Access Without Permission: A Practical RFID Relay Attack

Roman Silberschneider, Thomas Korak, and Michael Hutter

Institute for Applied Information Processing and Communications (IAIK),
Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria
Roman.Silberschneider@gmail.com
{Thomas.Korak, Michael.Hutter}@iaik.tugraz.at

Abstract

In this paper, we present a practical relay attack that can be mounted on RFID systems found in many applications nowadays. The described attack uses a self-designed proxy device to forward the RF communication from a reader to a modern NFC-enabled smart phone (Google Nexus S). The phone acts as a mole to inquire a victim's card in the vicinity of the system. As a practical demonstration of our attack, we target a widely used access-control application that usually grants access to office buildings using a strong AES authentication feature. Our attack successfully relays this authentication process via a Bluetooth channel (> 50 meters) within several hundred milliseconds. As a result, we were able to impersonate an authorized user and to enter the building without being detected.

1 Introduction

The Radio-Frequency Identification (RFID) technology has become very popular in the recent years. This is because of the simplicity and the ease of use in several application fields where in most cases the user does not have to configure parameters in order to initiate a communication. A communication is automatically established as soon as the two main components of an RFID system, a reader and a transponder, are in close proximity. RFID technology is nowadays widely applied in identification systems, ticketing systems for public transport, access control, e-passports, or mobile payment. Most of these applications make use of the common ISO/IEC 14443 standard or the Near-Field Communication (NFC) standard ISO/IEC 18092. Both standards are almost identical in the lower protocol levels and specify a reading range of up to 10 centimeters [4].

The various fields of application yield several different requirements. These requirements include the communication range of the system, the cost of the transponder, functionality, or the level of security. Applications in the payment or access-control sector, for instance, require cryptography to provide a barrier against basic attacks. Most of the commercially available (contactless) smart cards and RFID tags on the market therefore provide authentication or encryption features to protect the communication and transferred data between tag and readers.

In the last decade, however, it has been shown that these devices are susceptible to different attacks. To give an example, the widely used Digital Signature Transponder (DST) RFID transponder from Texas Instruments has been attacked from a research group from Johns Hopkins University and RSA Laboratories in 2005. The transponder provided encryption capabilities and was used in millions of cars to protect against theft (e.g., Ford, Toyota, Nissan etc.) and millions of payment-transaction systems (Exxon-Mobile Speedpass) that allows to pay contactlessly in supermarkets and restaurants (e.g., McDonalds deployed the system in over 400 restaurants in the Chicago area). To perform the attack, the authors used sixteen FPGAs and performed a brute-force attack to reveal the secret key. They demonstrated their attack in a practical scenario where they opened several cars on a parking lot and bought gas for free at Speedpass-enabled gas stations [1]. Other examples are the attack on Mifare Classic [13] or the KeeLoq system [3] which was used in many remote keyless entry systems such as car immobilizers and garage doors.

Besides these attacks and what has been upcoming over the last years are so-called *relay attacks*. The goal of these attacks is to make the reader believe that it communicates with a valid transponder inside the communication range. In fact, this transponder is a special hardware device (proxy) controlled by an attacker. It simply forwards the reader command to another device (mole) which establishes a communication with the valid transponder. With a setup like mentioned above the communication distance between reader and transponder can be extended from several centimeters up to many meters or kilometers, depending on the communication channel used between proxy and mole. Furthermore, since the communication is only forwarded, the secret key does not need to be known. The encrypted data or the authentication process is simply relayed between the reader and the tag without being detected.

In this paper, we present a practical relay attack on

a "real-world" access control application. The targeted implementation is used to grant access to buildings and is widely used in practice. It is based on the common ISO/IEC 14443 and ISO/IEC 7816-4 standards (which are also used for cashless payment and ticketing, for instance). First, we show how to impersonate a legitimate user to grant access to the building using a self-designed relay proxy and an Android smartphone as a mole. The attack works over a distance of about 50 meters and relays the communication within several hundred milliseconds over a Class 1 Bluetooth channel. Second, we present an attack by relying even an AES encrypted communication. Our results highlight the risks of these attacks and demonstrate the simplicity and convenience to implement them.

The rest of the paper is structured as follows. In Section 2, we give a brief overview on related work on this topic. Section 3 describes the used setup. Section 4 presents detailed information about the implementation. Section 5 gives the results and conclusions are drawn in Section 6.

