis is presented in Table III, with the following notation: X - weak or no countermeasure, Δ - partial protection, ✓ - fully protected.

The lack of forward secrecy for all protocols, except STS, makes them highly vulnerable to previous session data exposure, key material reuse (while having the same certificates), and node-capture attacks. However, we note that no algorithm is fully protected against the node-capture attacks, as even with STS, the protection can only be guaranteed for the previous messages, not the future ones. The mutual authentication for both SCIANC and PORAMB is based on symmetric cryptography with some concerns. PORAMB has the requirement to store individual keys per the number of devices, which makes future updates troublesome. SCIANC algorithm ties its session key with the KD authentication, meaning that if the session key gets exploited so will the future authentication. On the other hand, with S-ECDSA and STS, the authentication is based on the ECDSA with private keys used for signature derivation. Figure 8. shows the derived countermeasures on the listed threats for the STS-ECQV KD.

VI. CONCLUSION

In this work, we have presented a key derivation and session establishment model using the STS protocol within the ECQV implicit certificate framework, and its relation and comparison with other KD protocols on embedded devices. While requiring more time, the STS offers a good balance between providing additional security features and certainty without compromising much of the performance. It showed a slight run time increase of $\approx 21\%$ compared to a static ECDSA KD protocol, with no additional communication overhead. While other non-EC authentication-based KD protocols showed a noticeable faster execution time, they also lacked the security level acceptable for modern systems. To compensate for the STS run time, we introduced a series of optimization steps for the protocol operations. For future work, we plan to investigate the influence of security modules and hardware accelerators when considering the implicit certificate protocols on embedded devices, especially those related to session establishment.

ACKNOWLEDGMENT

This project has received funding from the "EFREtop: Securely Applied Machine Learning - Battery Management Systems" (Acronym "SEAMAL BMS", FFG Nr. 880564).

REFERENCES

[1] J. P. Hughes and W. Diffie, "The Challenges of IoT, TLS, and Random Number Generators in the Real World: Bad Random Numbers Are Still with Us and Are Proliferating in Modern Systems.," *Queue*, jun 2022.
[2] F. Pollicino *et al.*, "An experimental analysis of ECQV implicit certificates performance in VANETs," in *IEEE 92nd VTC2020-Fall*, 2020.
[3] P. Porambage *et al.*, "Two-phase authentication protocol for wireless sensor networks in distributed iot applications," in *IEEE WCNC*, 2014.
[4] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public Key Authentication and Key Agreement in IoT Devices With Minimal Airtime Consumption," *IEEE Embedded Systems Letters*, vol. 9, 2017.
[5] F. Basic *et al.*, "Trust your BMS: Designing a Lightweight Authentication Architecture for Industrial Networks," in *23rd IEEE ICIT*, 2022.
[6] V. Siddhartha, G. S. Gaba, and L. Kansal, "A Lightweight Authentication Protocol using Implicit Certificates for Securing IoT Systems," *Procedia Computer Science*, vol. 167, pp. 85–96, 2020.
[7] M. Campagna, *Standards for Efficient Cryptography 4 (SEC4): Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)*. Certicom Corp., 2013.
[8] D. Puellen *et al.*, "Using Implicit Certification to Efficiently Establish Authenticated Group Keys for In-Vehicle Networks," in *Proc. of the 11th IEEE VNC Conf.*, 2019.
[9] P. Porambage *et al.*, "Certificate-Based Pairwise Key Establishment Protocol for Wireless Sensor Networks," in *16th IEEE CSE*, 2013.
[10] Z.-Y. Liu *et al.*, "Extension of Elliptic Curve Qu–Vanstone Certificates and Their Applications," *J. Inf. Secur. Appl.*, vol. 67, jun 2022.
[11] W. Diffie *et al.*, "Authentication and Authenticated Key Exchanges," *Design, Codes and Cryptography*, p. 107–125, June 1992.
[12] C. Hlauschek *et al.*, "Prying Open Pandora's Box: KCI Attacks against TLS," in *9th USENIX WOOT*, (Washington, D.C.), Aug. 2015.
[13] OWASP, "OWASP IoT Top 10." https://owasp.org, 2018.
[14] S. Consult, "House Of Keys: 9 Months Later... 40% Worse." https://sec-consult.com/blog/detail/house-of-keys-9-months-later-40-worse/, 2016. Accessed: 05.09.2022.
[15] ENISA, *Cyber Security and Resilience of smart cars – Good practices and recommendations*. ENISA EU, 2016.
[16] D.-H. Lee and I.-Y. Lee, "A Lightweight Authentication and Key Agreement Schemes for IoT Environments," *Sensors*, vol. 20, 2020.
[17] F. Basic, C. Steger, and R. Kofler, "Poster: Establishing Dynamic Secure Sessions for Intra-Vehicle Communication Using Implicit Certificates," in *ACM EWSN 2022 Conf.*, oct 2022. poster session.
[18] D. R. L. Brown *et al.*, "Security of ECQV-Certified ECDSA Against Passive Adversaries." Cryptology ePrint Archive, Paper 2009/620, 2009.
[19] A. Fuchs *et al.*, "Securing Electric Vehicle Charging Systems Through Component Binding," in *Computer Safety, Reliability, and Security*, pp. 387–401, 2020.
[20] ISO 15765-2:2016, "Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 2: Transport protocol and network layer services," Standard, ISO, 2016.

A.7 [G] Wireless BMS Architecture for Secure Readout in Vehicle and Second life Applications

F. Basic, C. R. Laube, P. Stratznig, C. Steger, and R. Kofler, “Wireless BMS Architecture for Secure Readout in Vehicle and Second life Applications,” in *8th IEEE International Conference on Smart and Sustainable Technologies*, IEEE, 2023.

Abstract. Battery management systems (BMS) are becoming increasingly important in the modern age, where clean energy awareness is getting more prominent. They are responsible for controlling large battery packs in modern electric vehicles. However, conventional solutions rely only on a wired design, which adds manufacturing cost and complexity. Recent research has considered wireless solutions for the BMS. However, it is still challenging to develop a solution that considers both the active in-vehicle and the external second-life applications. The battery passport initiative aims to keep track of the batteries, both during active and inactive use cases. There is a need to provide a secure design while considering energy and cost-efficient solutions. We aim to fill this gap by proposing a wireless solution based on near-field communication (NFC) that extends previous work and provides a unified architecture for both use cases. To provide protection against common wireless threats, an advanced security analysis is performed, as well as a system design analysis for the wake-up process that reduces the daily power consumption of the stored battery packs from milli- to microwatts.

My Contribution. I contributed to the publication by realizing a unified design from the previous publications concerning BMS and NFC use cases. I also contributed to the majority of the paper’s text. Claudia Laube contributed to the paper by finding potential security pitfalls and extending the implementation design. Patrick Stratznig supported in realizing the wake-up concepts, implementing and testing them. Christian Steger provided guidance in establishing a relational link between the current and previous papers, introduction and challenges summarization, and guidance in analysis presentation in relation to the achieved results. To further support the evaluation section of the publication, I also provided a formal security analysis using the BAN logic model on the proposed security protocol for the BMS security readout using NFC.

DOI: 10.23919/SpliTech58164.2023.10193573
Reprinted, with permission, from https://ieeexplore.ieee.org/document/10193573

Wireless BMS Architecture for Secure Readout in Vehicle and Second life Applications

Fikret Basic, Claudia Rosina Laube, Patrick Stratznig, Christian Steger
Institute of Technical Informatics
Graz University of Technology
Graz, Austria
{basic, steger}@tugraz.at

Robert Kofler
R&D Battery Management Systems
NXP Semiconductors Austria GmbH Co & KG
Gratkorn, Austria
robert.kofler@nxp.com

Abstract—**Battery management systems (BMS) are becoming increasingly important in the modern age, where clean energy awareness is getting more prominent. They are responsible for controlling large battery packs in modern electric vehicles. However, conventional solutions rely only on a wired design, which adds manufacturing cost and complexity. Recent research has considered wireless solutions for the BMS. However, it is still challenging to develop a solution that considers both the active in-vehicle and the external second-life applications. The battery passport initiative aims to keep track of the batteries, both during active and inactive use cases. There is a need to provide a secure design while considering energy and cost-efficient solutions. We aim to fill this gap by proposing a wireless solution based on near-field communication (NFC) that extends previous work and provides a unified architecture for both use cases. To provide protection against common wireless threats, an advanced security analysis is performed, as well as a system design analysis for the wake-up process that reduces the daily power consumption of the stored battery packs from milli- to microwatts.**

Index Terms—**Battery Management System, Wireless, Security, Cyber-physical, RFID, NFC, Second life, Vehicle, Battery.**

I. Introduction

Battery management systems (BMS) represent one of the most important building blocks of modern electric vehicles (EV). They are responsible for ensuring safe and reliable use of large battery packs [1]. Since they are one the main driving forces of an EV, any problem that occurs with the batteries can directly affect the entire vehicle and also the driver's safety, leading to various hazards such as thermal runaway [2]. Modern BMS are deployed in several topologies where different modules are responsible for tasks ranging from battery cell sensor data acquisition to their transmission.

One challenge with modern EV battery systems is the reuse of used batteries at the end of their life cycle. Even if the batteries no longer meet the needs of a vehicle, they may be able to be used for other applications. This battery state is referred to as second life [3]. The battery's state of health (SoH), state of charge (SoC), and other diagnostic information must be tracked during its lifetime [4]. A BMS controller would track this information in an active EV use case through its internal communications, but if the batteries are stored in a warehouse, for example, the battery packs may only be accessed through an interface with a battery pack controller (BPC) using an external reader. BMS communication services can therefore be viewed from two angles: (i) internal - for communication between different sensors and bridge modules, and (ii) external - for diagnostic and monitoring purposes read from outside the system. There are several legislations underway aimed to introduce battery tracking via battery passports and even distributed monitoring in the cloud using electronic passports [5], [6]. It is important to find an affordable system design that allows flexible and fast readout with respect to both communication approaches. However, currently, there is no unified design that considers different BMS topologies.

Another challenge that is common with BMS is the use of cables for communication. The use of cables is associated with higher assembly cost, weight, complexity, limited scalability, and maintenance [7]–[9]. However, they also provide fast and reliable transmission between modules, which is important for systems such as BMS. An alternative would be to use wireless technologies, but here it is important to assess which technology should be used. There is currently no clear answer to this, with several works proposed ranging from the use of BLE [8], [10], Wi-Fi [11], [12], ZigBee [13], and other technologies [7]. Using multiple technologies in one system is not practical and can lead to increased costs and radio interference. Our goal is to present a design that unifies different use cases under the same wireless technology. We aim to achieve this through the use of radio-frequency identification (RFID), specifically accessible and low-cost near-field communication (NFC).

The BMS and batteries are vital components in modern EVs that also require adequate protection. Any vulnerable system component should be authenticatable and provide protection against tampering and even eavesdropping to protect user privacy [14]–[16]. NFC offers advantages from a security perspective because it enables short-range communication that is difficult to be exploited remotely. However, there are still many vulnerabilities in BMS and NFC that need to be addressed to provide a fully secure design [16]–[18].

In response to the aforementioned challenges, we present in this paper a complete design for wireless BMS internal and external data communication based on NFC. To extend on the previous work in this area, we unify two separate designs, one targeting internal sensor readout from battery packs and the other targeting external status and diagnostic readout in a BMS system design [19], [20].

Contributions. We extend the security architecture by presenting and formally verifying a complete authentication and session establishment protocol for the external readout use case. We also investigate the important wake-up readout cycle for other use cases and propose two different system designs. Summarized, our main contributions are:

- A system architecture that enables wireless internal and external readout of battery sensor data for various cases.
- Proposal of two methods for the wake-up process.
- Extending the existing security solution and its proof.

II. Background and Related Work

A. Battery management systems (BMSs)

A BMS is responsible for controlling a large number of battery cells connected either in series or in parallel to ensure safe and reliable operation [1]. There are several BMS topologies, with distributed systems being the most common [21]. Here, a powerful main BMS controller is responsible for controlling and relaying information to the outside world. The controller communicates with several BPCs that are responsible for intermediate operations and data collection from a group of battery cells. We call this group of elements a battery pack.

B. Near field communication (NFC)

NFC is a short-range wireless technology defined in the ISO 18092 NFC standard, which builds on the ISO 14443 RFID standard. It operates at a radio frequency (RF) of 13.56 MHz with a range and data rate of up to 10 cm and 848 kbit/s, respectively [22]. Today, it is used for fast terminal controls, payments, and applications that benefit from energy harvesting. NFC can operate in three main modes, reader/writer mode, peer-to-peer mode, and card emulation mode [22]. In our work, we rely on the reader/writer mode for all use cases. Here, devices are divided into an active and a passive class. In the context of BMS, NFC suffers from a short range compared to other wireless technologies [7], but it provides an advantage when considering interference, security concerns and accessibility with NFC-enabled devices [19]. It is also capable of generating electromagnetic fields through an active device to power the passive devices via energy harvesting [23].

C. Second life for batteries

A common problem with the use of batteries in EVs is the problem of battery recycling. When the maximum battery capacity falls below a certain threshold (around 80%), the current batteries need to be replaced with new ones. However, the high recycling cost and pollution make this process undesirable. Instead, it is proposed to maximize the potential use of a battery by reusing it for other uses that are not constrained by the same customer criteria as with EVs [3].

Second-life battery use is an ongoing discussion within the European Union (EU). A concept of ”battery passports” is being discussed, which aims to track batteries throughout their lifetime, from manufacturing to recycling [5], [6]. The tracking and identification of batteries would be performed via QR codes. However, QR technology is limited in terms of functionality, scalability, services and security. In this work, we propose an alternative lightweight design that aims to fill these drawbacks with minimal additional overhead. The use of battery passports is also of interest with respect to the new concepts related to the BMS service extension to the cloud environment, as shown in the work of Li et al. [24], Taesic et al. [9], and Kai et al. [25]. This cloud shift provides additional power for SoC and SoH computations and enables machine learning or digital twin models, and extra security verification.

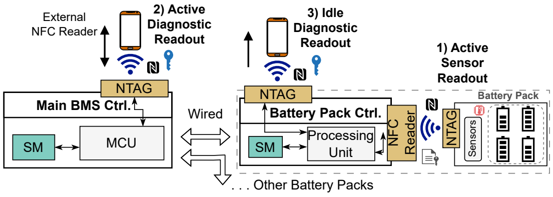


Fig. 1. Wireless NFC BMS architecture considering three different use cases, extending basis design with secure modules (SM) and NFC tags and readers.

TABLE I
Overview of the proposed use cases and deployment scenarios.

Use case	Deploy. scenario	Comm. direction	Readout data
Active sensor	internal; deployed	Batt. pack → BPC	sensor data
Idle diagnostic	external; stored	BPC → Ext. reader	status & sensor data
Active diagnostic	external; deployed	BMS ↔ Ext. reader	status & diagnostic

III. The Novel BMS Design for Wireless Readout

By relying only on NFC, we are able to provide a solution for three different scenarios that extend the BMS functionality, as described in Table I.: (i) active readout of internal sensors [19], (ii) idle state scenario considered for off-vehicle use cases, and (iii) active diagnostic readout [20].

The system architecture is shown in Figure 1. For both diagnostic readouts (active and idle), both devices are first authenticated before a secure channel is established. The data link communication layer is handled with NDEF records. The data storage depends on the reader application, either storing it offline or online in a database as proposed for battery passports [5]. Elements of the design and its implementation were handled in a recent master’s thesis [26].

Diagnostic readout for active and idle states differentiate on the devices used and the data direction. In both cases, communication takes place with an external NFC reader. The active diagnostics use case is intended for situations where BMSs are already deployed in a neighbouring system. In this case, the BMS controller is able to provide the current diagnostic data. The idle readout is intended for the storage of battery packs during their second life transfer [3]. Here,

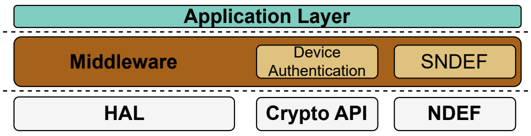


Fig. 2. Proposed wireless BMS software layer stack.

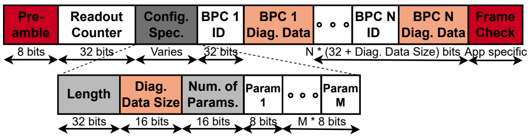


Fig. 3. Application packet structure for the active diagnostic use case.

it is considered beneficial to occasionally check the batteries' SoH to determine that no hazards have occurred. In the case of active diagnostic readout, the application data is first formatted before being transferred from the BMS host controller to the NFC reader. The proposed lightweight structure is shown in Figure 3. The BMS is expected to first collect the data from each BPC before forwarding it to the NFC reader. However, BMS topologies need to also be considered. The following considerations must be made for other topologies [21]:

- Centralized: consists of one main BMS control unit and no intermediate modules. Requires one NTAG and NFC reader. The idle diagnostic use case is considered unfeasible unless the controller is also stored.
- Modulated: multiple modules, with one main module and multiple followers. The main module enables the external interface, but each follower module also needs to include an NTAG and NFC reader for all use cases.
- Distributed: similar to the modulated design, but with a clearer separation between the central BMS controller and the battery packs. The architecture is shown in Figure 1.
- Decentralized: consists of multiple BMS subsystems. Considers linear growth of required NFC interfaces based on the topology of the underlying BMS subsystem.

Software architecture. The software architecture is generalized and is shown in Figure 2. It consists of three main layers. At the bottom is the hardware abstraction layer (HAL). It is vendor-specific and contains the driver components, e.g., for interfaces, clock, etc. The cryptographic application programming interface (API) is considered an independent entity that can be connected either directly to the HAL or to a separate security module. NDEF messages are controlled by the NFC driver. Their payload contains the components of the higher SNDEF data set located at the middle layer, i.e., the middleware. Independent of the SNDEF, the middleware is also responsible for device authentication. This means that no other access to the application layer is possible unless it has first been verified, confirmed, and processed by the middleware. Since the middle layer imposes no restrictions on the data format, the application layer can be freely defined by the developer.

IV. Security Model

To protect against common threats, a security model needs to be introduced for both internal and external communication. Several BMS threat analysis models have been created, each highlighting important vulnerabilities [9], [14], [15]. We aim not to invent a new protocol that could be susceptible to untracked attacks, but rather to adapt simple, yet effective solutions. The security model needs to answer to the current known BMS and NFC threats and fulfil the following requirements:

- Mutual device authentication
- Use of secure channels: encryption and tamper-proof
- Lightweight models with minimal performance overhead

A. Securing battery pack internal readout

Because battery packs are enclosed, any form of physical attack would directly damage the components and would therefore be infeasible or too costly for the attackers. Similarly, an external attack would also likely be difficult or impossible, although further research is needed to confirm these claims. This leaves the protection of reading the battery packs based on the proposal by Basic et al. in [19] sufficient, i.e., authentication protection based on either symmetric or asymmetric cryptography with previously embedded keys. Hence, on the device-level protection, two key strategies can be employed: (i) authentication; symmetric with pre-shared keys, or asymmetric with originality signatures, and (ii) authorization, with password-controlled read and write access.

