Automated Authentication Credential Derivation for the Secured Configuration of IoT Devices

SIES'18

Thomas Ulz, Graz UT
(thomas.ulz@tugraz.at)
Thomas Pieber, Graz UT
Christian Steger, Graz UT
Andrea Höller, Infineon Austria
Sarah Haas, Infineon Austria
Rainer Matischek, Infineon Austria

Outline

1. Introduction
   1. Motivation
   2. System Model
2. Credential Derivation
   1. Process & Protocol
   2. Hardware Architecture
3. Evaluation
   1. Threat Analysis & Performance
4. Conclusion

Introduction – Motivation

„The S in IoT stands for Security.“

(unknown / reddit)

Introduction – Motivation

Website attacks show vulnerability of having default passwords

Home devices linked to the web, in 'Internet of Things', open sites to hacking attacks

Mon, Oct 24, 2016, 07:32 | Updated: Mon, Oct 24, 2016, 08:17

Is 'admin' password leaving your IoT device vulnerable to cyberattacks?

Internet-connected devices in your home or office will be vulnerable to botnets and other attacks, if you don't change the original login credentials.

By Danny Palmer | April 26, 2017 -- 10:10 GMT (11:10 BST) | Topic: Security

Passwords used in the biggest ever cyberattack revealed - and '12345' and 'password' were top

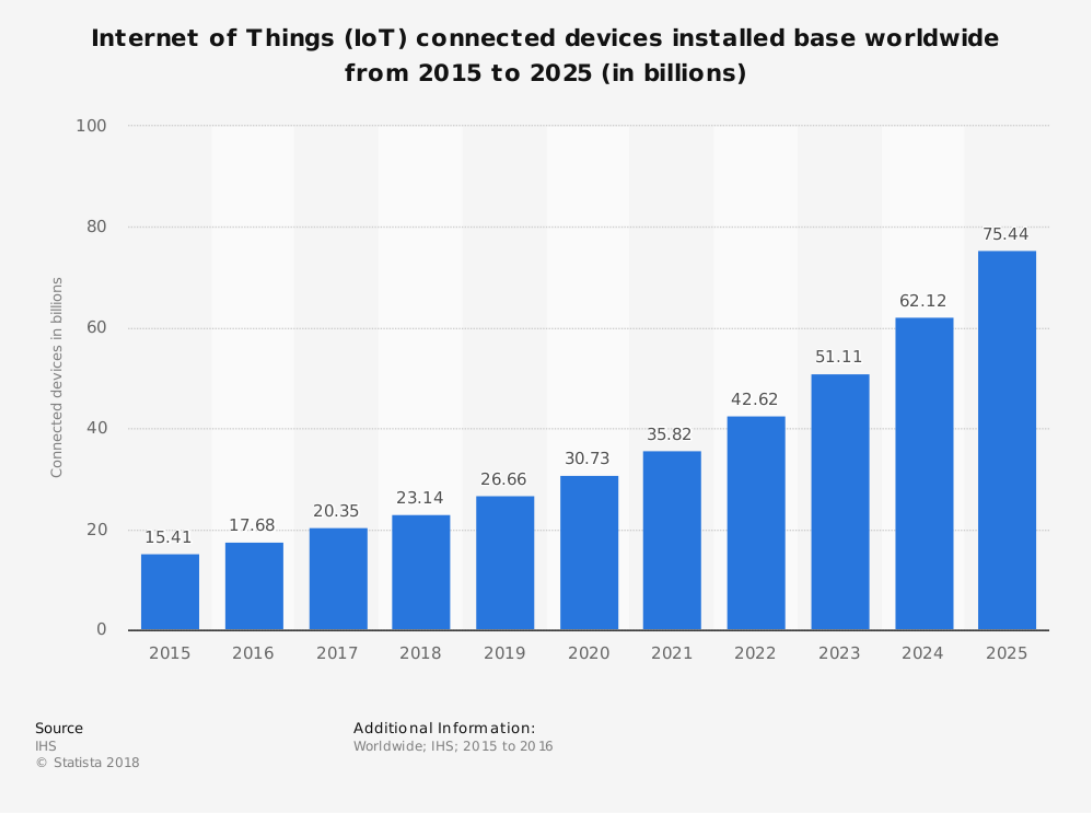
- **DDoS attack uses networks of computers that hackers bring under control**
- **It was revealed that Mirai botnet was one of two involved in recent attacks**
- **It used 61 unique username-password combinations to attempt access**
- **These were largely default credentials found among connected devices**