2 Related Work

There are several papers on relay attacks in literature presenting attacks and countermeasures on various RFID systems. One of the first work in this field was published by Kfir et al. [10] in 2005. The authors focus on increasing the communication range between reader and proxy and mole and victim transponder, respectively. In [7], several attacks on proximity coupling systems are presented, including a relay attack. A mobile phone with NFC functionality as mole and a programmable RFID tag prototype as proxy were used for the attack in [9]. In their work, Bluetooth was used as communication channel between proxy and mole as also used in our experiments. The authors mention that because of the delay introduced by the Bluetooth communication it is not possible to relay the Unique ID (UID) of the victim's card. The reason are strict timing constraints in the used communication standard which are several microseconds during the anticollision and selection of the tag. For this, a setup as presented by Thevenon et al. [16] is needed that relays the communication data using plain analogue components only. However, by reading out the UID of the victim card and setting it on the proxy in a first step, the authors circumvent the strict timing constraints. For higher-level commands (application data units, APDUs) the response time are higher going up to 5 seconds using waiting time extensions, making the relay attack with this setup possible.

As a response to these attacks several countermeasures have been proposed. In the work of Hancke et al. [6], for example, a distance-bounding protocol is presented in order to predict the distance between reader and transponder. The response time of the transponder to single-bit challenges is measured at the reader side for that purpose. In [11] a modification in order to increase the performance of the protocol by Hancke is proposed. Reid et al. improved the protocol in [14]. With this improvement, the protocol can resist more-sophisticated attacks described in their paper. Another improvement is presented in [12]. Most of the these countermeasures, however, require additional modifications to existing RFID systems and are not conform to the used ISO/IEC 14443 standard.

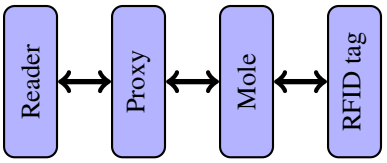

Figure 1. Overview diagram

3 Our Relay Setup

A relay attack consists of four major components: an RFID reader, a proxy, a mole, and a tag. Figure 1 shows an overview of the relay setup. The proxy and the mole are in-between the "classical" reader and tag communication and simply (passively) forward the entire communication over a so-called relay channel. There exist various channels such as WiFi, Bluetooth, Internet, direct cable, etc. Note that the faster the communication via the channel, the faster will be the relay of the attack.

Using a proxy and mole device, an adversary is able to enlarge the distance of the RFID communication up to several meters or even kilometers (as recently shown in [15]). As a proxy device, we used a HF RFID-tag emulator that is freely programmable. The Google Nexus S smart phone was used as a mole since it provides an NFC interface and Bluetooth capabilities. As a target device, we relay the communication of a self-designed cryptography-enabled RFID IC fabricated in 350 nm CMOS process technology using the Austriamicrosystems library. In the following, all components of our attack are described in a more detail.

3.1 Crypto-Enabled RFID Tag

For the relay attack, we used a self-designed RFID tag that operates in the 13.56 MHz frequency range. This HF field is generated by a reader that is used to power the tag as well as to allow communication between reader and tag. The tag works passively and does not need a power source like a battery for proper operation. The communication protocol between reader and tag is implemented according to the ISO 14443-4 standard [8] (type A). After the reader has selected the tag, application-data units (APDUs) are used in order to exchange information between reader and tag.

In order to allow authentication services, the Advanced Encryption Standard (AES, [2]) is implemented on the tag (as a cryptographic co-processor). AES is a symmetric block cipher that uses the same secret key for encryption and decryption. If tag and reader authenticate using AES,

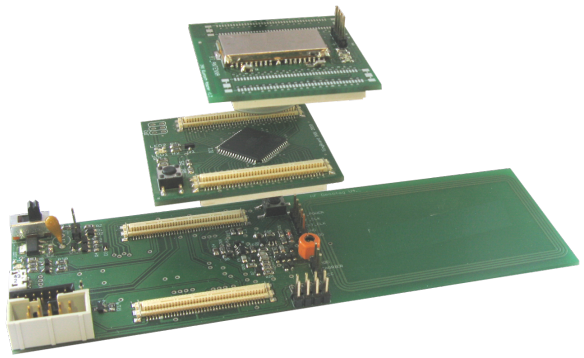

Figure 2. Components of the proxy: main board (bottom), microcontroller (middle), and Bluetooth (top).