B. Security protocol for external BMS readout

The protocol is based on symmetric, rather than asymmetric cryptography, to account for the potential performance or hardware limitations. If the master key is leaked, all past and future sessions could be compromised. Therefore, it is of utmost importance to secure the location of the master key with an SM. We rely on the use of SNDEF records [20].

The security design for external readout, which considers the idle state and active diagnostics use cases, is based on the lightweight security design for authentication and data exchange proposed in [20]. We formalize the protocol in Figure 4. and extend the solution to consider some additional security vulnerabilities that may arise from the original design. It consists of two phases, the authentication phase and the newly added session key possession confirmation phase. During the session key confirmation phase, previous messages are appended, to confirm session key ownership for both parties.

1) $N_R \rightarrow M_N : N_R, ch_r$ (1)

2) $M_N \rightarrow N_R : M_N, ch_t, \{\{M_N, ch_r\}_{K_M}\}_{K_M}$ (2)

3) $N_R \rightarrow M_N : \{\{N_R, ch_t\}_{K_M}^{-1}\}_{K_M}^{-1}$ (3)

4) $M_N \rightarrow N_R : \{M_N, X\}_{K_S}$ (4)

5) $N_R \rightarrow M_N : \{N_R, X'\}_{K_S}$ (5)

From the protocol; N_R: NFC reader id., M_N: BMS ctrl. id., ch_r: challenge nonce request from N_R, ch_t: challenge

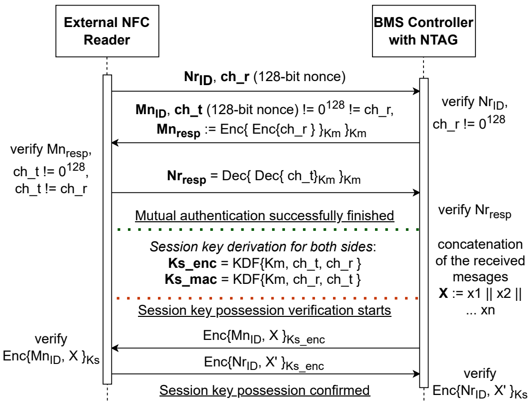

External NFC Reader
BMS Controller with NTAG
Nr_{ID}, ch_r (128-bit nonce)
verify Nr_{ID}, ch_r != 0^{128}
Mn_{ID}, ch_t (128-bit nonce) != 0^{128} != ch_r,
Mn_{resp} := Enc{ Enc{ch_r } $\}_{Km}$ $\}_{Km}$
verify Mn_{resp}, ch_t != 0^{128}, ch_t != ch_r
Nr_{resp} = Dec{ Dec{ ch_t$\}_{Km}$ $\}_{Km}$
verify Nr_{resp}
Mutual authentication successfully finished
Session key derivation for both sides:
Ks_enc = KDF{Km, ch_t, ch_r }
Ks_mac = KDF{Km, ch_r, ch_t }
concatenation of the received mesages
X := x1 || x2 || ... xn
Session key possession verification starts
Enc{Mn_{ID}, X $\}_{Ks\_enc}$
verify Enc{Mn_{ID}, X $\}_{Ks}$
Enc{Nr_{ID}, X' $\}_{Ks\_enc}$
verify Enc{Nr_{ID}, X' $\}_{Ks}$
Session key possession confirmed

Fig. 4. Security protocol sequence diagram for the external BMS readout.

TABLE II
OVERVIEW OF THE PROPOSED WAKE-UP SYSTEM DESIGN APPROACHES.

Model	Prerequisites	Pros / Cons
Event Detection (ED)	NTAG needs to have an event pin; requires a constant power source	[+] wake-up is possibly faster, [-] needs constant power source, [-] higher power consumption
Energy Harvest. (EH)	NTAG's EH needs to be specially configured; the reader has to have EH enabled	[+] NTAG is powered off in idle, [+] after wake-up, BPC can supply the NTAG, [-] wake-up takes longer

nonce request from M_N, K_M: pre-embedded master key, K_S: session key, X: concatenated previously received messages from N_R, X': concatenated received messages from M_N.

The double key encryption is used to fend off an oracle attack that could expose the vulnerabilities of the CMAC if used in the key derivation function (KDF) when based on the CBC-MAC computation. In addition, a nonce cannot be zero or equal to another. Another important point is that the underlying ciphers are not identical, i.e., in our case, BPC encrypts, while the external NFC reader decrypts the challenges, to protect against "chosen challenge oracle" attacks.

V. Wake Up Model Design

Previous works mainly investigated system design and communication with NFC. There are no clear design specifications for interaction with BMS controllers. The wake-up application is important for the idle state use case. Since a BPC is stored along with battery cells, it is not necessary to keep the controller on during the entire storage period. By controlling the wake-up of the BPC, it would be possible to minimize the total power consumption during the time the battery cells are stored and provide power during a communication period with an external NFC reader. Therefore, we want to propose a design that meets the following two main requirements:

1) Low power consumption through mode switching
2) Fast wake-up time

We propose two approaches to wake-up system design based on the different characteristics of BPC and NTAG: (i) with *event detection* (ED) and (ii) with *energy harvesting* (EH). They are evaluated based on total wake-up time and power consumption, as well as design requirements and HW/SW design considerations. Figure 5. shows the flowchart, with separate steps for both designs. Both rely on the use of I^2C for data transfer, but the difference is in how the wake-up events are triggered and how the power states are managed.

Wake-up with event detection. This method uses the event detection (ED) function on modern NTAG boards, where the ED pin acts as an event pin that can respond to different events, specifically here, the presence of an NFC field. When an RF field is detected, the ED pin is set to logic high. During idle time, the NTAG remains in the standby state and is constantly powered by the host BPC. The lowest theoretical consumption that can be achieved is if the host BPC uses a very low power state (VLPS) that can still power the NTAG and respond with the wake-up, with the NTAG being in standby mode.

Wake-up with energy harvesting. In this method, the NTAG can be completely shut down and the BPC can be put into a VLPS mode, as with the ED method. This results in lower power consumption, but may also result in a slightly longer wake-up time since the board wake-up depends on energy harvesting. The NTAG is put into EH mode, where energy is harvested from the RF field in close contact with the antenna. Since the BPC does not power the NTAG at all during the sleep phase, unlike the ED method, the NTAG must remain powered on after energy harvesting has triggered its wake-up. Thus, after the BCP wakes up and during the open communication session, it draws power from the battery cells for its normal operating mode and now also powers the NTAG.

VI. Evaluation

For the evaluation, we use the NCx33xx series of products from NXP as the NTAG and NFC Reader. They are NFC forum-compliant automotive-graded components that provide several benefiting features for testing [27]. The S32K144 microcontroller board was chosen as both the main BMS controller and the BPC. It is widely used in automotive and industrial applications, and it offers a ready BMS diagnostic application [28]. We evaluate the design on the security aspects both with informal and formal protocol analysis and on the real-device performance analysis by investigating the application overhead of the wake-up process.

A. Security informal analysis

The goal is to protect three main assets: (A1) hardware integrity, (A2) software integrity, (A3) transmitted data. To protect the integrity of (A2) and (A3), the protocol uses AES as the encryption algorithm and CMAC for tag verification. This is done to increase usability on different devices and

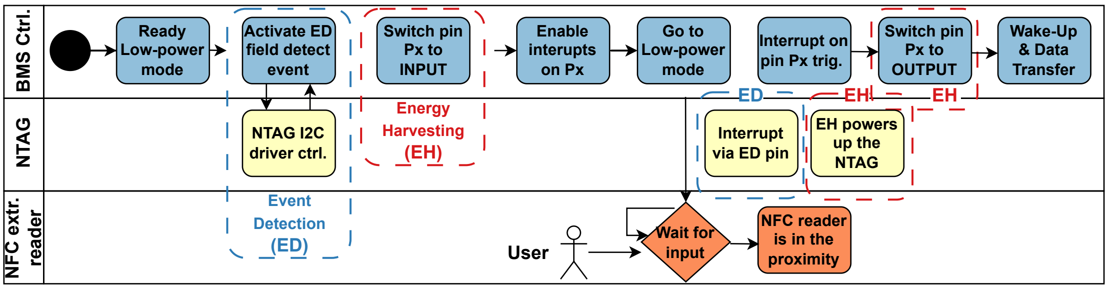


Fig. 5. Wake-up process flowchart: in dashed (ED) blocks steps are shown for the event detection, whereas in (EH) they are shown for the energy harvesting.

improve performance. The operation mode is CBC in Encrypt-then-MAC mode, as it provides higher security than other operation modes. It is important to use a different key between the AES and CMAC computation because using the same key would allow the attacker to forge the tags if they had access to an encryption oracle where they could query the values of the last CMAC computation block. Additionally, to protect against forms of replay attack, tag chaining was implemented. Tag chaining considers appending the previous tag into the buffer of the newly received message to calculate the total tag value, i.e., the MAC value, by using the following structure:

$$MAC_{buffer} := sec\_data \mid IV \mid add\_data \mid previous\_tag$$

(A1) is guaranteed by performing device authentication on the battery pack as described in Section IV-A, and by mutual authentication when communicating with an NFC reader.

B. Security formal analysis - BAN logic

A formal protocol analysis was done on the mutual authentication and session establishment protocol presented in Section IV-B. It uses the BAN logic and its postulates [29].

Idealized protocol. We use the protocol from Section IV-B:

$$1)\ all\ \ plaintext \tag{6}$$

$$2)\ M_N \rightarrow N_R : \{\{ch_r, N_R \overset{K_M}{\longleftrightarrow} M_N\}_{K_M}\}_{K_M} \tag{7}$$

$$3)\ N_R \rightarrow M_N : \{\{ch_t, N_R \overset{K_M}{\longleftrightarrow} M_N\}_{K_M}\}_{K_M} \tag{8}$$

$$4)\ M_N \rightarrow N_R : \{X, N_R \overset{K_S}{\longleftrightarrow} M_N\}_{K_S} \tag{9}$$

$$5)\ N_R \rightarrow M_N : \{X', N_R \overset{K_S}{\longleftrightarrow} M_N\}_{K_S} \tag{10}$$

Initial assumptions. The following assumptions are made. Firstly, both devices regard the sent nonces to be fresh:

$$N_R \mid\equiv \#(ch_r)\ ,\ M_N \mid\equiv \#(ch_t) \tag{11}$$

Both sides believe in the use of the shared master key:

$$N_R \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N\ ,\ M_N \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N \tag{12}$$

Goals. We want to make sure that both parties are mutually authenticated and know that the other side trusts that as well:

$$G1.1)\ \ N_R \mid\equiv M_N \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N \tag{13}$$

$$G1.2)\ \ M_N \mid\equiv N_R \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N \tag{14}$$

For the second-order goals, we want to make sure that both parties trust that the other party has the correct session key:

$$G2.1)\ \ N_R \mid\equiv M_N \mid\equiv N_R \overset{K_S}{\longleftrightarrow} M_N \tag{15}$$

$$G2.2)\ \ M_N \mid\equiv N_R \mid\equiv N_R \overset{K_S}{\longleftrightarrow} M_N \tag{16}$$

Verification. We will start first with G1.1 and G1.2 goals. To verify, we will apply the rules described in the BAN logic [29]. On Eq. 7 we first apply the *message-meaning rule*:

$$\frac{N_R \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N, N_R \triangleleft \{\{ch_r\}_{K_M}\}_{K_M}}{N_R \mid\equiv M_N \mid\sim (ch_r,\ N_R \overset{K_M}{\longleftrightarrow} M_N)} \tag{17}$$

We use the *freshness rule* for the nonce and key statement:

$$\frac{\#(ch_r)}{\#(ch_r, N_R \overset{K_M}{\longleftrightarrow} M_N)} \tag{18}$$

Then, on (18) & (17) we can apply the *nonce verification rule*:

$$\frac{N_R \mid\equiv (18), N_R \mid\equiv M_N \mid\sim (ch_r, N_R \overset{K_M}{\longleftrightarrow} M_N)}{N_R \mid\equiv M_N \mid\equiv (ch_r,\ N_R \overset{K_M}{\longleftrightarrow} M_N)} \tag{19}$$

Finally, to verify the goal G1.1, we can take the *belief rule*:

$$\frac{N_R \mid\equiv M_N \mid\equiv (ch_r,\ N_R \overset{K_M}{\longleftrightarrow} M_N)}{N_R \mid\equiv M_N \mid\equiv N_R \overset{K_M}{\longleftrightarrow} M_N} \tag{20}$$

The verification of goal G1.2 is symmetrical to G1.1, and thus also proved. For the second-order goals, we set additional assumptions. Since $X,\ X'$ from (9) and (10) are composed out of ch_r and ch_t, we assume by *freshness rule* that:

$$\frac{N_R \mid\equiv \#(ch_r)}{N_R \mid\equiv \#(X)}\ ,\ \frac{M_N \mid\equiv \#(ch_t)}{M_N \mid\equiv \#(X')} \tag{21}$$

Now, using the *belief rule* we get the important statements that both sides believe the session key possession:

$$\frac{N_R \mid\equiv (X,\ N_R \overset{K_S}{\longleftrightarrow} M_N)}{N_R \mid\equiv N_R \overset{K_S}{\longleftrightarrow} M_N} \tag{22}$$

$$\frac{M_N \mid\equiv (X',\ N_R \overset{K_S}{\longleftrightarrow} M_N)}{M_N \mid\equiv N_R \overset{K_S}{\longleftrightarrow} M_N} \tag{23}$$

To finalize the verification of G2.1 & G2.2, we use the same line of rules as for the G1.1 & G1.2, i.e., by applying

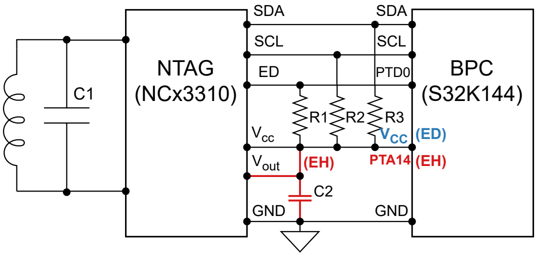


Fig. 6. Hardware design for the BPC that considers the wake-up use cases.

the message-meaning rule, then the freshness rule, nonce-verification and finally the belief rule. Here, G2.2 is also symmetrical in verification to G2.1.

C. Wake-up process evaluation

A real-world implementation was made using the components based on the hardware design presented in Figure 6. To test the design of ED, the NCx3310 is placed in standby mode with the S32K144 using the VLPS mode. For the idle phase, this results in $29.8\,\mu A$ from the S32K144 and $5.9\,\mu A$ from the NCx3310, for a total of $35.7\,\mu A$ as theoretical current consumption and power consumption of $117.81\,\mu W$. In the EH design case, the NTAG is completely disabled, which means that all power consumption comes only from the BPC, resulting in a theoretical current consumption of $29.8\,\mu A$ and power consumption of $98.3\,\mu W$ for our devices. While the design shows a working and usable test implementation, in a real application a level shifter should be considered to compensate for the potential cross currents that can occur on the connections between I2C, the return signal port, V_{cc} and V_{out}. Depending on the use case, both methods can be applied, but if the overall goal is to reduce power consumption, the method with EH is proposed. In this mode, the BPC remains in power-saving mode while the NTAG draws no power because it is completely powered down.

VII. Conclusion

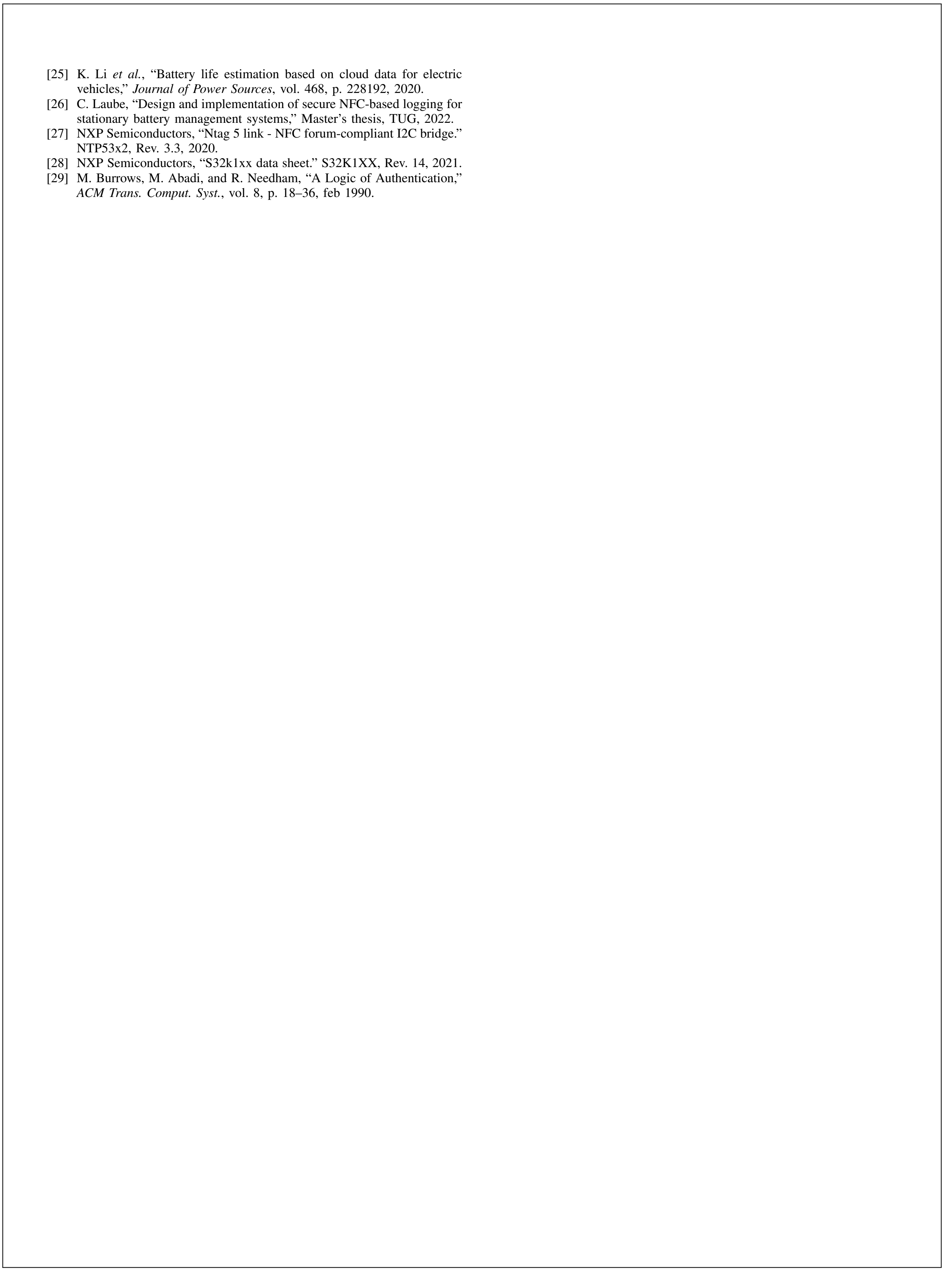
In this work, we have presented a novel design for the wireless deployment of BMS using NFC as an enabling technology, extending previous work. We have considered both internal and external system readouts, as well as active and inactive use cases. We show that it is feasible to use the BMS with NFC for both current active in-vehicle applications and second-life applications. An NFC interface is useful here as it allows flexible readout of the stored battery packs. Here, the security model was extended and evaluated using both informal security analysis and BAN logic. In addition, two wake-up system designs have been proposed and evaluated. For future work, further investigation of the wake-up method is planned, as well as investigation of other potential RFID options focused on lightweight applications, such as a fast readout of the ID of the stored battery packs.