By CHEYENNE MACDONALD and ABIGAIL BEALL FOR DAILYMAIL.COM
PUBLISHED: 19:20 BST, 6 October 2016 | **UPDATED:** 08:34 BST, 7 October 2016

Introduction – Motivation

15% of All IoT Device Owners Don't Change Default Passwords

By Catalin Cimpanu — June 19, 2017 — 10:35 AM — 0

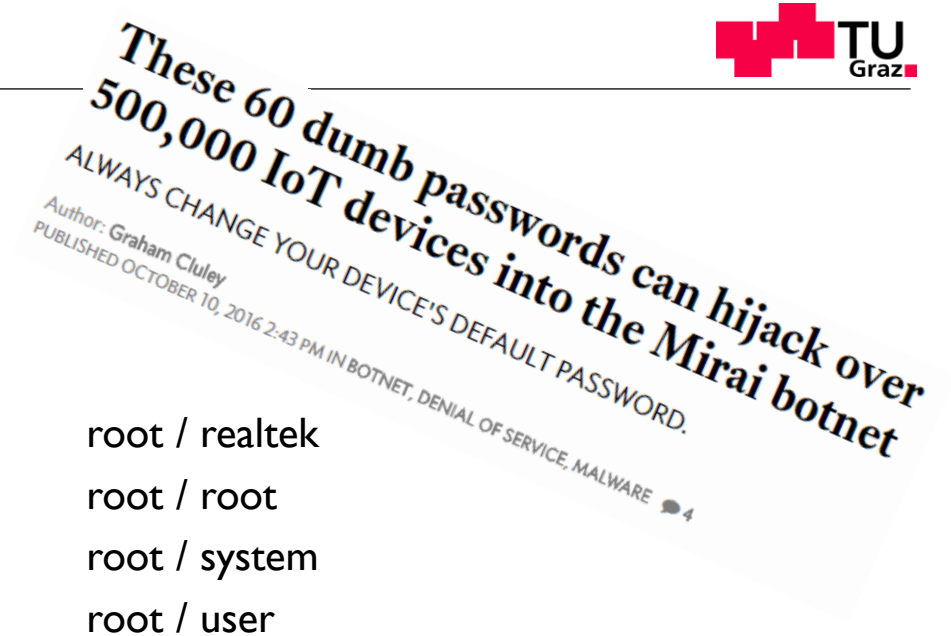


Five username-password combos is all you need

After performing several mass Internet scans, according to Positive Technology experts, just five username and password combos will be enough to get your hands on a large number of IoT devices, may they be DVRs, IP cameras, routers, smart washing machines, or anything else.

```
support/support
admin/admin
admin/0000
user/user
root/12345
```