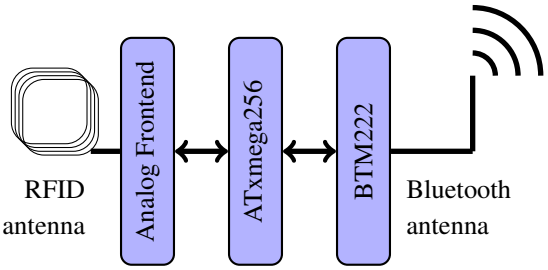

Figure 3. Schematic view of the proxy components

both parties require the same key (has to be known a priori).

According to ISO/IEC 7816-4, our tag supports the *Internal Authenticate (IntAuth)* APDU in order to authenticate to a reader. The *IntAuth* command sent from the reader, includes an eight bytes long challenge. As soon as the tag receives this command it encrypts the included challenge using AES and sends the encrypted data back to the reader. The reader can decrypt the received data again and verify the authenticity of the tag.

3.2 The Mole

As a mole, the Google Nexus S smart phone is used. The device is running the Android Operating System (Version 4.0.4) but the application is compatible to other versions and devices too. For application development, Eclipse with Android SDK (API level 17) was used. In practice, the mole is used to get in contact with the victims RFID tag using the NFC interface. The data is then transferred over Bluetooth to the proxy.

3.3 The Proxy

The proxy consists of an analog frontend, an 8-bit microcontroller, and a Bluetooth module. As shown in Figure 2, the RFID-tag emulation board consists of three main PCBs. The first PCB on the ground floor contains the voltage supply circuit, serial/USB connectors, an analog RF frontend, and a JTAG interface for programming. The PCB in the middle of the figure contains the microcontroller that is used to handle the protocols and the communication with the Bluetooth module.

Figure 3 shows an overview of the interaction between the different components. The proxy receives data from the reader over the RFID antenna and analog front-end, the microcontroller and Bluetooth module are used to forward the data to the mole. As a microcontroller, an ATxmega256 from Atmel is used and as a Bluetooth module we used a BTM222 from Rayson. The Class 1 Bluetooth module comes in SMD package and is placed on a small adapter PCB board on top of the microcontroller adapter board. We used the serial communication interface between the ATxmega256 and the BTM222 with LVTTL level (3.3V) and a baudrate of 115.2 kilobaud. Development is done using Rowley CrossWorks IDE (Version 2.0.4).

The proxy (RFID-tag emulator) is freely programmable. It is possible to emulate any smart card and tag and it is also possible to even clone the UID—a feature that is not supported by almost all modern smart phones. Using this setup, we are therefore even able to clone and relay the UID of smart cards and RFID tags which is powerful in cases where the UID gets checked by the RFID system (and rejected in case the UID is incorrect).

3.4 Backend Application

As a backend application, we have written a Java application that simulates a simple gate of an office building. In fact, this gate really exists and we were able to practically open the gate using our devices. However, for practical demonstrations, we implemented a simple GUI that shows if the gate opens when access is granted or still closed when it is denied. The application is connected to a Tagnology Multi ISO [5] reader that is connected over USB with a PC or laptop.

4 Implementation

In this section, we present the implementation details of the used relay components.

4.1 The Crypto-Enabled Tag

The crypto tag consists of several parts: an antenna, an analog front-end, and a digital part. A schematic view of the system is shown in Figure 4.

The analog front-end is connected to the antenna via two pins and implements all features to convert the analog RF signals into the digital world and vice versa. This includes the modulation and demodulation of the signals, the power-supply circuit, and the clock extraction. The digital output signals are then connected to a framing logic, an 8-bit (self-designed) microcontroller, a crypto unit, and a memory unit. All the components are connected via an 8-bit AMBA interface.

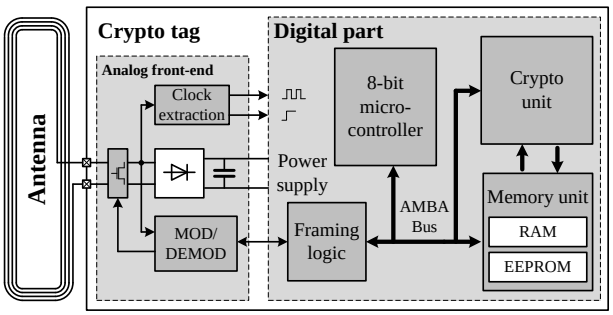

Figure 4. Schematic view of the crypto-enabled tag

The framing logic preprocesses the data received from the analog front-end and also forwards the response data to the modulation circuit. The microcontroller is mainly used for the protocol handling. It implements all necessary mandatory commands and handles application-specific features like AES authentication. For the latter purpose, the microcontroller makes use of a custom crypto-coprocessor that implements AES in hardware in order to fulfill the stringent requirements of passive RFID: low power consumption and small chip area. The memory unit consists of a RAM macro for volatile data which is lost as soon as the tag is removed from the reader field (tag is not powered anymore). Consistent data like the UID of the tag or the AES secret key is stored in the EEPROM macro block.