Acknowledgment

This paper is supported by the OPEVA project that has received funding within the Key Digital Technologies Joint Undertaking (KDT JU) from the European Union's Horizon Europe Programme and the National Authorities (France, Belgium, Czechia, Italy, Portugal, Turkey, Switzerland), under grant agreement 101097267. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or KDT JU. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] X. Hu *et al.*, "State estimation for advanced battery management: Key challenges and future trends," *Renew. & Sustain. Energy Reviews*, 2019.
[2] M. Hartmann and J. Kelly, "Thermal Runaway Prevention of Li-ion Batteries by Novel Thermal Management System," in *IEEE ITEC*, 2018.
[3] W.-C. Lih, J.-H. Yen, F.-H. Shieh, and Y.-M. Liao, "Second Use of Retired Lithium-ion Battery Packs from Electric Vehicles: Technological Challenges, Cost Analysis and Optimal Business Model," in *IS3C*, 2012.
[4] R. Xiong *et al.*, "Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles," *IEEE Access*, vol. 6, 2018.
[5] "Proposal for a regulation of the European Parliament and of the Council concerning batteries and waste batteries, repealing Directive 2006/66/EC and amending Regulation (EU) No 2019/1020." https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52020PC0798, 2020.
[6] K. Berger, J.-P. Schöggl, and R. J. Baumgartner, "Digital battery passports to enable circular and sustainable value chains: Conceptualization and use cases," *Journal of Cleaner Production*, vol. 353, 2022.
[7] A. Samanta and S. S. Williamson, "A Survey of Wireless Battery Management System: Topology, Emerging Trends, and Challenges," *Electronics*, vol. 10, no. 18, 2021.
[8] F. A. Rincon Vija *et al.*, "From Wired to Wireless BMS in Electric Vehicles," in *17th MSN*, pp. 255–262, 2021.
[9] T. Kim *et al.*, "Cloud-Based Battery Condition Monitoring and Fault Diagnosis Platform for Large-Scale Lithium-Ion Battery Energy Storage Systems," *Energies*, vol. 11, no. 1, 2018.
[10] C. Shell *et al.*, "Implementation of a Wireless Battery Management System (WBMS)," in *IEEE I2MTC*, pp. 1954–1959, 2015.
[11] T. Gherman *et al.*, "Smart Integrated Charger with Wireless BMS for EVs," in *44th IECON*, pp. 2151–2156, 2018.
[12] X. Huang *et al.*, "Wireless Smart Battery Management System for Electric Vehicles," in *IEEE ECCE*, pp. 5620–5625, 2020.
[13] A. Rahman, M. Rahman, and M. Rashid, "Wireless Battery Management System of Electric Transport," *IOP Conference Series: Materials Science and Engineering*, vol. 260, nov 2017.
[14] S. Sripad *et al.*, "Vulnerabilities of Electric Vehicle Battery Packs to Cyberattacks on Auxiliary Components," *ArXiv*, vol. 1711.04822, 2017.
[15] M. Cheah and R. Stoker, "Cybersecurity of Battery Management Systems," *HM TR series*, vol. 10, no. 3, p. 8, 2019.
[16] S. Plosz *et al.*, "Security Vulnerabilities and Risks in Industrial Usage of Wireless Communication," in *IEEE ETFA*, pp. 1–8, 2014.
[17] N. A. Chattha, "NFC — Vulnerabilities and Defense," in *CIACS*, 2014.
[18] E. Haselsteiner and K. Breitfuss, "Security in Near Field Communication (NFC) Strengths and Weaknesses," in *Workshop on RFID Security*, 2006.
[19] F. Basic, M. Gaertner, and C. Steger, "Secure and Trustworthy NFC-Based Sensor Readout for Battery Packs in Battery Management Systems," *IEEE Journal of Radio Frequency Identification*, vol. 6, 2022.
[20] F. Basic *et al.*, "A Novel Secure NFC-based Approach for BMS Monitoring and Diagnostic Readout," in *IEEE RFID*, pp. 23–28, 2022.
[21] A. Reindl, H. Meier, and M. Niemetz, "Scalable, Decentralized Battery Management System Based on Self-organizing Nodes," in *ARCS*, 2020.
[22] "Standards - ISO 14443, ISO 15693, NFC-Forum und NDEF." https://www.nfc-tag-shop.de/info/nfc-entwicklung/standards-iso-14443-nfc-forum-und-ndef.html, 2017. Accessed: 16.01.2023.
[23] M. Bouklachi *et al.*, "Energy Harvesting of a NFC Flexible Patch for Medical Applications," in *IEEE WPTC*, pp. 249–252, 2019.
[24] W. Li *et al.*, "Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation," *Journal of Energy Storage*, vol. 30, 2020.



[25] K. Li *et al.*, “Battery life estimation based on cloud data for electric vehicles,” *Journal of Power Sources*, vol. 468, p. 228192, 2020.

[26] C. Laube, “Design and implementation of secure NFC-based logging for stationary battery management systems,” Master’s thesis, TUG, 2022.

[27] NXP Semiconductors, “Ntag 5 link - NFC forum-compliant I2C bridge.” NTP53x2, Rev. 3.3, 2020.

[28] NXP Semiconductors, “S32k1xx data sheet.” S32K1XX, Rev. 14, 2021.

[29] M. Burrows, M. Abadi, and R. Needham, “A Logic of Authentication,” *ACM Trans. Comput. Syst.*, vol. 8, p. 18–36, feb 1990.

A.8 [H] Secure Data Acquisition for Battery Management Systems

F. Basic, C. Seifert, C. Steger, and R. Kofler, "Secure Data Acquisition for Battery Management Systems," in *26th Euromicro Conference Series on Digital System Design (DSD)*, 2023, *In Press:* This paper has been peer-reviewed and accepted for the oral presentation at the conference.

Abstract. The growing awareness of environmental sustainability has led to new investments in the field of electric vehicles. One of the most expensive and important components of electric vehicles are their batteries, with battery management systems (BMS) being responsible for their control. New regulations, such as those of the European Union, aim to introduce battery passports as a way to track battery lifecycle from manufacturing, over second-life use, to recycling. Given the vast amount of data generated during the lifecycle of a battery, the current research is focused on combining BMS with cloud connectivity. However, not much research has yet been done in the area of BMS cloud security and secure data logging. To address this gap, we propose a novel solution for secure BMS data acquisition for on-premise and cloud environments. In this paper, we make two main contributions: a secure data structure for BMS logging and a secure architecture for transferring BMS data from its source to cloud and end systems. We demonstrate the feasibility of the design by developing a prototype with real components and evaluate it in terms of security and performance.

My Contribution. My contribution to this paper was in providing the main design both for the secure BMS chain data structure and BMS data propagation. These also include design motivational aspects, challenges, background, and related work. I provided the majority of the paper's text. I extended on the current BMS test suite and implemented secure BMS block handling and their integration with the rest of the environment, specifically with the gateway. I extended the gateway to account for additional processing steps necessary for the evaluation. Christian Seifert contributed by providing the implementation and measurements of the gateway and cloud connection, as well as integrating a server to represent the backend system. Christian Steger specified the overall big picture in relation to the targeted use cases and challenges. He also provided guidance on the paper's organization and content. Robert Kofler provided insight from the industry, use case specification, and design guidance in relation to the provided components. I realized the evaluation on the provided results, calculated the data overhead of the BMS block concept and concentrated other measurements for the overall system evaluation. I also analyzed and provided the security evaluation.

Secure Data Acquisition for Battery Management Systems

Fikret Basic, Christian Seifert, Christian Steger
Institute of Technical Informatics
Graz University of Technology
Graz, Austria
{basic, christian.seifert, steger}@tugraz.at

Robert Kofler
R&D Battery Management Systems
NXP Semiconductors Austria GmbH Co & KG
Gratkorn, Austria
robert.kofler@nxp.com

***Abstract*—The growing awareness of environmental sustainability has led to new investments in the field of electric vehicles. One of the most expensive and important components of electric vehicles are their batteries, with battery management systems (BMS) being responsible for their control. New regulations, such as those of the European Union, aim to introduce battery passports as a way to track battery lifecycle from manufacturing, over second-life use, to recycling. Given the vast amount of data generated during the lifecycle of a battery, the current research is focused on combining BMS with cloud connectivity. However, not much research has yet been done in the area of BMS cloud security and secure data logging. To address this gap, we propose a novel solution for secure BMS data acquisition for on-premise and cloud environments. In this paper, we make two main contributions: a secure data structure for BMS logging and a secure architecture for transferring BMS data from its source to cloud and end systems. We demonstrate the feasibility of the design by developing a prototype with real components and evaluate it in terms of security and performance.**

***Index Terms*—Battery Management System; Security; Cyber-physical; Cloud; Battery; Passport; Logging; Authentication.**

I. Introduction

The energy and environmental crisis caused by ever-increasing carbon emissions have led to an enormous increase in demand for electric vehicles. Battery management systems (BMS) play an important role in modern electric and hybrid vehicles by providing safety control over the use of batteries, their most important operating resource. They ensure the safety of the human driver by detecting and mitigating potential safety risks in advance [1]–[3]. As the use of electric vehicles, e-bikes, mopeds, etc. increases, so does the need for more batteries and thus BMS. In the recent market study published by Meticulous Research, the battery market is projected to reach $175.11 billion during the forecast period between 2021 and 2028 [4]. The exponential growth of the battery market between 2018 and 2025 and the importance of the second-life battery use case have also been observed by H. Melin [5]. The ever-growing market brings new challenges and requires solutions that would allow easier tracking and monitoring of batteries by their associated BMS. It is desirable to use batteries more efficiently and enable easier replacement at the end of their lifecycle to reduce global battery waste [6], [7]. We see two main challenges that need to be addressed in the development of modern BMSs.

BMS data reliance. The first challenge with modern BMSs lies in the need for efficient and easily accessible logging of monitoring and diagnostic processes, given the vast amount of generated data [8], [9]. Local logging devices may have limited capacity, be difficult to access, or even interfere with the regular operation of the BMS. On the other hand, with new regulations in the European Union (EU) and other countries around the world, the use of cloud systems for battery monitoring is slowly becoming a reality [10]–[12]. Several research papers have already proposed methods and models that use cloud services to extend the usability of BMS in a system [3], [9], [13]–[15]. Cloud systems enable the creation of battery lifecycle profiles, a concept that considers the collection and storage of important battery-related data from the BMS. This is done to extend user services and add external monitoring and diagnostic control by providing higher computing power and faster processing [3], [16].

Notably, the use of cloud connectivity with BMS offers:

- Battery life tracking and predictive support in the form of artificial intelligence or digital twins [3], [17].
- Increased computational power and faster processing of BMS-related diagnostic data, such as state of health (SoH) and state of charge (SoC) [18].
- Faster fault detection and battery age improvement [9].
- The use of ”swarming” to collect and use data for predictive maintenance of not just one but multiple systems in a group, e.g., for vehicle fleets [3], [9].

However, relying only on cloud services has three major disadvantages [8], [18]: (i) it requires a constant Internet connection e.g. if an accident occurs in a tunnel, there is no way to safely rely on the data during this transition, (ii) there may be delays due to the multi-level technological services that provide update control, and (iii) changes in data legislations and business models that may affect or complicate future ownership or access to the stored cloud data. An adequate BMS data service design should focus fully or partially on the use of local, i.e., on-premise, data services alongside conventional cloud connectivity. As Neubauer et al. [19] noted, it should be possible to handle both the batteries’ on-site measurements, as well as to track the average use over time to facilitate the second life use case.

BMS security. The second challenge for modern BMS is to provide an adequate and lightweight security design. Most of the current BMS cloud research focuses on predictive estimation, digital twins, and machine learning [3], [17], [18], leaving the area of BMS cloud security largely unexplored. A security architecture for advanced BMS communications must address all data transmission layers. At the BMS level, it is important to consider the security of collected battery diagnostic data. Manipulation of diagnostic data by malicious parties can lead to hazards, such as thermal runaway in vehicles [20]. It is also important to ensure that BMS data is only processed by authorized parties to protect user privacy [21]. In addition, BMS cloud connectivity suffers from threats and vulnerabilities common to general networks. Thus, a BMS must always consider protection against man-in-the-middle (MitM) attacks, unauthorized access to storage, and the use of outdated protocols [8].

Contributions. Our goal is to propose a solution to the upcoming BMS challenges and present a hybrid logging architecture that combines both on-premise and cloud services for BMS data logging. To this end, we propose:

- A general BMS data structure independent of any topology or use case aimed at BMS monitoring and diagnostic data handling, while addressing security requirements.
- Furthermore, we present a layered model for a secure BMS cloud architecture based on a centralized gateway. While several papers have been published recently on cloud utilization with BMS, most of them are based on data-driven models, data propagation, or cloud-enhanced algorithms.

To the best of our knowledge, this is the first work in the field of cloud BMS that places data logging structure reliance and security as the primary focus.

II. Background

A. BMS and cloud computing

A BMS is a system responsible for the safety control of a large set of battery packs. Namely, they are accountable for battery cell monitoring, diagnostics, overall system safety, charging, cell balancing, and controlling the optimal discharging use of battery packs [1]–[3]. There are several established BMS network topologies, with modern ones being primarily based on modulated and distributed architectures having a central BMS controller with several battery pack controllers (BPC) [16], [22]. The majority of today's cloud systems are based on providing services for data storage, processing, and sharing. They aim to provide flexible and extensible services to end users behind a complex facade. As such, cloud systems provide modern solutions for processing the vast amount of BMS data and enable services such as remote monitoring and predictive maintenance [17], [18].

B. Logging BMS data

Modern BMS are responsible for processing a large amount of generated data. This data must be logged, either internally in the BMS with special modules or externally, e.g. in the cloud. The challenge here is to specify a flexible BMS data structure for efficient transmission. The size of the logged data depends on the tracked parameters, sampling frequency, and compression [8]. The BMS are responsible for interpreting the diagnostic data derived from the BPC or directly from the battery cells. For our analysis, we will focus on the following three main groups of data: (i) monitored data; which considers on-board read sensor or other measured data, e.g., battery cell voltage and temperature, (ii) diagnostic data, i.e. derived data based on observations, usually from a BPC, another module, or directly from the main BMS controller, and transmitted either as raw data or information such as SoC or SoH, (iii) fault diagnostic data, i.e., raw data derived from the register responsible for tracking the parameters of individual battery cells. Traditionally, most of the data generated on the BMS side was only stored locally for active monitoring. With the new initiatives related to the battery passport and secondary use, portions of this data would also need to be stored or sent to other systems to maintain tracking of the battery pack life cycle [11], [12]. For the remainder of this paper, we will refer to any BMS stored data as the BMS logging (log) data.

C. Battery passports and second life

Electric vehicle (EV) batteries reach the end of their life after 8 to 10 years of use, i.e., after they have dropped to 80% of their full capacity [7], [23]. After that, the batteries are either recycled or disassembled [5]. This is becoming a problem as the market for batteries keeps increasing, the rate at which they are recycled becomes limited and expensive, which also creates more environmental waste [5]. There is an initiative to allow batteries that have reached the end of their life in EV to be used for other applications, such as self-consumption in households or transmission deferral from EV to power grids [7], [24]. The EU Commission proposes the use of battery passports to track the lifecycle of batteries [10]–[12]. The battery passport is intended to be a digital representation of a battery that conveys all-important product information [10], [12]. An extension of this concept would be the battery e-passport, which could also dynamically record battery charge and discharge cycles, diagnostic information, faults, cell health, etc. This information could be used for rapid processing when the battery gets a second life in other applications during the disassembly process [5], [7]. Cloud systems provide such a solution, but there is currently no clear answer to secure BMS processing from the local to the cloud level, which we aim to extend in this work.

III. The Novel Secure BMS Data Structure Design

Based on the BMS lifecycle monitoring requirements, we propose a BMS data structure design based on a hierarchical distribution and differentiate between (i) log blocks, (ii) BMS blocks, and (iii) secure BMS blocks. A "*Log block*" (List. 3 with List. 1 & List. 2) contains a log header and a log body, where the log body contains the logged data based on a given structure. Log blocks form a payload that is represented by the "*BMS block*" (List. 4). To keep track of the sampling order,

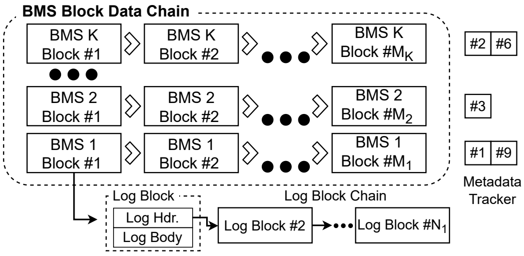


Fig. 1. BMS block data chain structure for logging of BMS lifecycle data.

the BMS blocks are intended to be stored in a data chain structure, as shown in Figure 1. Each BMS block contains a pointer to the first log block, which in turn contains a pointer to the next log block in the sequence. They are intended to encapsulate one BMS sub-system, with the “log chain” containing individual log samples per battery pack. The BMS block data chain identifies each individual sample point. In this example, ‘*M*’ indicates the number of currently logged BMS blocks in a BMS sub-system, while ‘*N*’ presents the total number of log blocks per one BMS block. ‘*K*’ is the number of tracked BMS sub-systems. The sequence of the BMS blocks is determined by their timestamp field. Battery passport data is contained in the metadata field. To save space, metadata may be sent only when its contents change, for example, when the BMS sub-system changes its configuration or its host system. An array can be implemented that tracks BMS blocks with major status changes. Each time a new BMS block is received that contains metadata, the tracker array is incremented by the identifier of that block.

The advantage of the proposed data model structure is that it can be used across all different BMS topologies [22] with the following considerations for one full sample:

- Centralized: 1 BMS block, 1 Log block.
- Modulated: 1 BMS block, N Log blocks
- Distributed: 1 BMS block, N Log blocks
- Decentralized: K BMS blocks, $N_1, ..., N_K$ Log blocks

To guard against eavesdropping or other potential manipulations with logged BMS data, the application data exchange is protected via the ”*Secure BMS block*” (List. 5), which uses the BMS block and attached log blocks as input for the encryption payload. Once the secure BMS block has successfully arrived at the end system, it can be decrypted and integrated into the log chain structure. Figure 2 shows the structure of the secure BMS block that is transmitted with each log sample. The secure BMS block consists of the unencrypted security header, the encrypted BMS block, and a security tag as a footer. The header contains information such as the current BMS secure identifier and cipher suite code. The security tag is used to verify the authenticity and integrity of the BMS block. It can be computed using the Message Authentication Code (MAC) or another security-related tag operation. The BMS block also contains its own header, logged data, and optional metadata. We propose that the header of the BMS block contain at least a unique identifier, a timestamp, the classifier ID, and the pointer to the first BMS log block. The protocol block contains at least its identifier, the identifier of the next block in the chain, and the block length.