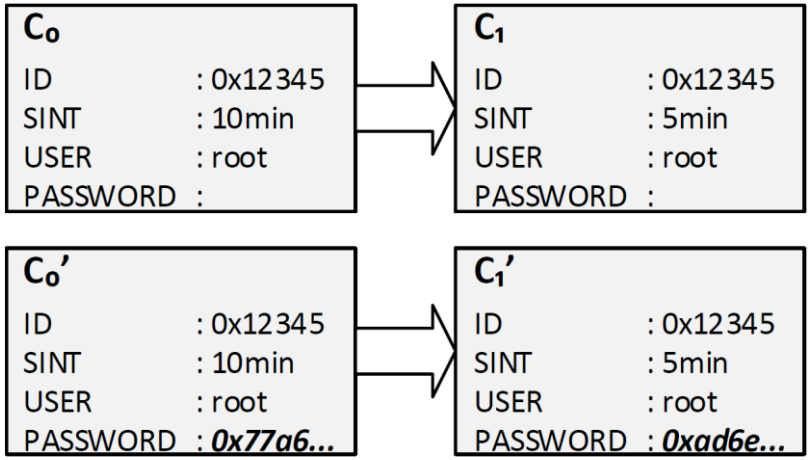
Introduction – Motivation



666666 / 666666	admin / smcadmin	root / 7ujMko0admin	root / realtek
888888 / 888888	admin1 / password	root / 7ujMko0vizxv	root / root
admin / (none)	administrator / 1234	root / 888888	root / system
admin / 1111	Administrator / admin	root / admin	root / user
admin / 1111111	guest / 12345	root / anko	root / vizxv
admin / 1234	guest / guest	root / default	root / xc3511
admin / 12345	mother / fucker	root / dreambox	root / xmhdipc
admin / 123456	root / (none)	root / hi3518	root / zlxx
admin / 54321	root / 00000000	root / ikwb	root / Zte521
admin / 7ujMko0admin	root / 1111	root / juantech	service / service
admin / admin	root / 1234	root / jvbzd	supervisor / supervisor
admin / admin1234	root / 12345	root / klv123	support / support
admin / meinsm	root / 123456	root / klv1234	tech / tech
admin / pass	root / 54321	root / pass	ubnt / ubnt
admin / password	root / 666666	root / password	user / user

Introduction – Possible Solutions

- More sophisticated and diverse default passwords?
  - E.g. as used for WiFi default passwords on routers
- Force users to change passwords?
  - When? On first login only? Repeatedly?
  - Leads to simple passwords such as ‚password‘
  - Enforce password constraints such as numbers
  - If complex passwords chosen, users might forget them



- Why not trigger an automated authentication credential derivation process?
  - On configuration changes → thats the reason we need credentials!

Introduction – System Model

- Arbitrary number of IoT devices
- Managed by a Configuration Back-End (CBE)
  - Is aware of current configuration state
  - Also all configuration updates known
  - Validates / attests correct configuration states
- Configuration data might contain confidential information
  - Such as IP or production relevant information for IIoT devices, WiFi keys for IoT devices, …
- Configuration data is protected while transferred
  - Confidentiality
  - Integrity
  - Authenticity

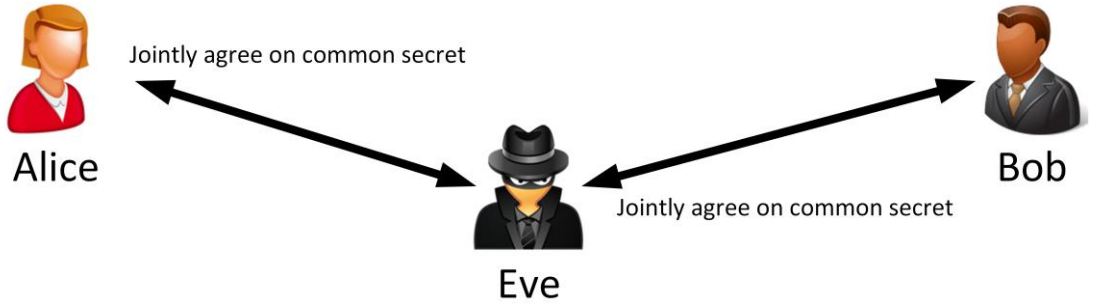
Outline

1. Introduction
   1. Motivation
   2. System Model
2. Credential Derivation
   1. Process & Protocol
   2. Hardware Architecture
3. Evaluation
   1. Threat Analysis & Performance
4. Conclusion

Credential Derivation – Primer on Key Agreement

- Two (or more) entities, agree on a common secret, such that
  - All involved entities influence the final key
  - An attacker is not capable of (easily) recovering the key