All components have been implemented in VHDL, parts are written in Verilog. The tag has been fabricated as a prototyping ASIC by Austriamicrosystems and operates passively by the Google Nexus S at a distance of up to 3 centimeters.

4.2 The Mole

The mole (Google Nexus S) communicates with the proxy via a Bluetooth connection on the one hand and with ISO/IEC 14443 tags on the other hand. For the Bluetooth connection, the mole acts as slave. This means that only requests of the proxy are received, processed, and answered. In particular, if the mole receives an *IntAuth* request from the proxy, i.e., "$IA \ || < random1 > || \ ;;$", it is forwarded to the victims tag where the included challenge gets encrypted. After receiving the answer of the victims tag, the mole forwards the answer to the proxy again.

We implemented an Application in Java for demonstration purposes as shown in Figure 5. The only action needed is to push the button "Start Relay!" on the screen which first establishes a Bluetooth connection to the proxy device. In the meanwhile, RFID tags are discovered automatically using the NFC interface. If a tag is within the reading range of the mole, the mole reads out the UID of the victims tag and sets it to the ISO/IEC 14443 SELECTED state. The UID is sent to the proxy via Bluetooth and clones the UID accordingly. After this, higher-level protocol commands can be relayed such as the AES authentication command *IntAuth*.

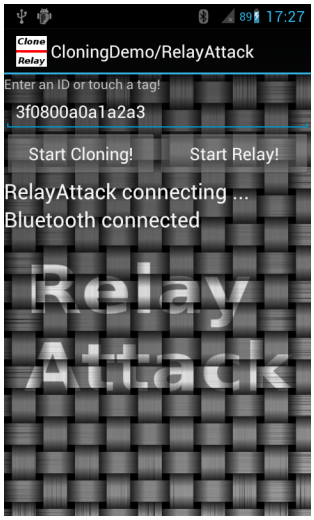

Figure 5. Screenshot Android Application

4.3 The Proxy

We implemented the proxy application in C and assembly language. The program is running on the ATxmega256. After establishing a Bluetooth connection with the mole, the proxy sets the UID of the victims tag and waits for an anticollision or select command from the backend reader. If a select command is received, the proxy answers with the UID of the victim's tag. Like for the mole, the proxy simply reads and forwards the higher-level APDUs [8] from the reader to the mole and back to the reader. The *IntAuth* command, in particular, will be transmitted in hex format as ASCII characters in the way specified in (1):

$$IA \ || < random1 > || \ ;; \qquad (1)$$

where $< random1 >$ represents the random challenge of the reader to perform the AES authentication. Note that no encryption is done at the proxy side but only relayed through the Bluetooth channel.

4.4 The Backend Application

Figure 6 and 7 shows the implemented backend application running on a demonstration PC. Figure 6 shows an open gate of an office building whereas Figure 7 shows when it is closed. In Figure 6, an authorized tag with UID="3F0800A0A1A2A3" and decrypted challenge "030AE3A1A326998" was successfully relayed over Bluetooth and authenticated by the backend system. Figure 7 shows the same device where the UID was the same but the secret key was incorrect, thus denying the access to the building. Note that in both figures the upper right combobox ("use AES Authentication (Crypta)") was enabled. If disabled, no authentication is done and only