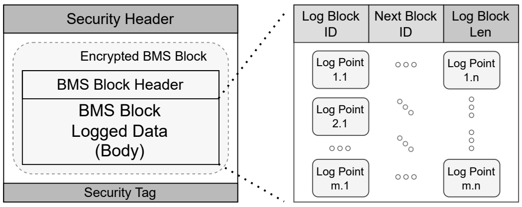


Fig. 2. Proposed design of the secure BMS data block.

```
Struct LogSample contains
    Log_Meas* measurements;
    Log_Diag* diagnostics;
    Log_Fault* fault_regs;
end
```

Listing 1: Data logging sample data structure.

```
Struct LogBlockHdr contains
    int block_id;
    int next_block_id;
    uint32 block_body_len;
end
```

Listing 2: Log block header data structure.

```
Struct LogBlock contains
    LogBlockHdr log_header;
    LogSample    log_body;
end
```

Listing 3: Log block with log header and body entries.

```
Struct BmsBlock contains
    uint32 bms_block_id;
    uint32 timestamp;
    uint16 unit_id;
    int init_log_block_id;
    uint16 metadata_len;
    Bms_Metadata bms_metadata;
end
```

Listing 4: BMS block structure with optional metadata.

```
Struct SecBmsBlock contains
    uint16 version;
    uint16 length;
    uint32 sec_bms_block_id;
    uint32 sec_bms_block_serial;
    uint16 cipher_info;
    uint16 enc_bms_block_len;
    uint8* iv;
    uint8* enc_bms_block;
    uint8* mac;
end
```

Listing 5: Secure BMS block for symmetric crypto.

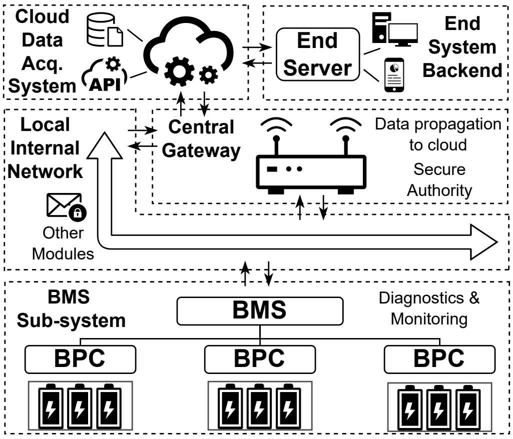


Fig. 3. Proposed BMS architecture for battery lifecycle monitoring.

IV. Secure BMS Data Acquisition Architecture

A. Security requirements

We observe the secure architecture from two perspectives: (i) the security of "cloud layers", i.e., the security on and from the central gateway, to the cloud acquisition system, and finally the end system, and (ii) BMS sub-system and its network.

For a BMS, security can be considered as (i) security during data transmission and (ii) security within the device. We want to ensure that the data from the source (battery sensors) to the end device (cloud system or end-user systems) is not compromised [6], [22]. Attacks targeting the BMS itself would be difficult to carry out because BMS communicate with battery packs that are enclosed and isolated. Still, various attacks could take place, usually in the form of spoofed devices or remote attacks if a vulnerability is found [6], [16]. MitM attacks are possible either between the BMS and the central gateway or when connecting to the cloud system. Data must be protected against these forms of attack by guaranteeing its authenticity, integrity, and confidentiality [21]. In addition, protection must be provided by message counters and inspections to ward off various replay attacks. Other attacks may take the form of denial-of-service (DoS) attacks, which would target either the cloud systems or the local networks. The attacker can also target the log content itself by launching attacks on delayed, reordered, or manipulated packets [25]. Accurate implementation and validation on the end system side should be performed to mitigate these types of attacks.

To support the proposed security architecture and secure BMS data structure, we observe the following design points:

- The security architecture with the cloud system is done over a trusted and verifiable service.
- The key generation and distribution by the original equipment manufacturer (OEM) are based on a trusted design, i.e., the end system device can securely receive the key necessary to decrypt the received BMS log data.
- Security operations are done over a trusted secure module to mitigate hardware-based vulnerabilities.

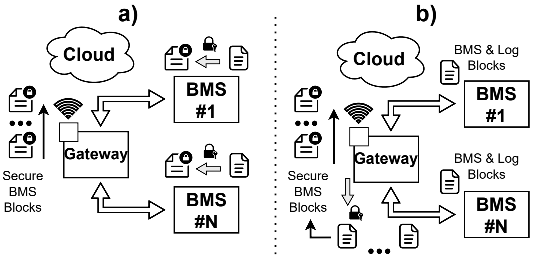


Fig. 4. Suggested secure BMS block data on-premise processing methods: a) security layer already added on BMS, b) security layer added on the gateway.

B. System layers

While different publications might refer to them with different notations, BMS cloud architectures generally consist of a perceptual layer, a network layer, and an end-user application layer [2], [15]. We further divide these layers to account for two additional middle layers to consider the on-premise BMS activities. Our main architecture design is presented in Figure 3. The system consists of five main layers: (i) BMS subsystems, (ii) internal local network, (iii) central gateway, (iv) cloud data acquisition system, and (v) end system backend.

BMS sub-system. This layer considers the BMS and its connected battery packs as an independent entity. As an enclosed system, it is difficult to perform attacks from the outside. Nevertheless, it is recommended to provide authentication at the device level. The main BMS controller is responsible for data collection and preparation of the secure BMS blocks. This should be done on the device using a secure module. Due to widespread use, it is proposed to rely on symmetric encryption algorithms, specifically the traditional block ciphers, e.g., Advanced Encryption Standard (AES) with modes or authentication encryption (AE) primitives.

Internal local network. At the internal network system layer, device authentication, key derivation, and session establishment are performed using an appropriate security architecture [26]. This considers the communication between the BMS controller, the gateway, and any other local device. The secure session is established using either static or dynamic keys, taking into account perfect forward secrecy [27].

Central gateway (GW). The central device responsible for collecting BMS log data. It enables the connection with the cloud system. It is also the secure authority for the local network. For performance reasons, the rest of the data encapsulation is done on the GW side to prepare the BMS blocks for transmission over the cloud to the end system.

Cloud data acquisition system. Cloud systems rely on proven and robust security protocols. Cloud data communications for IoT solutions typically rely on the use of the underlying Transport Layer Security (TLS) or Datagram-TLS (DTLS) layers. At the application layer, DTLS is often used with the CoAP protocol and TLS with Message Queuing Telemetry Transport (MQTT) [28]. DTLS improves performance, but data might be lost, requiring retransmissions. The use of the right protocol depends primarily on the intended use case, with

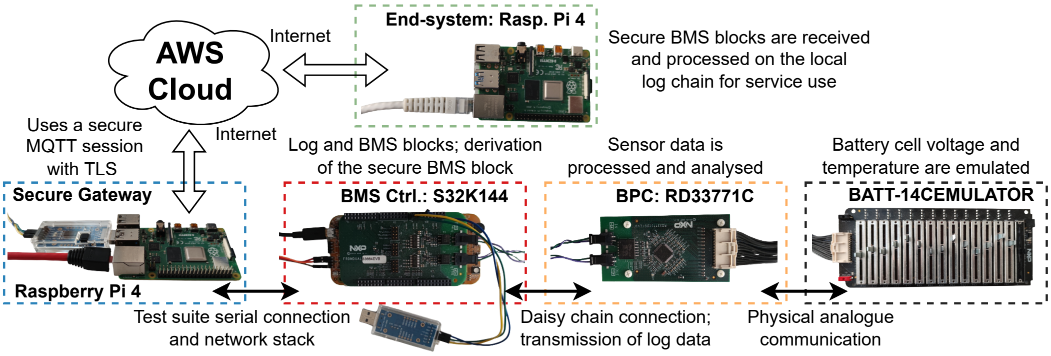

AWS Cloud
Internet
End-system: Rasp. Pi 4
Secure BMS blocks are received and processed on the local log chain for service use
Uses a secure MQTT session with TLS
Internet
Log and BMS blocks; derivation of the secure BMS block
Sensor data is processed and analysed
Battery cell voltage and temperature are emulated
Secure Gateway
BMS Ctrl.: S32K144
BPC: RD33771C
BATT-14CEMULATOR
Raspberry Pi 4
Test suite serial connection and network stack
Daisy chain connection; transmission of log data
Physical analogue communication

Fig. 5. Deployed BMS test suite. BMS sub-system: S32K144 BMS ctrl. with BPCs and battery emulators, Internal local network: implemented full local secure stack over a serial link, Central gateway: Rasp. Pi 4, Cloud data acq. system: Amazon web server with device shadow, End system backend: Rasp. Pi 4.

BMS relying on stricter safety standards and regulations that must be supplemented. It should be noted that both TLS and DTLS provide data protection from BMS to the gateway, but not for the end-to-end devices, i.e., from a BMS to the end device, which is why we also consider security at other layers.

End system backend. This considers any server or user device that handles the BMS log data for data processing, visualization, etc. Specifically, it is a device that relies on the battery e-passport services for lifecycle tracking and one that is authorized to access the encrypted secure BMS block data.

C. End system infrastructure discussion

For the presented architecture, an effort was made to keep the design generic and flexible regardless of the underlying session key exchange design. The end system receives the encrypted BMS data, but the means of processing this data is left open. Either the same or an additional external server would need to be used to disseminate and share the necessary certificates and other security-related configuration data. A similar concept could be adopted from the proposed standard ISO 15118 regarding the distribution of certificates to the respective OEM [29], [30]. Security solutions such as end-to-end encryption with adjustments could also be employed [31].

For processing individual BMS blocks to secure BMS blocks, we observe two approaches as seen in Figure 4:

a) Security functionality and storage are performed on the main BMS controller, i.e., it sends full encrypted data.
b) Security handling is performed on the gateway device.

We focus on the first approach, where data processing is performed on individual main BMS controllers. Here, the gateway acts as a buffer and bridge to ensure that each data block is correctly received and processed to the cloud service. This is an advantage for ad hoc BMS sub-systems that may store intermediate data between sessions or rely only on the on-premise use case. It is also more flexible for decentralized topologies where processing between BMS units is independent and therefore there is no bottleneck at the gateway. In addition, there are systems where there may not be a secure gateway and communication with cloud systems is done directly through the main BMS controller.

V. System Prototype Implementation

We implemented our proposed design on real hardware to demonstrate the feasibility of the presented methods, shown in Figure 5. A BMS emulator from NXP Semiconductors was used, consisting of an S32K144 microcontroller as the BMS controller, an RD33771C as the BPC, and a battery emulator that generates battery voltage and temperature data. To emulate local network communication, we used a Raspberry Pi 4 as the gateway device. Appropriate software was implemented and integrated for both the S32K144 and the Raspberry Pi to enable encapsulation of the test data and secure communication transmission. The security functionality was provided via the BearSSL library [32]. Security communication handling and secure BMS blocks were implemented based on an existing BMS diagnostic functionality. The security architecture for authentication and session key derivation is based on an Elliptic Curve Qu-Vanstone model. The communication between the gateway and the BMS controller is done over a serial link configured at a baud rate of 57 kbps, and uses the implemented network stack discussed in Section V-A. A symmetric cryptographic model was used for the secure BMS block, relying on the AES and hash-MAC (HMAC).

A. Test suite communication model

The test suite relies on the use of different network communication layers. The structures of the data packets used at each layer can be seen in Figure 6. The format of the data layer can be tailored to the needs of the target system. In our case, it is aimed at the communication between the secure gateway and the BMS controller, but it remains flexible and open.

The application layer is responsible for transmitting application-specific data, i.e., log data. The secure BMS block is included as a payload for our test cases. The application payload data is encrypted and tagged along with the added header to protect data confidentiality and integrity in the internal network. The transport layer allows for the fragmentation of large payloads. In our test case, the data-link layer packets can only contain up to 255 bytes in one packet, so the additional transport layer is required. It is modeled after the ISO 15765-2 standard, which is also used in similar environments [33].

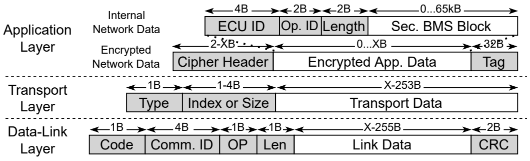


Fig. 6. Developed network communication model for the BMS test suite.

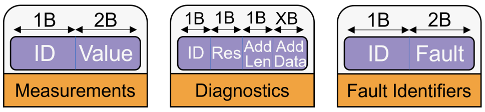


Fig. 7. Derived BMS log data model for the lifecycle monitoring.

B. Cloud setup

Cloud hosting is done on an Amazon web server (AWS). The gateway communicates with AWS and sends data using the secure MQTT protocol. After receiving the BMS block data, AWS propagates it directly to the assigned end system using an HTTPS REST push request. Both rely on the TLS protocol. In our test environment, the end system is represented by a Raspberry Pi 4 hosting a local server. The end system hosts a web application to display the captured BMS lifecycle data, running on a Flask server.

C. BMS data structure preprocessing

The battery log data collected by BPC and processed by BMS is based on the structure described in Section II-B and with List. 1, with the format shown in Figure 7. We optimize the processed data to incur as little overhead as possible. The monitored data includes a total of 3 bytes, one for the ID and two for the raw value. At least 3 bytes are allocated to the diagnostic data: one for ID, one for the diagnostic status, and one for the length of the additional data. The additional data is application specific. If there is no additional data per diagnostic entry, the length of the additional data is zero. The total amount of data for a log sample is 162 bytes. We assume that our system will have no more than 256 monitoring and diagnostic entries per BPC, so only one byte is reserved for identification. Otherwise, this entry could also be extended.

VI. Evaluation

A. Security analysis

We analyze the proposed design in terms of achieved security. The analysis is based on the security requirements described in Section IV-A. For our analysis, we derive *assets* {A}, *threats* {T}, *countermeasures* {C}, and *assumptions* {As}. The following assets are derived, i.e., the objects of protection: (A1) BMS log data, (A2) gateway-to-cloud payload, and (A3) cloud-to-end-system payload. To limit the security analysis to our proposed solution, we make the following assumptions: (As1) battery sensors, BPC and their channels are considered secure and trusted, (As2) no attack in the form of physical tampering is possible, (As3) security functions and keys are stored in a protected storage environment, (As4) cloud and end system are protected against common online threats. The security analysis model is built using the data flow diagram (DFD), with the derived model and results shown in Figure 8.

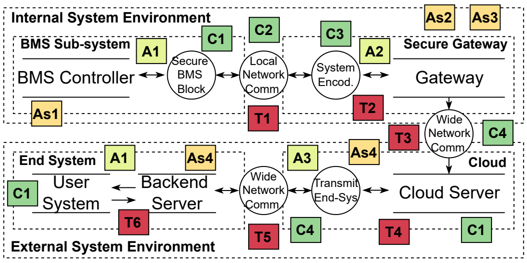


Fig. 8. Data flow diagram of BMS lifecycle monitoring security analysis.

From the analysis, we list each potential threat with the targeted asset and analysed countermeasure strategies. Under "network attacks", we consider eavesdropping, tampering, replay, and MitM attacks on the messages. The threats are:

- [T1] Network attack on the BMS sent log data $\mapsto (A1)$, (C1) using secure BMS block with encryption and MAC, and with (C2) pre-established secure network session.
- [T2] Spoofing attack on the gateway $\mapsto (A1), (A2)$, (C3) the gateway is a secure authority with authentication.
- [T3] Network attack on the pushed cloud data $\mapsto (A2)$, (C4) uses secure MQTT, and TLS with certificates.
- [T4] Spoofing and privacy attacks on the cloud $\mapsto (A2)$, even if compromised, BMS data is protected with (C1).
- [T5] Network attack on the end system transfer $\mapsto (A3)$, (C4) again, with TLS and HTTPS for the pull request.
- [T6] BMS log data confidentiality compromise $\mapsto (A1)$, similar to [T4], (C1) mitigates any unauthorized readout.

B. Overhead analysis

The data overhead comes in the form of additional header data. It can be divided into two parts: static and dynamic data. The static data has a fixed size, regardless of the amount of processed log data for each secure BMS block. Here we refer to the variable sizes from Section III. Both secure and standard BMS blocks require 16 bytes each, with each additional log block requiring 12 bytes for its own header, regardless of the total size of the log body. Thus, the formula for calculating the total header size is $32 + 12 * X$ bytes, where 'X' is the total number of log blocks. The dynamic component entails only the metadata in the BMS block, which is optional and depends on the implementation. It also depends on the underlying cypher protocols, i.e., larger block sizes mean larger key and MAC values. The total log block header size is also dynamic, as it depends on the total number of log blocks, i.e., BPC. Figure 9 shows the theoretical data overhead relative to the total secure BMS block size with variable log block length, where Table I shows the overhead analysis for the test suite with a variable number of log blocks, i.e., each with a size of 162 bytes. As noted, the secure BMS block data structure allows for minimal impact on overhead when deployed in a real-world environment, as it does not scale with the log block size.

TABLE I
OVERHEAD PER BMS BLOCK COMPARED TO THE NUM. OF LOG BLOCKS.

# of Log Blocks	1	2	4	8	12	16	32
Overhead (%)	21.4	14.7	11.0	9.0	8.3	8.0	7.4

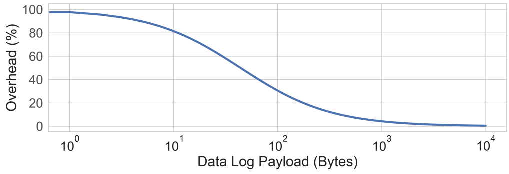


Fig. 9. Secure BMS block overhead in relation to log payload for one block.

C. BMS block encoding analysis

We analyze the encoding on the BMS controller using: (i) real hardware with emulated battery data, and (ii) simulated log input to the BMS controller for greater sizes. Under encoding, we consider the time of secure BMS block preparation after sampling and also the secure local network packet encoding. Table II shows the average results after one hundred test runs for individual encoding phases using emulators. The standard deviation is not included as it was negligible and $< 0.01\,ms$ for all test cases. As concluded, most of the encoding time is spent on security functions, which means that their optimization primarily affects the duration of the encoding process. Figure 10 shows the total encoding time for the simulated data. It shows the linear growth of the encoding time for three log block sizes compared to up to 10 BPC.