  - However, no authentication of involved entities



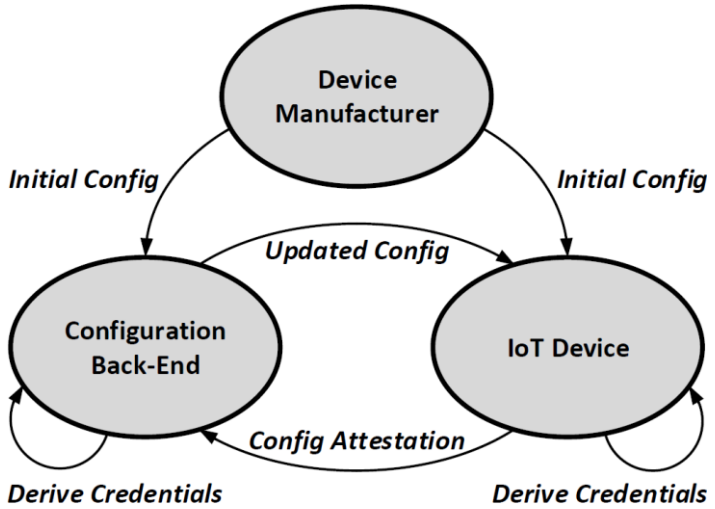
Credential Derivation – Primer on Key Agreement

- Solution: authenticated key agreement process
  - E.g. Diffie-Hellman (DH) with authentication

- SPAKE2
  - Lightweight authenticated key agreement based on DH
  - Uses passwords for authentication → previously discussed issues

Credential Derivation – Process

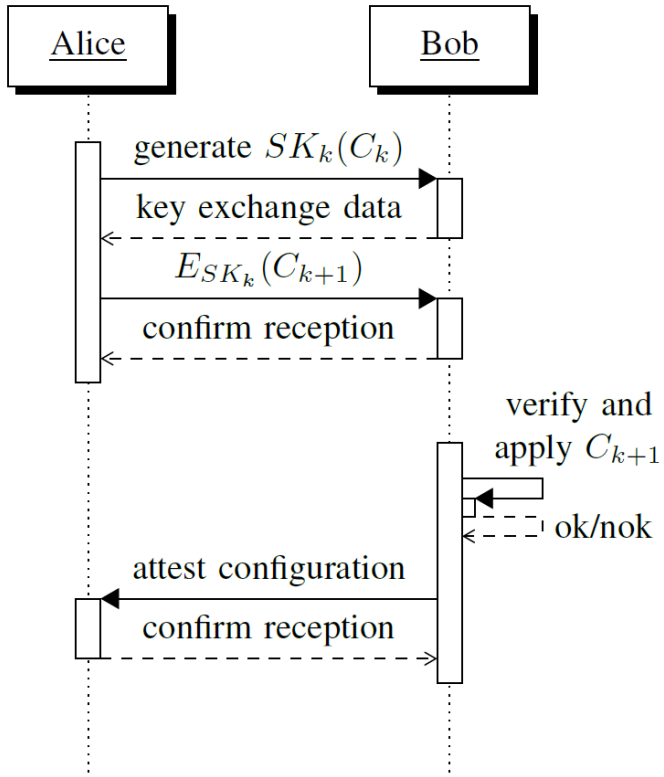
- In general, we have two types of configurations
  - Initial configuration by the device manufacturer
  - Subsequent configurations by the device‘s user
- IoT Device and CBE independently can derive authentication credentials
  - Based on currently applied configuration