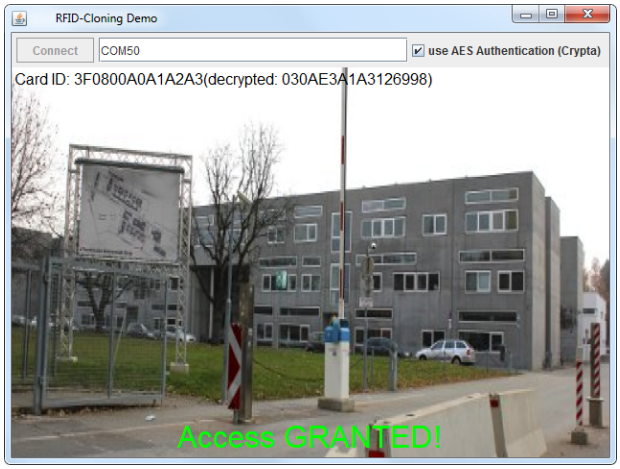

Figure 6. Screenshot of backend GUI: access granted

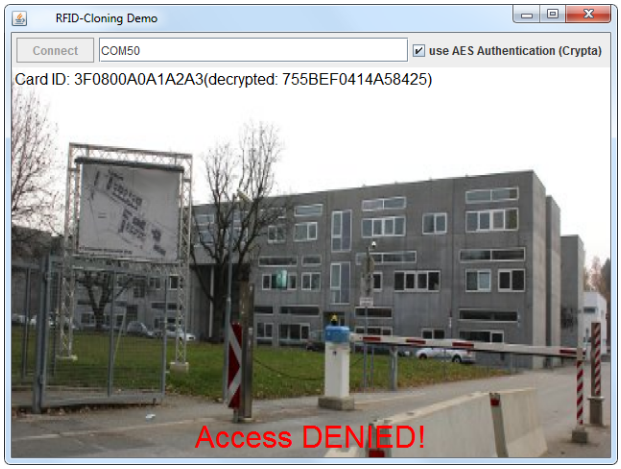

Figure 7. Screenshot of backend GUI: access denied

the UID is relayed and checked by the backend system (as done in real world by the tested office building).

The backend application was written in Java and is connected to an RFID reader over RS232. After setting the correct COM port and pushing the "Connect" button, a connection is established with the reader and an AES challenge ($random1$) is sent every second (if checkbox enabled). In a periodic manner the following commands are sent: REQA, Select, RATS, PPS, and *IntAuth* as shown in Table 1. The first two commands are used to select an ISO/IEC 14443-3 tag. The last three commands are ISO/IEC 14443-4 compatible commands used to implement the *IntAuth* command. The bold text in the last two command in Table 1 represent the sent 8-byte challenge of the reader (random number) and the 16-byte AES encrypted data from the crypto tag.

The received message is then decrypted with the secret key. The result looks as follows:

$$< random1 > \stackrel{?}{=} < random2 >, \qquad (2)$$

where $< random1 >$ represents the challenge sent in the *IntAuth* command and $< random2 >$ represents the eight byte challenge received by the proxy. If $random1$ is equal to $random2$, authentication was successful and the gate will be opened otherwise it keeps closed.

Figure 8 shows the communication flow between the backend application and the crypto tag.

Table 1. ISO/IEC 14443-4 command flow

Direction	Command	Data in Hex
send	REQA	26
receive	ATQA	4400
send	Select	9320
receive	...	883F0800BF
send	...	9370883F0800BF5C7C
receive	...	04DA17
send	...	9520
receive	...	A0A1A2A300
send	...	9570A0A1A2A300EBEA
receive	SAK	20FC70
send	RATS	E0803173
receive	ATS	057800A0027215
send	PPS req.	D0110052A6
receive	PPS resp.	D07387
send	*IntAuth* req.	0A000088000008 **EF7A383B9DD6D37E** 10C06B
receive	*IntAuth* resp.	0B00 **F1D64E3C15597C3B 3D55876B302BA11C** 9000BF9E783FBD41

5 Results

Using the described setup, we were able to relay a secure communication between a crypto tag and a reader. As a result, access is granted to an office building as demonstrated in a real world experiment as well as in a self-written Java demo. The total time for one relay procedure is about 200 ms. Most of this time is caused by communication. For example, a ping from the proxy to the mole and back takes up to 150 ms. This is a rather long time for practical attacks, however, it is fast enough to relay ISO/IEC 14443-4 commands because the response time can be manually extended by the proxy up to 5 seconds (using so-called Waiting Time Extensions). Using the Bluetooth channel, we were able to enlarge the communication distance between reader and tag to more than 50 meters.

We also made experiments using other NFC-enabled mobile devices to act as a mole. For this, we used the same Android application and successfully tested the demo on a Galaxy Nexus smart phone (Android Version 4.2.2) and a Nexus 7 tablet (Android Version 4.2.2).