For the gateway, we want to ensure that the following goal is met: keep the transmission and processing time at a minimum, with the lowest time equal to the total sampling and processing time on the BMS controllers. The time for the decoding of the secure BMS blocks, i.e., the decryption from the application layer, MAC verification, and data extraction, is negligible compared to the total BMS processing time. Full network decoding accounts for $1.35\,ms \pm 0.11\,ms$, where decoding of the secure BMS block is $0.48\,ms \pm 0.01\,ms$. We note that this metric is highly dependent on the system and implementation used, but we can assume based on the devices used for this test suite that the same criteria would also be met in real systems. The sampling rate of the battery data is application dependent, but in our test suite, it was $\approx 112\,ms$ per BPC.

D. Transmission measurement

The tests were performed with one BPC on real hardware. The first step is the transmission from the BMS to the gateway, which takes $85.2\,ms \pm 3\,ms$. After decoding, the transmission via the gateway (Rasp. Pi 4) to the cloud (AWS) works on the *device shadow* principle, i.e., data is automatically forwarded when the BMS shadow is updated to ensure that log data is read only when needed. This step requires a total of $1.37\,s \pm 0.2\,s$ per request. After receiving the data from the gateway, the AWS forwards it to the end system (Rasp. Pi 4), which then decodes it and further processes the BMS block, requiring only $1.6\,ms \pm 0.4\,ms$. The Rasp. Pi measurements were averaged after one hundred runs. We have also tested decoding by adding additional data per message at various intervals up to 1 kB, but found only a small increase in processing time. The reported time for the additional 1 kB payload is $1.39\,s \pm 0.21\,s$ for the gateway and $2.2\,ms \pm 0.9\,ms$ for the end system. For a complete run, we took measurements from the battery sampling to the end system, as shown in Figure 11. The first plot shows the variation in time over the increase in additional logging data for one hundred cycles, while the second plot shows the total time for a different number of sampling cycles with no additional data.

TABLE II
BMS BLOCK ENCODING TIME WITH THE EMULATED DEVICES.

Encoding	Log body	BMS block	Secure block	Secure network
1 BPC	$0.09\,ms$	$0.24\,ms$	$18.16\,ms$	$20.68\,ms$
2 BPC	$0.17\,ms$	$0.38\,ms$	$23.63\,ms$	$26.33\,ms$

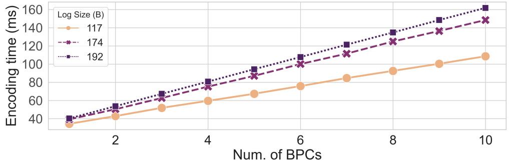


Fig. 10. Simulated BMS log encoding time for 117, 174 & 192 B log sizes.

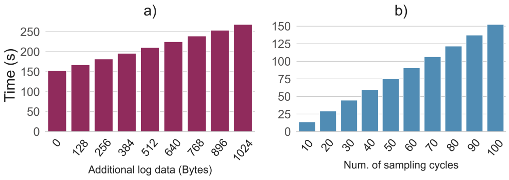


Fig. 11. Complete implementation run times for lifecycle monitoring with: a) variable additional log data, b) incrementing number of cycles.

We see that the main bottleneck is in the transmission of data to the cloud system, where multiple BMS sampling cycles can be performed during one cloud request. However, as mentioned earlier, even without considering further optimizations, the data is temporarily stored on the gateway device and can be pushed on the next request. The BMS can continue to safely sample new data without having to change its operating rate.

VII. RELATED WORK

The use of the cloud in conjunction with BMS has gained significant momentum in recent years, although many questions remain, especially those related to data storage and distribution of services [13]. While most of the publications are from recent years [15], some of the earlier proposals came from industry, notably from Fujitsu, where a system was envisioned that combines the use of cloud services for battery-sharing information between BMS [14]. More recently, Yang et al. [15] present a BMS cloud architecture based on the cyber hierarchy and interactive network framework, although they do not look into the BMS data acquisition design.

Digital twins are becoming increasingly associated with BMS in the context of cloud connectivity. In this area, Li et al. [3] describe a model for comparing measured battery data and estimated digital twin data, with estimates based on the use of extended H-infinity filters and particle swarm optimization. Similarly, Wu et al. [17] present a cloud-side data-driven solution for BMS SoH estimation by focusing on machine learning methods for input noise reduction and using random forest regression to build a battery degradation model.

Concerning data logging, Mansor et al. [34] propose a secure logging approach for vehicles based on the use of mobile and cloud applications. Their focus is on the use of hardware security modules for in-vehicle units, such as those proposed in the EVITA project [35]. For solutions specifically targeting BMS, Zhou et al. [8] present a frequency division based storage and compression method that can be used for BMS log data. In their work, they also present three main requirements for using large battery storage. Among them, the limitations of the communication technology used in terms of data rate as well as the duration of data storage are argued. In our paper, we propose a design that is independent of these system constraints. However, we do consider the amount of argued data as one of the requirements which we discuss in Sect. III and implement in Sect. V. The paper also points out the possibility of bottlenecks when transmitting a large amount of BMS data. Our design provides a solution to this challenge by partitioning the task management of data acquisition and forwarding when a central gateway is considered.

VIII. Conclusion

In this paper, we have presented a novel approach to secure BMS lifecycle monitoring considering both on-premise and cloud environments. The proposed architecture has been developed in mind for the current and upcoming use cases concerning battery passports and regulations. The design allows for intermediate secure storage of BMS blocks on a local gateway device, before they are able to be further processed on the cloud and end systems. The BMS data is securely processed from the main BMS controller, over the internal network, to the cloud service and end system. A demonstrator has been successfully implemented to evaluate the design's performance in real-world environments. For future work, we see solutions such as OSCORE [28], aimed at constrained IoT devices, as a potential extension to the current security design on the local BMS network layer. Additionally, it is planned to cover modern solutions used for plug-and-charge services and create an adaptive layer with the current data structure design.

Acknowledgment

This paper is supported by the OPEVA project that has received funding within the Key Digital Technologies Joint Undertaking (KDT JU) from the European Union's Horizon Europe Programme and the National Authorities (France, Belgium, Czechia, Italy, Portugal, Turkey, Switzerland), under grant agreement 101097267. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or KDT JU. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] H. A. Gabbar *et al.*, "Review of Battery Management Systems (BMS) Development and Industrial Standards," *Technologies*, vol. 9, 2021.

[2] R. Xiong and W. Shen, *Advanced Battery Management Technologies for Electric Vehicles*. Wiley, 2019.

[3] W. Li *et al.*, "Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation," *Journal of Energy Storage*, vol. 30, p. 101557, 2020.

[4] "Electric Vehicle Battery Market - Global Forecast to 2028," tech. rep., Meticulous Research, Nov 2021.

[5] H. Eric Melin, "The lithium-ion battery end-of-life market – A baseline study," tech. rep., Global Battery Alliance, 2018.

[6] B. Jayaraman, "EVERLASTING D8.16 – White Paper 13," tech. rep., EU Project, 08 2020.

[7] L. C. Casals, B. Amante García, and C. Canal, "Second life batteries lifespan: Rest of useful life and environmental analysis," *Journal of Environmental Management*, vol. 232, pp. 354–363, 2019.

[8] L. Zhou *et al.*, "Massive battery pack data compression and reconstruction using a frequency division model in battery management systems," *Journal of Energy Storage*, vol. 28, p. 101252, 2020.

[9] M. S. Hossain Lipu *et al.*, "Smart Battery Management Technology in Electric Vehicle Applications: Analytical and Technical Assessment toward Emerging Future Directions," *Batteries*, vol. 8, no. 11, 2022.

[10] GBA, "Battery Passport." https://www.globalbattery.org/battery-passport/, 2023. Accessed: 07.03.2023.

[11] "Proposal for a regulation of the European Parliament and of the Council concerning batteries and waste batteries, repealing Directive 2006/66/EC and amending Regulation (EU) No 2019/1020." https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52020PC0798, 2020.

[12] V. Halleux, "EU Legislation. New EU regulatory framework for batteries.," tech. rep., Members' Research Service, 2022.

[13] M.-K. Tran *et al.*, "Concept Review of a Cloud-Based Smart Battery Management System for Lithium-Ion Batteries: Feasibility, Logistics, and Functionality," *Batteries*, vol. 8, no. 2, 2022.

[14] T. Tanizawa *et al.*, "Cloud-connected Battery Management System Supporting e-Mobility," *Fujitsu Science Tech. Journal*, vol. 51, 2015.

[15] S. Yang *et al.*, "Implementation for a cloud battery management system based on the CHAIN framework," *Energy and AI*, vol. 5, 2021.

[16] A. Samanta and S. S. Williamson, "A Survey of Wireless Battery Management System: Topology, Emerging Trends, and Challenges," *Electronics*, vol. 10, no. 18, 2021.

[17] J. Wu, X. Liu, J. Meng, and M. Lin, "Cloud-to-edge based state of health estimation method for lithium-ion battery in distributed energy storage system," *Journal of Energy Storage*, vol. 41, p. 102974, 2021.

[18] K. Li *et al.*, "Battery life estimation based on cloud data for electric vehicles," *Journal of Power Sources*, vol. 468, p. 228192, 2020.

[19] J. Neubauer *et al.*, "A Second Life for Electric Vehicle Batteries: Answering Questions on Battery Degradation and Value," *SAE International Journal of Materials and Manufacturing*, vol. 8, 2015.

[20] W. Huang *et al.*, "Questions and Answers Relating to Lithium-Ion Battery Safety Issues," *Cell Reports Physical Science*, vol. 2, 2021.

[21] S. Kumbhar *et al.*, "Cybersecurity for Battery Management Systems in Cyber-Physical Environments," *IEEE ITEC*, pp. 761–766, 2018.

[22] A. Reindl *et al.*, "Scalable, Decentralized Battery Management System Based on Self-organizing Nodes," in *ARCS*, pp. 171–184, 2020.

[23] E. Wood *et al.*, "Investigation of battery end-of-life conditions for plug-in hybrid electric vehicles," *Journal of Power Sources*, vol. 196, 2011.

[24] R. Faria *et al.*, "Primary and secondary use of electric mobility batteries from a life cycle perspective," *Journal of Power Sources*, vol. 262, 2014.

[25] A. Ali *et al.*, "BCALS: Blockchain-Based Secure Log Management System for Cloud Computing," *Trans. E. Telecomm. Tech.*, vol. 33, 2022.

[26] F. Basic *et al.*, "Trust your BMS: Designing a Lightweight Authentication Architecture for Industrial Networks," in *IEEE ICIT*, 2022.

[27] F. Basic *et al.*, "Establishing Dynamic Secure Sessions for ECQV Implicit Certificates in Embedded Systems," in *DATE*, pp. 1–6, 2023.

[28] M. Gunnarsson *et al.*, "Evaluating the performance of the OSCORE security protocol in constrained IoT environments," *Internet of Things*, vol. 13, p. 100333, 2021.

[29] ISO, "ISO 15118-20:2022 Road vehicles — Vehicle to grid communication interface - Part 20," 2022.

[30] A. Fuchs *et al.*, “HIP-20: Integration of Vehicle-HSM-Generated Credentials into Plug-and-Charge Infrastructure,” in *4th ACM CSCS*, 2020.

[31] B. Hale and C. Komlo, “On End-to-End Encryption.” Cryptology ePrint Archive, Paper 2022/449, 2022.

[32] “BearSSL.” https://bearssl.org/, 2018. Accessed: 25.03.2023.

[33] ISO, “ISO 15765-2:2016 Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 2,” 2016.

[34] H. Mansor *et al.*, “Log Your Car: The Non-invasive Vehicle Forensics,” in *IEEE Trustcom/BigDataSE/ISPA*, pp. 974–982, 2016.

[35] “EVITA. E-safety vehicle intrusion protected applications.” https://evita-project.org/, 2011. Accessed: 08.03.2023.

Bibliography

[1] Y. Wu and L. Zhang, "Can the development of electric vehicles reduce the emission of air pollutants and greenhouse gases in developing countries?" *Transportation Research Part D: Transport and Environment*, vol. 51, pp. 129–145, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920916302759 → Cited on page 1.

[2] J. Fleischmann, M. Hanicke, E. Horetsky, D. Ibrahim, S. Jautelat, M. Linder, P. Schaufuss, L. Torscht, and A. van de Rijt, "Battery 2030: Resilient, sustainable, and circular," McKinsey with Global Battery Alliance, Tech. Rep., Jan 2023. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/battery-2030-resilient-sustainable-and-circular/ → Cited on pages 1, 2, and 19.

[3] H. Eric Melin, "The lithium-ion battery end-of-life market – A baseline study," Global Battery Alliance, Tech. Rep., 2018. [Online]. Available: https://www3.weforum.org/docs/GBA_EOL_baseline_Circular_Energy_Storage.pdf → Cited on pages 1, 2, and 7.

[4] Business Wire, "Global Lithium-Ion Battery Recycling Markets 2018-2025: Focus on Technology, Chemistry, End Source, and Regional Analysis - ResearchAndMarkets.com," https://www.nxp.com/company/about-nxp/ai-powered-cloud-connected-battery-management-system-for-electric-vehicles:NW-NXP-AI-POWERED-CLOUD-CONNECTED-BATTERY, 2020, Accessed: 2023-03-02. → Cited on page 1.

[5] Meticulous Research, "Electric Vehicle Battery Market - Global Forecast to 2028," Meticulous Market Research Pvt. Ltd., Tech. Rep., Nov 2021. [Online]. Available: https://www.meticulousresearch.com/product/EV-battery-market-5210 → Cited on pages 1 and 7.

[6] "Global EV Outlook 2022," https://iea.blob.core.windows.net/assets/e0d2081d-487d-4818-8c59-69b638969f9e/GlobalElectricVehicleOutlook2022.pdf, International Energy Agency (IEA), 2022, Accessed: 08.05.2023. → Cited on page 1.

[7] M. Li, J. Yang, S. Liang, H. Hou, J. Hu, B. Liu, and R. Kumar, "Review on clean recovery of discarded/spent lead-acid battery and trends of recycled products," *Journal of Power Sources*, vol. 436, p. 226853, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775319308468 → Cited on page 1.

[8] Y. Hu, Y. Yu, K. Huang, and L. Wang, "Development tendency and future response about the recycling methods of spent lithium-ion batteries based on bibliometrics analysis," *Journal of Energy Storage*, vol. 27, p. 101111, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X19307881 → Cited on page 1.

[9] A. Kwade, W. Haselrieder, R. Leithoff, A. Modlinger, F. Dietrich, and K. Droeder, "Current status and challenges for automotive battery production technologies," *Nature Energy*, vol. 3, 2018. [Online]. Available: https://www.nature.com/articles/s41560-018-0130-3 → Cited on pages 1 and 2.

[10] W.-C. Lih, J.-H. Yen, F.-H. Shieh, and Y.-M. Liao, "Second Use of Retired Lithium-ion Battery Packs from Electric Vehicles: Technological Challenges, Cost Analysis and Optimal Business Model," in *IS3C*, 2012. → Cited on page 1.

[11] B. Jayaraman, "EVERLASTING D8.16 – White Paper 13," EU Project, Tech. Rep., 08 2020. → Cited on pages 1, 7, 37, 61, 62, and 73.

[12] J. M. Green, B. Hartman, and P. F. Glowacki, "A System-based View of the Standards and Certification Landscape for Electric Vehicles," *World Electric Vehicle Journal*, vol. 8, no. 2, pp. 564–575, 2016. [Online]. Available: https://www.mdpi.com/2032-6653/8/2/564 → Cited on pages 1, 16, and 18.

[13] L. C. Casals, B. Amante García, and C. Canal, "Second life batteries lifespan: Rest of useful life and environmental analysis," *Journal of Environmental Management*, vol. 232, pp. 354–363, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301479718313124 → Cited on pages 2, 7, and 18.

[14] S. Bobba, F. Mathieux, and G. A. Blengini, "How will second-use of batteries affect stocks and flows in the EU? A model for traction Li-ion batteries," *Resources, Conservation and Recycling*, vol. 145, pp. 279–291, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921344919300795 → Cited on pages 2 and 18.

[15] "Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL concerning batteries and waste batteries, repealing Directive 2006/66/EC and amending Regulation (EU) No 2019/1020," https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52020PC0798, 2020, accessed: 16.01.2023. → Cited on pages 2, 8, 19, 41, and 49.

[16] V. Halleux, "EU Legislation. New EU regulatory framework for batteries – Setting sustainability requirements," Members' Research Service, Tech. Rep., 2022. → Cited on pages 2 and 19.

[17] "Battery Passport," https://www.globalbattery.org/battery-passport/, 2022, accessed: 07.03.2023. → Cited on pages 2 and 19.

[18] J. Jiang and C. Zhang, *Fundamentals and Applications of Lithium-ion Batteries in Electric Drive Vehicles.* John Wiley & Sons Singapore Pte Ltd., 05 2015. → Cited on pages 2 and 13.

[19] W. Sung and C. B. Shin, "Electrochemical model of a lithium-ion battery implemented into an automotive battery management system," *Computers & Chemical Engineering*, vol. 76, pp. 87–97, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135415000526 → Cited on pages 2 and 13.

[20] B. Balasingam, M. Ahmed, and K. Pattipati, "Battery Management Systems—Challenges and Some Solutions," *Energies*, vol. 13, no. 11, 2020. [Online]. Available: https://www.mdpi.com/1996-1073/13/11/2825 → Cited on pages 2 and 3.

[21] F. A. Rincon Vija, S. Cregut, G. Z. Papadopoulos, and N. Montavont, "From Wired to Wireless BMS in Electric Vehicles," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, 2021, pp. 255–262. → Cited on pages 2, 31, and 41.

[22] A. Samanta and S. S. Williamson, "A Survey of Wireless Battery Management System: Topology, Emerging Trends, and Challenges," *Electronics*, vol. 10, no. 18, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/18/2193 → Cited on pages 2, 4, 5, 10, 15, 27, 30, 32, 41, 59, and 62.

[23] Renesas, "Wireless Battery Management System," https://www.renesas.com/us/en/application/automotive/electrified-drivetrain-xev/wireless-battery-management-system, 2023, Accessed: 2023-05-18. → Cited on pages 2 and 30.

[24] Franz Dugand, "CEVA - Automotive Battery Management System Using Bluetooth LE," https://www.ceva-dsp.com/ourblog/automotive-battery-management-system-using-bluetooth-le/, 2023, Accessed: 2023-05-18. → Cited on pages 2 and 30.

[25] M. S. Hossain Lipu, M. S. Miah, S. Ansari, S. B. Wali, T. Jamal, R. M. Elavarasan, S. Kumar, M. M. Naushad Ali, M. R. Sarker, A. Aljanad, and N. M. L. Tan, "Smart Battery Management Technology in Electric Vehicle Applications: Analytical and Technical Assessment toward Emerging Future Directions," *Batteries*, vol. 8, no. 11, 2022. [Online]. Available: https://www.mdpi.com/2313-0105/8/11/219 → Cited on pages 2, 7, 18, and 33.

[26] H. A. Gabbar, A. M. Othman, and M. R. Abdussami, "Review of Battery Management Systems (BMS) Development and Industrial Standards," *Technologies*, vol. 9, no. 2, 2021. [Online]. Available: https://www.mdpi.com/2227-7080/9/2/28 → Cited on pages 2, 3, 7, 13, 15, 16, 59, and 61.