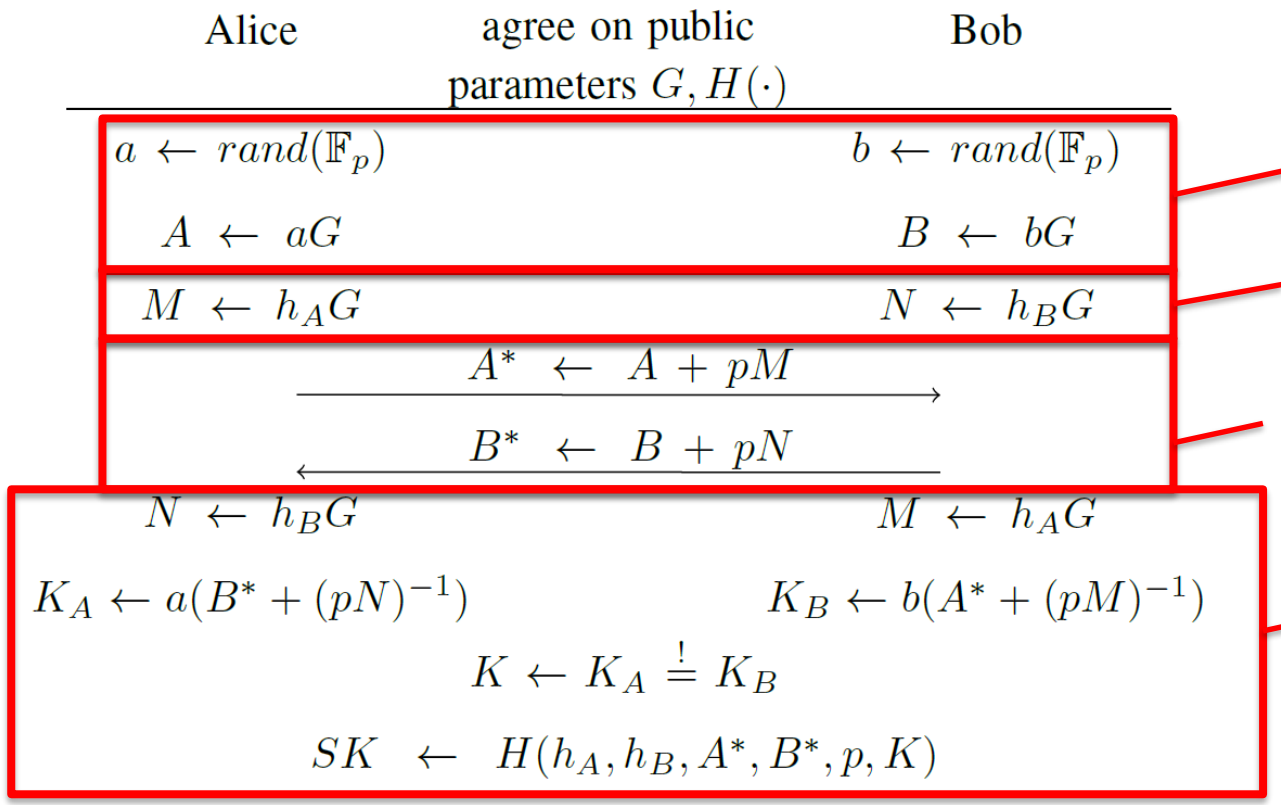
- Advantages
  - Improved security since process is automatically triggered by configuration update
  - Users do not need to remember passwords, since CBE manages derived credentials

Credential Derivation – Process

- Automatically derive passwords whenever changing a configuration
- Configuration is considered as shared secret
  - Thus, needs to be kept confidential
- K+1-th configuration is transferred encrypted
  - Based on a Session Key (SK)
  - Authenticated by credentials that are derived from the K-th configuration
  - And that is generated by an authenticated DH