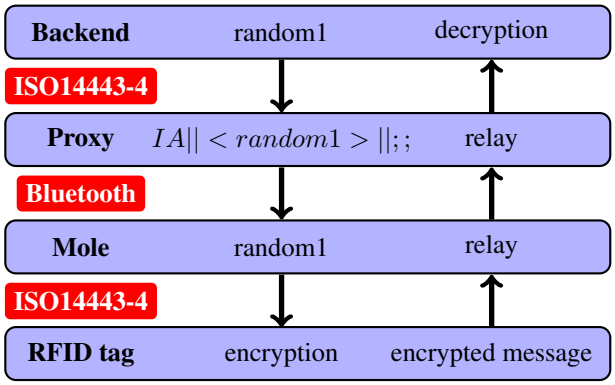

Figure 8. Communication flow of the attack

6 Conclusion

In this paper, we demonstrated a practical relay attack on an existing "real-world" access control system. In contrast to existing work, we relayed an encrypted communication using AES. With a custom proxy device and an off-the-shelf smart phone the distance between reader and tag could be extended to more than 50 meters. The attacks have been performed with different NFC-based smart phones, all running the Android operating system and our developed application. Countermeasures, e.g., distance bounding protocols, have already been proposed in order to make relay attacks infeasible. The fact that these protocols however are not standard conform as well as the effort for modifying the hardware makes the integration into existing systems hard. Our attack shows that it is highly recommended to update existing RFID systems in order to protect them against relay attacks.

Acknowledgements.

The work has been supported in part by the Austrian Government through the research program SeCoS (project number 836628) and by the Austrian Science Fund (FWF) under the grant number TRP251-N23.

References

[1] Steve Bono, Matthew Green, Adam Stubblefield, Ari Juels, Avi Rubin, and Michael Szydlo. Security Analysis of a Cryptographically-Enabled RFID Device. In *USENIX, Baltimore, Maryland, USA, July-August, 2005*, pages 1–16.

[2] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 2002.

[3] Thomas Eisenbarth, Timo Kasper, Amir Moradi, Christof Paar, Mahmoud Salmasizadeh, and Mohammad T. Manzuri Shalmani. On the Power of Power Analysis in the Real World: A Complete Break of the KEELOQ Code Hopping Scheme. In *CRYPTO 2008, Santa Barbara, CA, USA, August 17-21*, pages 203–220. Springer.

[4] Klaus Finkenzeller. *RFID-Handbook*. Carl Hanser Verlag, 2nd edition, April 2003. ISBN 0-470-84402-7.

[5] TAGnology RFID GmbH. TAGscan Industry | HF Multi ISO. `http://www.rfid-webshop.com`, July 2013.

[6] Gerhard Hancke and Markus Kuhn. An RFID Distance Bounding Protocol. In *SecureComm 2005, Athens, Greece, 5-9 September.*, pages 67–73.

[7] Gerhard P. Hancke. Practical Attacks on Proximity Identification Systems. In *IEEE Security and Privacy – S&P 2006, Berkeley/Oakland, California, USA, 21-24 May*, pages 328–333.

[8] International Organization for Standardization (ISO). ISO/IEC 14443-4: Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards - Part4: Transmission Protocol. Available online at `http://www.iso.org`, 2008.

[9] Wolfgang Issovits and Michael Hutter. Weaknesses of the ISO/IEC 14443 Protocol Regarding Relay Attacks. In *RFID-TA 2011, Barcelona, Spain, September 15-16.*, pages 335–342.

[10] Ziv Kfir and Avishai Wool. Picking Virtual Pockets using Relay Attacks on Contactless Smartcard Systems. In *SecureComm 2005, Athens, Greece, 5-9 September.*, pages 47–58.

[11] Jorge Munilla, Andres Ortiz, and Alberto Peinado. Distance Bounding Protocols with void-challenges for RFID. In *RFIDSec 2006, Graz, Austria, July 12-14.*

[12] Jorge Munilla and Alberto Peinado. Enhanced low-cost rfid protocol to detect relay attacks. In *Wirel. Commun. Mob. Comput*, 2010.

[13] Karsten Nohl. Cryptanalysis of Crypto-1. Computer Science Department University of Virginia, White Paper, 2008.

[14] Jason Reid, Juan Gonzalez Neito, Tee Tang, and Bouchra Senadji. Detecting Relay Attacks with Timing Based Protocols. ASIACCS, Singapore, March 20-22, 2007.

[15] Luigi Sportiello and Andrea Ciardulli. Long distance relay attacks. In *RFIDsec 2013, Graz, Austria, 9-11 July.*

[16] Pierre-Henri Thevenon, Olivier Savry, and Smail Tedjini. On the weakness of contactless systems under relay attacks. In *SoftCOM 2011*, pages 1–5.