[27] A. K. M. A. Habib, M. K. Hasan, G. F. Issa, D. Singh, S. Islam, and T. M. Ghazal, "Lithium-Ion Battery Management System for Electric Vehicles: Constraints, Challenges, and Recommendations," *Batteries*, vol. 9, no. 3, 2023. [Online]. Available: https://www.mdpi.com/2313-0105/9/3/152 → Cited on pages 2 and 14.

[28] A. Brighente, M. Conti, D. Donadel, R. Poovendran, F. Turrin, and J. Zhou, "Electric Vehicles Security and Privacy: Challenges, Solutions, and Future Needs," 2023. → Cited on pages 2, 33, 60, and 61.

[29] M. Cheah and R. Stoker, "Cybersecurity of Battery Management Systems," *HM TR series*, vol. 10, no. 3, p. 8, 2019. → Cited on pages 2, 6, 43, 44, and 60.

[30] A. Khalid, A. Sundararajan, A. Hernandez, and A. I. Sarwat, "FACTS Approach to Address Cybersecurity Issues in Electric Vehicle Battery Systems," in *2019 IEEE Technology & Engineering Management Conference (TEMSCON)*, 2019, pp. 1–6. → Cited on pages 2, 6, 32, 44, 60, and 61.

[31] Y. Mo, T. H.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–Physical Security of a Smart Grid Infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, 2012. → Cited on page 2.

[32] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber–Physical System Security for the Electric Power Grid," *Proc. of the IEEE*, vol. 100, 2012. → Cited on page 2.

[33] A. T. Elsayed, C. R. Lashway, and O. A. Mohammed, "Advanced Battery Management and Diagnostic System for Smart Grid Infrastructure," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 897–905, 2016. → Cited on pages 3 and 13.

[34] K. Liu, K. Li, Q. Peng, and C. Zhang, "A brief review on key technologies in the battery management system of electric vehicles," *Frontiers of Mechanical Engineering*, vol. 14, no. 1, p. 47, 2019. [Online]. Available: https://journal.hep.com.cn/fme/EN/abstract/article_21858.shtml → Cited on page 3.

[35] R. Xiong, J. Cao, Q. Yu, H. He, and F. Sun, "Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles," *IEEE Access*, vol. 6, pp. 1832–1843, 2018. → Cited on page 3.

[36] X. Hu, F. Feng, K. Liu, L. Zhang, J. Xie, and B. Liu, "State estimation for advanced battery management: Key challenges and future trends," *Renewable and Sustainable Energy Reviews*, vol. 114, 2019. [Online]. Available: https://doi.org/10.1016/j.rser.2019.109334 → Cited on page 3.

[37] D. Alonso, O. Opalko, and K. Dostert, "Physical layer performance analysis of a wireless data transmission approach for automotive lithium-ion batteries," in *2015 IEEE Vehicular Networking Conference (VNC)*, 2015, pp. 235–242. → Cited on pages 4 and 5.

[38] M. Lee, J. Lee, I. Lee, J. Lee, and A. Chon, "Wireless Battery Management System," in *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*, 2013, pp. 1–5. → Cited on pages 4 and 29.

[39] J. Farmer, J. Chang, J. Zumstein, J. Kotovsky, E. Zhang, A. Dobley, G. Moore, F. Puglia, S. Osswald, K. Wolf, J. Kaschmitter, S. Eaves, and T. Bandhauer, "Wireless Battery Management System for Safe High-Capacity Li-Ion Energy Storage," Lawrence Livermore National Laboratory, Tech. Rep., 01 2013. [Online]. Available: https://www.osti.gov/servlets/purl/1132034 → Cited on page 4.

[40] T. Vogt, "Wired vs. Wireless Communications in EV Battery Management," Texas Instruments, Tech. Rep., 10 2020. → Cited on pages 4 and 5.

[41] T. Kim, D. Makwana, A. Adhikaree, J. S. Vagdoda, and Y. Lee, "Cloud-Based Battery Condition Monitoring and Fault Diagnosis Platform for Large-Scale Lithium-Ion Battery Energy Storage Systems," *Energies*, vol. 11, no. 1, 2018. [Online]. Available: https://www.mdpi.com/1996-1073/11/1/125 → Cited on pages 4, 14, 34, and 43.

[42] O. Aiello, "Electromagnetic Susceptibility of Battery Management Systems' ICs for Electric Vehicles: Experimental Study," *Electronics*, vol. 9, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/3/510 → Cited on page 4.

[43] T. Gherman, M. Ricco, J. Meng, R. Teodorescu, and D. Petreus, "Smart Integrated Charger with Wireless BMS for EVs," in *IECON - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018. → Cited on pages 5 and 31.

[44] G. De Maso-Gentile, A. Bacà, L. Ambrosini, S. Orcioni, and M. Conti, "Design of CAN to Bluetooth gateway for a Battery Management System," in *12th International Workshop on Intelligent Solutions in Embedded Systems (WISES)*, 2015, pp. 171–175. → Cited on pages 5 and 31.

[45] C. Shell, J. Henderson, H. Verra, and J. Dyer, "Implementation of a Wireless Battery Management System (WBMS)," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 2015, pp. 1954–1959. → Cited on pages 5 and 31.

[46] A. Rahman, M. Rahman, and M. Rashid, "Wireless Battery Management System of Electric Transport," *IOP Conference Series: Materials Science and Engineering*, vol. 260, nov 2017. [Online]. Available: https://doi.org/10.1088/1757-899x/260/1/012029 → Cited on pages 5 and 32.

[47] K. Lounis and M. Zulkernine, "Attacks and Defenses in Short-Range Wireless Technologies for IoT," *IEEE Access*, vol. 8, 2020. → Cited on page 5.

[48] S. Sripad, S. Kulandaivel, V. Pande, V. Sekar, and V. Viswanathan, "Vulnerabilities of Electric Vehicle Battery Packs to Cyberattacks," *ArXiv*, 2017. [Online]. Available: http://arxiv.org/abs/1711.04822 → Cited on pages 5, 32, 43, 44, 60, and 61.

[49] S. Kumbhar, T. Faika, D. Makwana, T. Kim, and Y. Lee, "Cybersecurity for Battery Management Systems in Cyber-Physical Environments," in *2018 IEEE Transportation Electrification Conference and Expo (ITEC)*, 2018, pp. 934–938. → Cited on pages 5, 6, 8, 32, 44, 60, and 62.

[50] P. Sun, R. Bisschop, H. Niu, and X. Huang, "A Review of Battery Fires in Electric Vehicles," *Fire Technology*, pp. 1–50, 01 2020. [Online]. Available: https://doi.org/10.1007/s10694-019-00944-3 → Cited on pages 6 and 13.

[51] A. B. Lopez, K. Vatanparvar, A. P. Deb Nath, S. Yang, S. Bhunia, and M. A. Al Faruque, "A Security Perspective on Battery Systems of the Internet of Things," *Journal of Hardware and Systems Security*, 2017. → Cited on pages 6, 32, 43, 60, and 61.

[52] P. R. Babu, B. Palaniswamy, A. G. Reddy, V. Odelu, and H. S. Kim, "A survey on security challenges and protocols of electric vehicle dynamic charging system," *Security and Privacy*, vol. 5, no. 3, p. e210, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spy2.210 → Cited on pages 6, 33, and 60.

[53] A. Fuchs, D. Kern, C. Krauß, and M. Zhdanova, "Securing Electric Vehicle Charging Systems Through Component Binding," in *Computer Safety, Reliability, and Security*, 2020, pp. 387–401. → Cited on pages 6, 33, 36, 66, 84, and 99.

[54] Kyusk Han and Andre Weimerskirch and Kang G. Shin, "Automotive Cybersecurity for In-Vehicle Communication," IQT Quartely, Technical Paper, 2014. [Online]. Available: https://www.iqpc.com/media/1001748/37529.pdf → Cited on pages 6 and 36.

[55] L. Zhou, L. He, Y. Zheng, X. Lai, M. Ouyang, and L. Lu, "Massive battery pack data compression and reconstruction using a frequency division model in battery management systems," *Journal of Energy Storage*, vol. 28, p. 101252, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X19312587 → Cited on pages 7, 8, and 35.

[56] W. Li, M. Rentemeister, J. Badeda, D. Jöst, D. Schulte, and D. U. Sauer, "Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation," *Journal of Energy Storage*, vol. 30, p. 101557, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X20308495 → Cited on pages 8, 14, 18, and 34.

[57] S. Yang, Z. Zhang, R. Cao, M. Wang, H. Cheng, L. Zhang, Y. Jiang, Y. Li, B. Chen, H. Ling, Y. Lian, B. Wu, and X. Liu, "Implementation for a cloud battery management system based on the CHAIN framework," *Energy and AI*, vol. 5, p. 100088, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666546821000422 → Cited on pages 8, 14, 18, and 34.

[58] M.-K. Tran, S. Panchal, T. D. Khang, K. Panchal, R. Fraser, and M. Fowler, "Concept Review of a Cloud-Based Smart Battery Management System for Lithium-Ion Batteries: Feasibility, Logistics, and Functionality," *Batteries*, vol. 8, no. 2, 2022. [Online]. Available: https://www.mdpi.com/2313-0105/8/2/19 → Cited on pages 8, 13, 14, 18, and 33.

[59] K. Li, P. Zhou, Y. Lu, X. Han, X. Li, and Y. Zheng, "Battery life estimation based on cloud data for electric vehicles," *Journal of Power Sources*, vol. 468, p. 228192, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037877532030495X → Cited on pages 8, 18, 33, and 34.

[60] J. Neubauer, E. Wood, and A. Pesaran, "A Second Life for Electric Vehicle Batteries: Answering Questions on Battery Degradation and Value," *SAE International Journal of Materials and Manufacturing*, vol. 8, 01 2015. [Online]. Available: https://www.nrel.gov/docs/fy15osti/63524.pdf → Cited on pages 8, 19, and 41.

[61] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles," *IEEE Industrial Electronics Magazine*, vol. 7, 2013. → Cited on page 13.

[62] Y. Xing, E. W. M. Ma, K. L. Tsui, and M. Pecht, "Battery Management Systems in Electric and Hybrid Vehicles," *Energies*, vol. 4, no. 11, pp. 1840–1857, 2011. [Online]. Available: https://www.mdpi.com/1996-1073/4/11/1840 → Cited on page 13.

[63] M. Ouyang, D. Ren, L. Lu, J. Li, X. Feng, X. Han, and G. Liu, "Overcharge-induced capacity fading analysis for large format lithium-ion batteries with LiyNi1/3Co1/3Mn1/3O2+LiyMn2O4 composite cathode," *Journal of Power Sources*, vol. 279, pp. 626–635, 2015, 9th International Conference on Lead-Acid Batteries – LABAT 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037877531500052X → Cited on page 13.

[64] M.-K. Tran, S. Sherman, E. Samadani, R. Vrolyk, D. Wong, M. Lowery, and M. Fowler, "Environmental and Economic Benefits of a Battery Electric Vehicle Powertrain with a Zinc–Air Range Extender in the Transition to Electric Vehicles," *Vehicles*, vol. 2, no. 3, pp. 398–412, 2020. [Online]. Available: https://www.mdpi.com/2624-8921/2/3/21 → Cited on page 13.

[65] K. Friansa, I. N. Haq, B. M. Santi, D. Kurniadi, E. Leksono, and B. Yuliarto, "Development of Battery Monitoring System in Smart Microgrid Based on Internet of Things (IoT)," *Procedia Engineering*, vol. 170, pp. 482–487, 2017, engineering Physics International Conference 2016 – EPIC 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705817311931 → Cited on pages 14 and 34.

[66] H. M. Canilang, A. Caliwag, and W. Lim, "Design of Modular BMS and Real-Time Practical Implementation for Electric Motorcycle Application," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. PP, 07 2021. → Cited on page 15.

[67] A. Reindl, H. Meier, and M. Niemetz, “Scalable, Decentralized Battery Management System Based on Self-organizing Nodes,” in *Architecture of Computing Systems – ARCS 2020*, A. Brinkmann, W. Karl, S. Lankes, S. Tomforde, T. Pionteck, and C. Trinitis, Eds. Springer International Publishing, 2020, pp. 171–184. → Cited on pages 15, 61, and 73.

[68] Z. Honglei, Z. Wu, W. Dalong, and S. Jiayao, “Design and Implementation of Distributed Battery Management System,” *Advanced Materials Research*, vol. 608-609, p. 1039, 12 2012. → Cited on pages 15 and 16.

[69] ISO 6469-1:2019, “Electrically propelled road vehicles — Safety specifications — Part 1: Rechargeable energy storage system (RESS),” International Organization for Standardization, Standard, 2019. → Cited on page 16.

[70] ISO 6469-3:2021, “Electrically propelled road vehicles — Safety specifications — Part 3: Electrical safety,” International Organization for Standardization, Standard, 2021. → Cited on page 16.

[71] SAE J2288_200806, “Life Cycle Testing of Electric Vehicle Battery Modules,” Society of Automotive Engineers, Standard, 2008. → Cited on page 16.

[72] IEEE Std 1679.1-2017, “IEEE Guide for the Characterization and Evaluation of Lithium-Based Batteries in Stationary Applications,” Institute of Electrical and Electronics Engineers, Standard, 2018. → Cited on page 16.

[73] B. Canis and D. Randall, “"Black Boxes" in Passenger Vehicles: Policy Issues,” University of North Texas Libraries, Tech. Rep., 07 2014. [Online]. Available: https://digital.library.unt.edu/ark:/67531/metadc462832 → Cited on page 16.

[74] K. Chae, D. Kim, S. Jung, J. Choi, and S. Jung, “Evidence Collecting System from Car Black Boxes,” in *2010 7th IEEE Consumer Communications and Networking Conference*, 2010, pp. 1–2. → Cited on page 16.

[75] G. R. Hartung, “Advanced Cryptographic Techniques for Protecting Log Data,” Ph.D. dissertation, Karlsruher Institut für Technologie (KIT), 2020, 46.12.03; LK 01. → Cited on pages 17, 34, and 108.

[76] H. Mansor, K. Markantonakis, R. N. Akram, K. Mayes, and I. Gurulian, “Log Your Car: The Non-invasive Vehicle Forensics,” in *2016 IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 974–982. → Cited on pages 17 and 34.

[77] B. Furht and A. Escalante, *Handbook of Cloud Computing*, 1st ed. Springer Publishing Company, Incorporated, 2010. → Cited on page 18.

[78] J. Wu, X. Liu, J. Meng, and M. Lin, “Cloud-to-edge based state of health estimation method for Lithium-ion battery in distributed energy storage system,” *Journal of Energy Storage*, vol. 41, p. 102974, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X21006885 → Cited on pages 18 and 34.

[79] F. Basic, C. Seifert, C. Steger, and R. Kofler, “Secure Data Acquisition for Battery Management Systems,” in *26th Euromicro Conference Series on Digital System Design (DSD)*, 2023, in Press. → Cited on page 18.

[80] T. Tanizawa, T. Suzumiya, and K. Ikeda, "Cloud-connected Battery Management System Supporting e-Mobility," *Fujitsu Science Technical Journal*, vol. 51, pp. 27–35, 2015. → Cited on pages 18 and 34.

[81] CSS Electronics, "Electric Vehicle Data Logger - Cloud Battery/BMS Telematics," https://www.csselectronics.com/pages/electric-vehicle-data-logger-cloud-battery-telematics, 2023, Accessed: 2023-02-27. → Cited on page 18.

[82] Bosch Mobility Solutions, "Battery in the cloud," https://www.bosch-mobility-solutions.com/en/solutions/software-and-services/battery-in-the-cloud/battery-in-the-cloud/, 2023, Accessed: 2023-02-27. → Cited on page 18.

[83] NXP Semiconductors, "AI-Powered Cloud-Connected Battery Management System for Electric Vehicles," https://www.nxp.com/company/about-nxp/ai-powered-cloud-connected-battery-management-system-for-electric-vehicles:NW-NXP-AI-POWERED-CLOUD-CONNECTED-BATTERY, 2022, Accessed: 2023-02-27. → Cited on page 18.

[84] R. Faria, P. Marques, R. Garcia, P. Moura, F. Freire, J. Delgado, and A. T. de Almeida, "Primary and secondary use of electric mobility batteries from a life cycle perspective," *Journal of Power Sources*, vol. 262, pp. 169–177, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775314004157 → Cited on page 18.

[85] N. Nicholas, "End-of-life Electric vehicle batteries: Recycling or second-life?" https://www.smart-energy.com/features-analysis/end-of-life-electric-vehicle-batteries-recycling-or-second-life/, 2020, accessed: 26.11.2022. → Cited on page 18.

[86] E. Wood, M. Alexander, and T. H. Bradley, "Investigation of battery end-of-life conditions for plug-in hybrid electric vehicles," *Journal of Power Sources*, vol. 196, no. 11, pp. 5147–5154, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037877531100379X → Cited on page 18.

[87] K. Smith, Y. Shi, and S. Santhanagopalan, "Degradation mechanisms and lifetime prediction for lithium-ion batteries — A control perspective," in *2015 American Control Conference (ACC)*, 2015, pp. 728–730. → Cited on page 19.

[88] K. Smith, "Battery Lifespan," https://www.nrel.gov/transportation/battery-lifespan.html, 2014, Accessed: 2023-03-02. → Cited on page 19.

[89] J. Neubauer and E. Wood, "Thru-life impacts of driver aggression, climate, cabin thermal management, and battery thermal management on battery electric vehicle utility," *Journal of Power Sources*, vol. 259, pp. 262–275, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775314002766 → Cited on page 19.

[90] ——, "The impact of range anxiety and home, workplace, and public charging infrastructure on simulated battery electric vehicle lifetime utility," *Journal of Power Sources*, vol. 257, pp. 12–20, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775314000998 → Cited on page 19.

[91] "EU Batteries Regulation: Four Position Paper," https://eeb.org/library/eu-batteries-regulation-four-position-paper/, 2022, accessed: 16.01.2023. → Cited on page 19.

[92] L. Gressl, "Towards Security-Aware Design Space Exploration for Embedded Systems," Ph.D. dissertation, Graz University of Technology (TU Graz), 2020. → Cited on pages 19, 20, 23, and 24.

[93] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004. → Cited on page 19.

[94] S. Samonas and D. Coss, "The CIA Strikes Back: Redefining Confidentiality, Integrity and Availability in Security," *Journal of Information System Security*, vol. 10, p. 21–45, 2021. [Online]. Available: https://www.jissec.org/Contents/V10/N3/V10N3-Samonas.html → Cited on page 19.

[95] R. Anderson, *Security Engineering - A Guide to Building Dependable Distributed Systems.* Wiley, 2020. → Cited on pages 20, 21, 22, and 58.

[96] N. Ferguson and B. Schneier, *Practical Cryptography.* Wiley, 2003. → Cited on pages 20 and 58.