Credential Derivation – Protocol



Standard ECC DH

Alice's and Bob's Identity

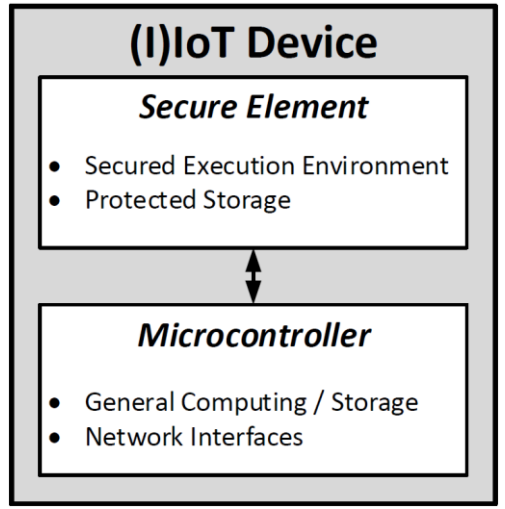
Authenticated Key Exchange
$p = H(C_k)$

Shared Session Key

Credential Derivation – Protocol

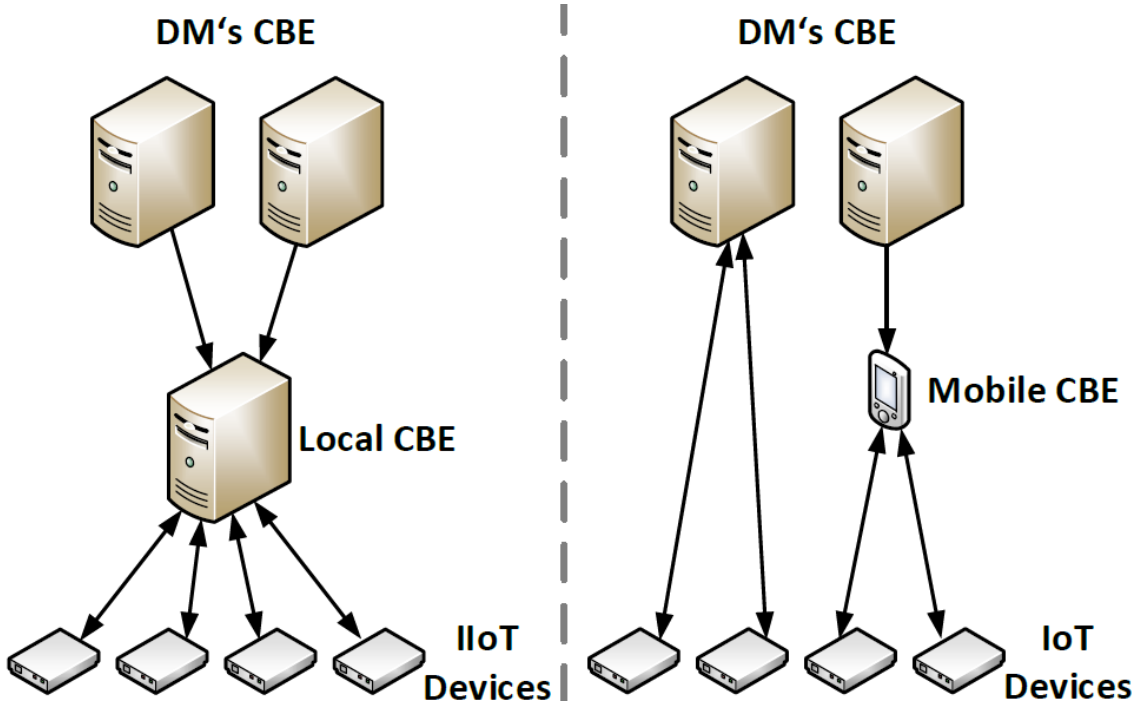
- Advantages
  - Key agreement authenticated
  - Only 1 roundtrip for authentication and key agreement
  - Shared secret for session key derivation is derived from current configuration

- To protect confidential information, we propose to use dedicated security hardware such as Secure Elements
  - To store confidential information
  - To perform cryptographic operations



Credential Derivation – Architecture

- Depending on usage scenario either
  - Local CBE hosted on dedicated hardware
    - Data not known by DM
  - Local CBE run on mobile device
    - Data not known by DM
  - DM‘s global CBE is used



Outline

1. Introduction
   1. Motivation
   2. System Model
2. Credential Derivation
   1. Process & Protocol
   2. Hardware Architecture
3. Evaluation
   1. Threat Analysis & Performance
4. Conclusion

Evaluation – Prototype

- Implemented and evaluated on Infineon hardware
  - XMC4500 general purpose microcontroller
  - SLE78 Secure Element (Common Criteria 5+ certified)

Evaluation – Security

- Comparison with password-based approach
  - Based on so-called Levenshtein distance
  - Distance between a password and a dictionary of words
  - E.g. distance of passwor*k* would be 1