[97] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback, and J. Dray, "Advanced Encryption Standard (AES)," 2001-11-26 2001. → Cited on page 21.

[98] K. P. Singh and S. Dod, "An Efficient Hardware design and Implementation of Advanced Encryption Standard (AES) Algorithm," *IACR Cryptol. ePrint Arch.*, 2016. → Cited on page 21.

[99] M. A. Jimale, M. R. Z'aba, M. L. B. M. Kiah, M. Y. I. Idris, N. Jamil, M. S. Mohamad, and M. S. Rohmad, "Authenticated Encryption Schemes: A Systematic Review," *IEEE Access*, vol. 10, pp. 14 739–14 766, 2022. → Cited on page 21.

[100] J. Jonsson, "On the Security of CTR + CBC-MAC," in *Selected Areas in Cryptography*, K. Nyberg and H. Heys, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 76–93. → Cited on page 21.

[101] M. Bellare, R. Canetti, and H. Krawczyk, "Message Authentication using Hash Functions— The HMAC Construction," *RSA Laboratories' CryptoBytes*, vol. 2, no. 1, p. 12–15, 1996. [Online]. Available: https://cseweb.ucsd.edu/~mihir/papers/hmac-cb.pdf → Cited on page 21.

[102] T. Iwata and K. Kurosawa, "OMAC: One-Key CBC MAC," Cryptology ePrint Archive, Paper 2002/180, 2002, https://eprint.iacr.org/2002/180. [Online]. Available: https://eprint.iacr.org/2002/180 → Cited on page 21.

[103] P. Gutmann, "Encrypt-then-MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)," RFC 7366, Sep. 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7366 → Cited on page 21.

[104] H. Krawczyk, "The order of encryption and authentication for protecting communications (Or: how secure is SSL?)," Cryptology ePrint Archive, Paper 2001/045, 2001, https://eprint.iacr.org/2001/045. [Online]. Available: https://eprint.iacr.org/2001/045 → Cited on page 21.

[105] W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976. → Cited on page 22.

[106] H. Krawczyk, “Cryptographic Extraction and Key Derivation: The HKDF Scheme,” in *Advances in Cryptology – CRYPTO 2010*, T. Rabin, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 631–648. → Cited on page 22.

[107] S. Mangard, N. Pramstaller, and E. Oswald, “Successfully Attacking Masked AES Hardware Implementations,” in *Cryptographic Hardware and Embedded Systems – CHES 2005*, J. R. Rao and B. Sunar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 157–171. → Cited on page 22.

[108] N. Santos, H. Raj, S. Saroiu, and A. Wolman, “Using ARM Trustzone to Build a Trusted Language Runtime for Mobile Applications,” in *Proceedings of the 19th international conference on Architectural support for programming languages and operating systems*, ser. ASPLOS ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 67–80. [Online]. Available: https://doi.org/10.1145/2541940.2541949 → Cited on page 22.

[109] M. Sabt, M. Achemlal, and A. Bouabdallah, “The Dual-Execution-Environment Approach: Analysis and Comparative Evaluation,” in *ICT Systems Security and Privacy Protection*, H. Federrath and D. Gollmann, Eds. Cham: Springer International Publishing, 2015, pp. 557–570. → Cited on page 22.

[110] A. J. Cabrera-Gutiérrez, E. Castillo, A. Escobar-Molero, J. A. Álvarez Bermejo, D. P. Morales, and L. Parrilla, “Integration of Hardware Security Modules and Permissioned Blockchain in Industrial IoT Networks,” *IEEE Access*, vol. 10, pp. 114 331–114 345, 2022. → Cited on page 22.

[111] A. Höller, N. Druml, C. Kreiner, C. Steger, and T. Felicijan, “Hardware/software co-design of elliptic-curve cryptography for resource-constrained applications,” in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2014, pp. 1–6. → Cited on page 22.

[112] S. Myagmar, A. Lee J., and W. Yurcik, “Threat Modeling as a Basis for Security Requirements,” *in SREIS*, 2005. → Cited on pages 23 and 91.

[113] A. Shostack, *Threat Modeling: Designing for Security.* Wiley, 2014. → Cited on pages 23 and 91.

[114] M. Howard and D. E. Leblanc, *Writing Secure Code*, 2nd ed. Microsoft Press, 2002. → Cited on pages 23 and 91.

[115] G. Macher, E. Armengaud, E. Brenner, and C. Kreiner, “Threat and Risk Assessment Methodologies in the Automotive Domain,” *Procedia Computer Science*, vol. 83, pp. 1288–1294, 2016, the 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Affiliated Workshops. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050916303015 → Cited on pages 23 and 60.

[116] Z. Ma and C. Schmittner, “Threat Modeling for Automotive Security Analysis,” in *Future Generation Information Technology*, 2016, pp. 333–339, the 9th International Conference on Security Technology. → Cited on page 23.

[117] J. Dobaj, C. Schmittner, M. Krisper, and G. Macher, “Towards Integrated Quantitative Security and Safety Risk Assessment,” in *Computer Safety, Reliability, and Security*, A. Romanovsky, E. Troubitsyna, I. Gashi, E. Schoitsch, and F. Bitsch, Eds. Cham: Springer International Publishing, 2019, pp. 102–116. → Cited on page 23.

[118] C. Shan, B. Jiang, J. Xue, F. Guan, and N. Xiao, "An Approach for Internal Network Security Metric Based on Attack Probability," *Security and Communication Networks*, vol. 2018, pp. 1–11, 04 2018. → Cited on page 23.

[119] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, p. 18–36, feb 1990. [Online]. Available: https://doi.org/10.1145/77648.77649 → Cited on pages 23 and 93.

[120] C. J. F. Cremers, "The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols," in *Computer Aided Verification*, A. Gupta and S. Malik, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 414–418. → Cited on page 23.

[121] C. Cremers, M. Horvat, J. Hoyland, S. Scott, and T. van der Merwe, "A Comprehensive Symbolic Analysis of TLS 1.3," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1773–1788. [Online]. Available: https://doi.org/10.1145/3133956.3134063 → Cited on page 23.

[122] S. Faily, R. Scandariato, A. Shostack, L. Sion, and D. Ki-Aries, "Contextualisation of Data Flow Diagrams for Security Analysis," in *Graphical Models for Security*, H. Eades III and O. Gadyatskaya, Eds. Cham: Springer International Publishing, 2020, pp. 186–197. → Cited on page 23.

[123] T. Ulz, T. Pieber, C. Steger, C. Lesjak, H. Bock, and R. Matischek, "SECURECONFIG: NFC and QR-code based hybrid approach for smart sensor configuration," in *2017 IEEE International Conference on RFID (RFID)*, 2017, pp. 41–46. → Cited on pages 23, 40, 42, and 49.

[124] P. Johnson, R. Lagerström, M. Ekstedt, and U. Franke, "Can the Common Vulnerability Scoring System be Trusted? A Bayesian Analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, pp. 1002–1015, 2018. → Cited on page 23.

[125] Common Criteria, "Common Criteria for Information Technology Security Evaluation Part 1 - 3. Version 3.1 Revision 5," International Organization for Standardization, Standard, 2018. → Cited on pages 24 and 73.

[126] ISO/SAE 21434:2021, "Road vehicles — Cybersecurity engineering," International Organization for Standardization, Standard, 2021. → Cited on pages 24 and 73.

[127] ISO 15118-20:2022, "Road vehicles — Vehicle to grid communication interface," International Organization for Standardization, Standard, 2022. → Cited on pages 24 and 74.

[128] ISO 26262-1:2018, "Road vehicles — Functional safety — Part 1: Vocabulary," International Organization for Standardization, Standard, 2018. → Cited on page 24.

[129] "OWASP Threat Modeling Proccess," https://owasp.org/www-community/Threat_Modeling_Process, 2023, accessed: 13.05.2023. → Cited on page 24.

[130] F. Pollicino, D. Stabili, L. Ferretti, and M. Marchetti, "An experimental analysis of ECQV implicit certificates performance in VANETs," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, 2020, pp. 1–6. → Cited on pages 24, 38, 81, and 98.

[131] D. A. Ha, K. T. Nguyen, and J. K. Zao, “Efficient Authentication of Resource-Constrained IoT Devices Based on ECQV Implicit Certificates and Datagram Transport Layer Security Protocol,” in *Proceedings of the 7th Symposium on Information and Communication Technology*, ser. SoICT ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 173–179. [Online]. Available: https://doi.org/10.1145/3011077.3011108 → Cited on pages 24 and 39.

[132] C.-S. Park, “A Secure and Efficient ECQV Implicit Certificate Issuance Protocol for the Internet of Things Applications,” *IEEE Sensors Journal*, vol. 17, no. 7, pp. 2215–2223, 2017. → Cited on pages 25 and 38.

[133] M. Campagna, *SEC 4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)*, Certicom Corp., 2013. [Online]. Available: https://www.secg.org/sec4-1.0.pdf → Cited on pages 25 and 63.

[134] D. Wagner, “A Generalized Birthday Problem,” in *Advances in Cryptology — CRYPTO 2002*, M. Yung, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 288–304. → Cited on page 25.

[135] M. Erb, “Design and Implementation of an Advanced Near-Field Communication Interoperability Test Automation System,” Ph.D. dissertation, Graz University of Technology (TU Graz), 2021. → Cited on page 26.

[136] T. Ulz, “Towards Trustworthy Smart Sensors,” Ph.D. dissertation, Graz University of Technology (TU Graz), 2019. → Cited on pages 26 and 28.

[137] V. Coskun, B. Ozdenizci, and K. Ok, “The Survey on Near Field Communication,” *Sensors (Basel)*, vol. 15, no. 6, jun 2015. [Online]. Available: https://doi.org/10.3390/s150613348 → Cited on pages 26, 27, and 52.

[138] M. Bouklachi, M. Biancheri-Astier, A. Diet, and Y. L. Bihan, “Energy Harvesting of a NFC Flexible Patch for Medical Applications,” in *2019 IEEE Wireless Power Transfer Conference (WPTC)*, 2019, pp. 249–252. → Cited on pages 26, 27, and 52.

[139] ISO/IEC 14443-4:2018, “Cards and security devices for personal identification — Contactless proximity objects — Part 4: Transmission protocol,” International Organization for Standardization, Standard, 2018. → Cited on page 27.

[140] ISO/IEC 15693-2:2019, “Cards and security devices for personal identification — Contactless vicinity objects — Part 2: Air interface and initialization,” International Organization for Standardization, Standard, 2019. → Cited on page 27.

[141] ISO/IEC 18092:2013, “Information Technology—Telecommunications and Information Exchange Between Systems—Near Field Communication—Interface and Protocol (NFCIP-1),” International Organization for Standardization, Standard, 2013. → Cited on page 27.

[142] ISO/IEC 21481:2021, “Information technology — Telecommunications and information exchange between systems — Near field communication interface and protocol 2 (NFCIP-2),” International Organization for Standardization, Standard, 2021. → Cited on page 27.

[143] “NFC Forum,” https://nfc-forum.org/, 2023, accessed: 28.04.2023. → Cited on page 27.

[144] F. Basic, C. R. Laube, C. Steger, and R. Kofler, “A Novel Secure NFC-based Approach for BMS Monitoring and Diagnostic Readout,” in *IEEE International Conference on RFID (RFID)*, 2022, pp. 23–28. → Cited on pages 27 and 42.

[145] J. Ondrus and Y. Pigneur, “An Assessment of NFC for Future Mobile Payment Systems,” in *International Conference on the Management of Mobile Business (ICMB 2007)*, 2007, pp. 43–43. → Cited on page 27.

[146] R. Widmann, S. Grunberger, B. Stadlmann, and J. Langer, “System Integration of NFC Ticketing into an Existing Public Transport Infrastructure,” in *2012 4th International Workshop on Near Field Communication*, 2012, pp. 13–18. → Cited on page 27.

[147] D. M. Monteiro, J. J. P. C. Rodrigues, J. Lloret, and S. Sendra, “A hybrid NFC–Bluetooth secure protocol for Credit Transfer among mobile phones,” *Security and Communication Networks*, vol. 7, no. 2, pp. 325–337, 2014. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.732 → Cited on page 27.

[148] N. B. Thorat and C. Laulkar, “Survey on Security Threats and Solutions for Near Field Communication,” *International Journal of Research in Engineering and Technology*, vol. 03, pp. 291–295, 2014. → Cited on pages 27 and 28.

[149] F. Basic, M. Gaertner, and C. Steger, “Secure and Trustworthy NFC-Based Sensor Readout for Battery Packs in Battery Management Systems,” *IEEE Journal of Radio Frequency Identification*, vol. 6, 2022. → Cited on pages 27 and 42.

[150] S. Klee, A. Roussos, M. Maass, and M. Hollick, “NFCGate: Opening the Door for NFC Security Research with a Smartphone-Based Toolkit,” in *14th USENIX Workshop on Offensive Technologies (WOOT 20)*. USENIX Association, Aug. 2020. [Online]. Available: https://www.usenix.org/conference/woot20/presentation/klee → Cited on page 27.

[151] S. Plosz, A. Farshad, M. Tauber, C. Lesjak, T. Ruprechter, and N. Pereira, “Security Vulnerabilities and Risks in Industrial Usage of Wireless Communication,” in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014, pp. 1–8. → Cited on pages 28, 40, 43, and 44.

[152] C. H. Chen, I. C. Lin, and C. C. Yang, “NFC Attacks Analysis and Survey,” in *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2014, pp. 458–462. → Cited on pages 28 and 44.

[153] E. Haselsteiner and K. Breitfuß, “Security in Near Field Communication (NFC) Strengths and Weaknesses,” in *Workshop on RFID security*, 2006. → Cited on pages 28 and 44.

[154] N. A. Chattha, “NFC — Vulnerabilities and Defense,” in *2014 Conference on Information Assurance and Cyber Security (CIACS)*, 2014, pp. 35–38. → Cited on pages 28 and 43.

[155] A. Goikoetxea Yanci, “Smart card security,” Ph.D. dissertation, University of Glasgow, 2012. → Cited on page 28.

[156] T. Faika, T. Kim, and M. Khan, “An Internet of Things (IoT)-Based Network for Dispersed and Decentralized Wireless Battery Management Systems,” in *2018 IEEE Transportation Electrification Conference and Expo (ITEC)*, 2018, pp. 1060–1064. → Cited on page 29.

[157] P. Bansal and P. Nagaraj, "Wireless Battery Management System for Electric Vehicles," in *IEEE ITEC-India*, 2019, pp. 1–5. → Cited on page 30.

[158] Zachariah Peterson, "Wireless BMS Design and Chipset Options," https://octopart.com/blog/archives/2021/05/wireless-bms-design-and-chipset-options, 2021, Accessed: 2023-05-18. → Cited on page 30.

[159] M. Spörk, C. A. Boano, and K. Römer, "Performance and Trade-Offs of the New PHY Modes of BLE 5," in *Proceedings of the ACM MobiHoc Workshop on Pervasive Systems in the IoT Era*, ser. PERSIST-IoT '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 7–12. [Online]. Available: https://doi.org/10.1145/3331052.3332471 → Cited on page 30.

[160] F. A. Rincon Vija, S. Cregut, G. Z. Papadopoulos, and N. Montavont, "Enabling Robust Wireless Communication for BMS on Electric Vehicles," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, 2021, pp. 423–426. → Cited on page 31.

[161] X. Huang, A. B. Acharya, J. Meng, X. Sui, D.-I. Stroe, and R. Teodorescu, "Wireless Smart Battery Management System for Electric Vehicles," in *IEEE ECCE*, 2020, pp. 5620–5625. → Cited on page 31.

[162] M. Schneider, S. Ilgin, N. Jegenhorst, R. Kube, S. Püttjer, K.-R. Riemschneider, and J. Vollmer, "Automotive Battery Monitoring by Wireless Cell Sensors," in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, 2012, pp. 816–820. → Cited on page 32.

[163] A. Ahalawat, S. Adepu, and J. Gardiner, "Security Threats in Electric Vehicle Charging," in *2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2022, pp. 399–404. → Cited on pages 33 and 60.

[164] "EVITA. E-safety vehicle intrusion protected applications," https://evita-project.org/, 2011, Accessed: 08.05.2023. → Cited on pages 33, 34, and 37.

[165] S. Câmara, L. Pirmez, and L. F. R. C. Carmo, "Towards a Storage-Efficient and Categorized Secure Log Structure Scheme for Embedded Systems," in *2017 IEEE 15th International Conference on Dependable, Autonomic and Secure Computing*, 2017, pp. 519–526. → Cited on pages 35 and 108.

[166] D. Ma and G. Tsudik, "Extended Abstract: Forward-Secure Sequential Aggregate Authentication," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, 2007, pp. 86–91. → Cited on page 35.

[167] U. Sinha, A. A. Hadi, T. Faika, and T. Kim, "Blockchain-Based Communication and Data Security Framework for IoT-Enabled Micro Solar Inverters," in *2019 IEEE CyberPELS (CyberPELS)*, 2019, pp. 1–5. → Cited on page 35.

[168] A. Ali, A. Khan, M. Ahmed, and G. Jeon, "BCALS: Blockchain-Based Secure Log Management System for Cloud Computing," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 4, apr 2022. [Online]. Available: https://doi.org/10.1002/ett.4272 → Cited on pages 35 and 62.

[169] B. C. Florea and D. D. Taralunga, "Blockchain IoT for Smart Electric Vehicles Battery Management," *Sustainability*, vol. 12, no. 10, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/10/3984 → Cited on page 35.

[170] T. Faika, T. Kim, J. Ochoa, M. Khan, S.-W. Park, and C. S. Leung, “A Blockchain-Based Internet of Things (IoT) Network for Security-Enhanced Wireless Battery Management Systems,” in *2019 IEEE Industry Applications Society Annual Meeting*, 2019, pp. 1–6. → Cited on page 35.

[171] T. Kim, J. Ochoa, T. Faika, H. A. Mantooth, J. Di, Q. Li, and Y. Lee, “An Overview of Cyber-Physical Security of Battery Management Systems and Adoption of Blockchain Technology,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 10, pp. 1270–1281, 2020. → Cited on page 35.

[172] J. J. Ochoa, G. Bere, I. R. Aenugu, T. Kim, and K.-K. R. Choo, “Blockchain-as-a-Service (BaaS) for Battery Energy Storage Systems,” in *2020 IEEE Texas Power and Energy Conference (TPEC)*, 2020, pp. 1–6. → Cited on page 35.

[173] G. Bere, J. J. Ochoa, T. Kim, and I. R. Aenugu, “Blockchain-Based Firmware Security Check and Recovery for Battery Management Systems,” in *2020 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2020, pp. 262–266. → Cited on page 35.

[174] I. R. Aenugu, G. Bere, J. J. Ochoa, T. Kim, C. Lee, and J.-h. Park, “Battery Data Management and Analytics Platform Using Blockchain Technology,” in *2020 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2020, pp. 153–157. → Cited on page 35.

[175] M. Chiu, A. Goldsmith, and U. Kalabić, “Blockchain for Embedded System Accountability,” in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2021, pp. 1–5. → Cited on page 35.

[176] H. Liu, Z. Lv, Z. Song, S. Zhou, W. Liu, M. Fang, and C.-H. Wu, “Application of Blockchain Technology in Electric Vehicle Charging Piles Based on Electricity Internet of Things,” *Wireless Communications & Mobile Computing*, vol. 2022, jan 2022. [Online]. Available: https://doi.org/10.1155/2022/8533219 → Cited on page 35.

[177] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016. → Cited on page 35.

[178] Richard Soja, “Automotive Security: From Standards to Implementation,” NXP Semiconductors - Freescale, White Paper, 2014. → Cited on pages 36 and 60.

[179] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, “Evaluation of CAN Bus Security Challenges,” *Sensors*, vol. 20, no. 8, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/8/2364 → Cited on pages 36 and 60.

[180] A. Hazem and H. M. A. Fahmy, “LCAP-A Lightweight CAN Authentication Protocol for Securing In-Vehicle Networks,” in *10th Embedded Security in Cars*, 2012. → Cited on page 36.

[181] C.-W. Lin and A. Sangiovanni-Vincentelli, “Cyber-Security for the Controller Area Network (CAN) Communication Protocol,” in *2012 International Conference on Cyber Security*, 2012, pp. 1–7. → Cited on pages 36 and 60.

[182] E. R. Verheul, “Issue First Activate Later Certificates for V2X – Combining ITS efficiency with privacy,” Cryptology ePrint Archive, Paper 2016/1158, 2016, https://eprint.iacr.org/2016/1158. [Online]. Available: https://eprint.iacr.org/2016/1158 → Cited on pages 36, 38, 64, and 109.

[183] P. Mundhenk, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, and S. Chakraborty, "Lightweight Authentication for Secure Automotive Networks," in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp. 285–288. → Cited on pages 36 and 61.

[184] I. E. C. Roca, J. Wang, J. Du, and S. Wei, "A Semi-centralized Security Framework for In-Vehicle Networks," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020, pp. 1–6. → Cited on pages 36 and 61.

[185] M. Steger, M. Karner, J. Hillebrand, W. Rom, C. Boano, and K. Römer, "Generic Framework Enabling Secure and Efficient Automotive Wireless SW Updates," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–8. → Cited on page 37.

[186] M. Steger, C. A. Boano, T. Niedermayr, M. Karner, J. Hillebrand, K. Roemer, and W. Rom, "An Efficient and Secure Automotive Wireless Software Update Framework," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2181–2193, 2018. → Cited on page 37.

[187] M. Olsson, "HEAVENS - HEAling Vulnerabilities to ENhance Software Security and Safety, Executive Summary," https://www.vinnova.se/en/p/heavens-healing-vulnerabilities-to-enhance-software-security-and-safety/, Volvo Technology AB, 2016. → Cited on page 37.

[188] A. Lautenbach, M. Almgren, and T. Olovsson, "Proposing HEAVENS 2.0 – an Automotive Risk Assessment Model," in *Proceedings of the 5th ACM Computer Science in Cars Symposium*, ser. CSCS '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3488904.3493378 → Cited on page 37.

[189] "INCOBAT - Innovative Cost Efficient Management System for Next Generation High Voltage Batteries," https://www.2zeroemission.eu/research-project/incobat/, 2016, Accessed: 08.05.2023. → Cited on page 37.

[190] W. Huang, J. Lin, Q. Wang, Y. Teng, H. Wan, and W. Wang, "Certificate Transparency for ECQV Implicit Certificates," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6. → Cited on page 37.

[191] D. Püllen, N. A. Anagnostopoulos, T. Arul, and S. Katzenbeisser, "Using Implicit Certification to Efficiently Establish Authenticated Group Keys for In-Vehicle Networks," in *2019 IEEE Vehicular Networking Conference (VNC)*, 2019, pp. 1–8. → Cited on pages 37, 61, 62, and 63.

[192] G. T. Becker, "The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs," in *Cryptographic Hardware and Embedded Systems – CHES 2015*, T. Güneysu and H. Handschuh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 535–555. → Cited on pages 38 and 109.

[193] N. Wisiol, C. Mühl, N. Pirnay, P. Nguyen, M. Margraf, J.-P. Seifert, M. van Dijk, and U. Rührmair, "Splitting the Interpose PUF: A Novel Modeling Attack Strategy," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 97–120, 06 2020. → Cited on pages 38 and 109.

[194] N. Wisiol, G. T. Becker, M. Margraf, T. A. A. Soroceanu, J. Tobisch, and B. Zengin, "Breaking the Lightweight Secure PUF: Understanding the Relation of Input Transformations and Machine Learning Resistance," in *Smart Card Research and Advanced Applications*, S. Belaïd and

T. Güneysu, Eds. Cham: Springer International Publishing, 2020, pp. 40–54. → Cited on pages 38 and 109.

[195] A. M. Almuhaideb and S. S. Algothami, "ECQV-Based Lightweight Revocable Authentication Protocol for Electric Vehicle Charging," *Big Data and Cognitive Computing*, vol. 6, no. 4, 2022. [Online]. Available: https://www.mdpi.com/2504-2289/6/4/102 → Cited on page 38.

[196] P. Porambage, P. Kumar, C. Schmitt, A. Gurtov, and M. Ylianttila, "Certificate-Based Pairwise Key Establishment Protocol for Wireless Sensor Networks," in *2013 IEEE 16th International Conference on Computational Science and Engineering*, 2013, pp. 667–674. → Cited on page 38.

[197] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase Authentication Protocol for Wireless Sensor Networks in Distributed IoT Applications," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, 2014, pp. 2728–2733. → Cited on pages 38, 66, 98, 100, and 101.

[198] G. Dini and M. Tiloca, "Considerations on Security in ZigBee Networks," in *2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2010, pp. 58–65. → Cited on page 38.

[199] V. Siddhartha, G. S. Gaba, and L. Kansal, "A Lightweight Authentication Protocol using Implicit Certificates for Securing IoT Systems," *Procedia Computer Science*, vol. 167, pp. 85–96, 2020, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920306505 → Cited on page 38.

[200] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public Key Authentication and Key Agreement in IoT Devices With Minimal Airtime Consumption," *IEEE Embedded Systems Letters*, vol. 9, no. 1, pp. 1–4, 2017. → Cited on pages 39, 67, 98, and 101.

[201] D.-H. Lee and I.-Y. Lee, "A Lightweight Authentication and Key Agreement Schemes for IoT Environments," *Sensors (Basel, Switzerland)*, vol. 20, 09 2020. → Cited on page 39.

[202] Z.-Y. Liu, Y.-F. Tseng, R. Tso, P. S. Wang, and Q.-W. Su, "Extension of elliptic curve Qu–Vanstone certificates and their applications," *Journal of Information Security and Applications*, vol. 67, p. 103176, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221421262200059X → Cited on page 39.

[203] M. Q. Saeed and C. D. Walter, "A Record Composition/Decomposition attack on the NDEF Signature Record Type Definition," in *International Conference for Internet Technology and Secured Transactions*, 2011, pp. 283–287. → Cited on page 40.

[204] T. Ulz, T. Pieber, C. Steger, S. Haas, and R. Matischek, "Sneakernet on Wheels: Trustworthy NFC-based Robot to Machine Communication," in *2017 IEEE International Conference on RFID Technology & Application (RFID-TA)*, 2017, pp. 260–265. → Cited on page 40.

[205] ——, "QSNFC: Quick and Secured Near Field Communication for the Internet of Things," in *IEEE RFID*, 2018, pp. 1–8. → Cited on page 40.

[206] P. Urien and S. Piramuthu, "LLCPS and SISO: A TLS-based framework with RFID for NFC P2P retail transaction processing," in *2013 IEEE International Conference on RFID (RFID)*, 2013, pp. 152–159. → Cited on page 40.

[207] C. Lesjak, "Design and Implementation of a Secure NFC-based System for Mobile Electronic Coupons," Master's thesis, Graz University of Technology, 2013. → Cited on page 40.

[208] C.-L. Chen, Y.-Y. Chen, T.-F. Shih, and T.-M. Kuo, "An RFID Authentication and Anti-counterfeit Transaction Protocol," in *2012 International Symposium on Computer, Consumer and Control*, 2012, pp. 419–422. → Cited on page 40.

[209] W. Sun, X. Zhu, T. Zhou, Y. Su, and B. Mo, "Application of Blockchain and RFID in Anti-counterfeiting Traceability of Liquor," in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, 2019, pp. 1248–1251. → Cited on page 40.

[210] G. Khalil, R. Doss, and M. Chowdhury, "A Novel RFID-Based Anti-Counterfeiting Scheme for Retail Environments," *IEEE Access*, vol. 8, pp. 47 952–47 962, 2020. → Cited on page 40.

[211] N. C. K. Yiu, "An NFC-Enabled Anti-Counterfeiting System for Wine Industry," *ArXiv*, vol. abs/1601.06372, 2016. → Cited on page 40.

[212] M. Wazid, A. K. Das, M. K. Khan, A. A.-D. Al-Ghaiheb, N. Kumar, and A. V. Vasilakos, "Secure Authentication Scheme for Medicine Anti-Counterfeiting System in IoT Environment," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1634–1646, 2017. → Cited on page 40.

[213] B. A. Alzahrani, K. Mahmood, and S. Kumari, "Lightweight Authentication Protocol for NFC Based Anti-Counterfeiting System in IoT Infrastructure," *IEEE Access*, vol. 8, pp. 76 357–76 367, 2020. → Cited on page 40.

[214] M. Gaertner, "Design and Implementation of a NFC-based Solution for Secure Battery Management Systems," Master's thesis, Graz University of Technology, 2021. → Cited on pages 41 and 86.

[215] C. Laube, "Design and Implementation of Secure NFC-based Logging for Stationary Battery Management Systems," Master's thesis, Graz University of Technology, 2022. → Cited on pages 41 and 86.

[216] F. Basic, M. Gaertner, and C. Steger, "Towards Trustworthy NFC-based Sensor Readout for Battery Packs in Battery Management Systems," in *2021 IEEE International Conference on RFID Technology and Applications (RFID-TA)*, 2021, pp. 285–288. → Cited on page 42.

[217] M. K. Hasan, M. Mahmud, A. Ahasan Habib, S. Motakabber, and S. Islam, "Review of electric vehicle energy storage and management system: Standards, issues, and challenges," *Journal of Energy Storage*, vol. 41, p. 102940, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X21006575 → Cited on page 42.

[218] "NIST NVD: CVE-2020-27524," https://nvd.nist.gov/vuln/detail/CVE-2020-27524, 2020, accessed: 02.02.2023. → Cited on page 43.

[219] "NIST NVD: CVE-2017-9212," https://nvd.nist.gov/vuln/detail/CVE-2017-9212, 2017, accessed: 16.01.2023. → Cited on page 43.

[220] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study," in *Proceedings of the 19th USENIX Conference on Security*, ser. USENIX Security'10. USA: USENIX Association, 2010, p. 21. → Cited on page 43.

[221] S. Melançon, "EV Battery Pack Deigns: An Overview," https://www.laserax.com/blog/battery-pack-design, 2022, accessed: 21.05.2023. → Cited on page 43.

[222] "E-mobility engineering: Cell-to-pack batteries," https://www.emobility-engineering.com/cell-to-pack-batteries/, 2023, accessed: 21.05.2023. → Cited on page 43.

[223] G. Belingardi and A. Scattina, "Battery Pack and Underbody: Integration in the Structure Design for Battery Electric Vehicles -Challenges and Solutions," *Vehicles*, vol. 5, no. 2, pp. 498–514, 2023. [Online]. Available: https://www.mdpi.com/2624-8921/5/2/28 → Cited on page 43.

[224] J. Zhu, X. Zhang, T. Wierzbicki, Y. Xia, and G. Chen, "Structural Designs for Electric Vehicle Battery Pack against Ground Impact," in *WCX World Congress Experience*, 04 2018. → Cited on page 43.

[225] A. Lazaro, R. Villarino, and D. Girbau, "A Survey of NFC Sensors Based on Energy Harvesting for IoT Applications," *Sensors*, vol. 18, no. 11, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/3746 → Cited on page 52.

[226] F. Basic, C. R. Laube, P. Stratznig, C. Steger, and R. Kofler, "Wireless BMS Architecture for Secure Readout in Vehicle and Second life Applications," in *8th International Conference on Smart and Sustainable Technologies*, 2023. → Cited on page 52.

[227] "Microchip - Serial EEPROM Products," https://www.microchip.com/en-us/products/memory/serial-eeprom, 2023, accessed: 21.05.2023. → Cited on page 54.

[228] D. Alonso, O. Opalko, M. Sigle, and K. Dostert, "Towards a Wireless Battery Management System: Evaluation of Antennas and Radio Channel Measurements Inside a Battery Emulator," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, 2014, pp. 1–5. → Cited on page 54.

[229] F. A. P. Petitcolas, *Kerckhoffs' Principle*. Boston, MA: Springer US, 2011, pp. 675–675. [Online]. Available: https://doi.org/10.1007/978-1-4419-5906-5_487 → Cited on pages 58 and 61.

[230] W. Diffie, P. C. Van Oorschot, and M. J. Wiener, "Authentication and Authenticated Key Exchanges," *Design, Codes and Cryptography*, vol. 2, no. 2, p. 107–125, Jun. 1992. [Online]. Available: https://doi.org/10.1007/BF00124891 → Cited on pages 58 and 66.

[231] H. Krawczyk, *Perfect Forward Secrecy*. Boston, MA: Springer US, 2005, pp. 457–458. [Online]. Available: https://doi.org/10.1007/0-387-23483-7_298 → Cited on page 61.

[232] V. Bernat. (2011) TLS & Perfect Forward Secrecy. [Online]. Available: https://vincent.bernat.ch/en/blog/2011-ssl-perfect-forward-secrecy → Cited on page 61.

[233] F. Basic, C. Steger, C. Seifert, and R. Kofler, "Trust your BMS: Designing a Lightweight Authentication Architecture for Industrial Networks," in *2022 IEEE International Conference on Industrial Technology (ICIT)*, 2022, pp. 1–6. → Cited on pages 62 and 98.

[234] F. Basic, C. Steger, and R. Kofler, "Establishing Dynamic Secure Sessions for ECQV Implicit Certificates in Embedded Systems," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2023, pp. 1–6. → Cited on pages 62 and 98.

[235] M. H. Eldefrawy, N. Pereira, and M. Gidlund, "Key Distribution Protocol for Industrial Internet of Things Without Implicit Certificates," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 906–917, 2019. → Cited on pages 63 and 93.

[236] D. R. L. Brown, M. J. Campagna, and S. A. Vanstone, "Security of ECQV-Certified ECDSA Against Passive Adversaries," Cryptology ePrint Archive, Paper 2009/620, 2009. [Online]. Available: https://eprint.iacr.org/2009/620 → Cited on pages 66 and 100.

[237] F. Basic, C. Steger, and R. Kofler, "Poster: Establishing Dynamic Secure Sessions for Intra-Vehicle Communication Using Implicit Certificates," in *Proceedings of the 2022 International Conference on Embedded Wireless Systems and Networks (EWSN)*, ser. EWSN '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 196–197. → Cited on page 67.

[238] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software.* USA: Addison-Wesley Longman Publishing Co., Inc., 1995. → Cited on page 72.

[239] F. Basic, C. Steger, and R. Kofler, "Embedded Platform Patterns for Distributed and Secure Logging," in *26th European Conference on Pattern Languages of Programs*, ser. EuroPLoP'21. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3489449.3490004 → Cited on page 72.

[240] M. Gunnarsson, J. Brorsson, F. Palombini, L. Seitz, and M. Tiloca, "Evaluating the performance of the OSCORE security protocol in constrained IoT environments," *Internet of Things*, vol. 13, p. 100333, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660520301645 → Cited on pages 74, 87, and 109.

[241] T. Pornin, "BearSSL - a smaller SSL/TLS library," https://bearssl.org/, 2023, accessed: 16.05.2023. → Cited on page 81.

[242] AN5401, *Application Note: Getting Started with CSEc Security Module - Rev. 1*, NXP Semiconductors, 03 2018. → Cited on pages 81 and 82.

[243] D. Weiss, *Encapsulation.* GBR: John Wiley and Sons Ltd., 2003, p. 648–649. → Cited on page 81.

[244] *Specification of Secure Hardware Extensions*, AUTOSAR, 2019. [Online]. Available: https://www.autosar.org/fileadmin/standards/R22-11/FO/AUTOSAR_TR_SecureHardwareExtensions.pdf → Cited on page 82.

[245] ISO 15765-2:2016, "Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 2: Transport protocol and network layer services," International Organization for Standardization, Standard, 2021. → Cited on page 83.

[246] NXP Semiconductors, "NTA5332: NTAG 5 boost - NFC Forum-compliant I2C bridge for tiny devices," 2020, Product data sheet. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/NTA5332.pdf → Cited on page 86.

[247] F. A. Rincon Vija, S. Cregut, G. Z. Papadopoulos, and N. Montavont, "Autonomous Balancing Sleep Mode for Wireless BMS in Electric Vehicles," in *2021 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2021, pp. 65–71. → Cited on page 91.

[248] M. S. Alkatheiri, M. H. Eldefrawy, and M. K. Khan, “BAN Logic-Based Security Proof for Mobile OTP Authentication Scheme,” in *Future Information Technology, Application, and Service*, J. J. (Jong Hyuk) Park, V. C. Leung, C.-L. Wang, and T. Shon, Eds. Dordrecht: Springer Netherlands, 2012, pp. 53–59. → Cited on page 93.

[249] A. Rubin and P. Honeyman, “Formal Methods for the Analysis of Authentication Protocols,” CITI Technical Report 93-7, Tech. Rep., Nov 1993. [Online]. Available: http://www.citi.umich.edu/techreports/reports/citi-tr-93-7.pdf → Cited on page 93.

[250] P. Syverson and I. Cervesato, “The Logic of Authentication Protocols,” in *International School on Foundations of Security Analysis and Design*. Springer, 2000, pp. 63–137. → Cited on page 93.

[251] A. Mathuria, “Automating BAN Logic,” Master’s thesis, University of Wollongong, 1994. → Cited on page 93.

[252] N. Bindel and S. McCarthy, “The Need for Being Explicit: Failed Attempts to Construct Implicit Certificates from Lattices,” *The Computer Journal*, 10 2022, bxac132. [Online]. Available: https://doi.org/10.1093/comjnl/bxac132 → Cited on page 109.

[253] Y. Gao, S. F. Al-Sarawi, and D. Abbott, “Physical unclonable functions,” *Nature Electronics*, 2020. [Online]. Available: https://doi.org/10.1038/s41928-020-0372-5 → Cited on page 109.

[254] G. Selimis, R. Wang, R. Maes, G.-J. Schrijen, M. Münzer, S. Ilić, F. M. J. Willems, and L. Kusters, “RESCURE: A Security Solution for IoT Life Cycle,” in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ser. ARES ’20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3407023.3407075 → Cited on page 109.