- Since configuration parameters could be observable (such as WiFi names)
  - Include salt in form of a true random number (generated by the SE)

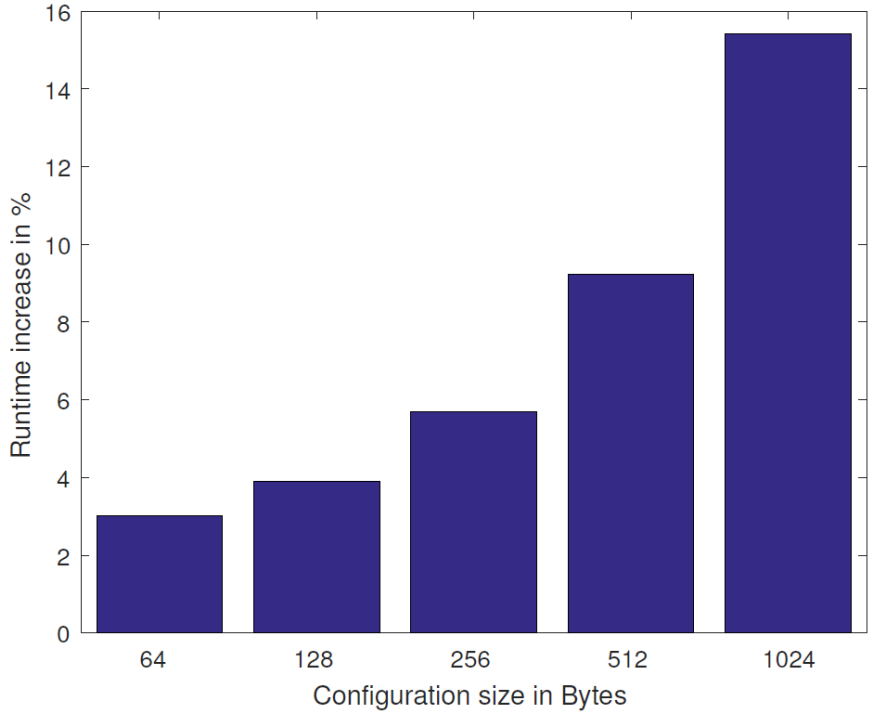
- However, if one intermediate configuration is known, subsequent SKs cannot be revealed due to DH properties → forward secrecy!

Evaluation – Security

- Threat analysis
- 2 Assets that need to be protected
  - IoT device and its functionality
  - Configuration data
- 10 threats are identified
  - 9 of them are completely mitigated
  - 1 is only partialy mitigated
- Residual risk
  - Denial of Service attacks
  - However, only SE is attacked
  - Normal operation of IoT device not influenced (besides side effects such as draining battery)

Evaluation – Performance

- Not allowed to tell absolute numbers
  - Runtime increase compared to unauthenticated DH



Outline

1. Introduction
   1. Motivation
   2. System Model
2. Credential Derivation
   1. Process & Protocol
   2. Hardware Architecture
3. Evaluation
   1. Threat Analysis & Performance
4. Conclusion

Conclusion

- Default passwords are a major issue for IoT devices
- However, forcing users to change them does not necessarily increase the device‘s security
- Thus, we proposed an automated authentication credential derivation process
  - Triggered by configuration updates
  - Using configuration data to derive these credentials
  - Based on authenticated DH to provide forward secrecy
- Induced overhead is reasonable
  - Thus, we think the approach is feasible for IoT devices

Acknowledgements

This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 692480. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Netherlands, Spain, Austria, Belgium, Slovakia.

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

IoSense is funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the program "ICT of the Future" between May 2016 and April 2019. More information: https://iktderzukunft.at/en/

bmvit
Austrian Ministry
for Transport,
Innovation and Technology

Thank you! Any questions?

Thomas Ulz

thomas.ulz@tugraz.at

Institute for Technical Informatics
Hardware/Software-Codesign Group
Graz University of Technology