Master thesis

Network Forensics, 60 credits

Evaluating intrusion detection points in an end-to-end solution

Digital Forensics, 15 credits

26th May 2023

Lilla Pankaczi

HALMSTAD UNIVERSITY

Lilla Pankaczi: *Evaluating intrusion detection points in an end-to-end solution*

SUPERVISOR:
Mohamed Eldefrawy

ABSTRACT

Evaluating all intrusion detection points in an end-to-end cyber - physical system can be challenging. This master thesis focuses on evaluating the security of the most exposed part of such systems, Radio Frequency Identification (RFID) communication. As both the RFID reader and tag can be located outside of secure premises, RFID communication can be a target of several cyber threats. Common cyber-attacks such as replay attacks, eavesdropping, or tag cloning can be associated with the lack of security of the communication channel between the reader and the tag or flaws of the implemented authentication protocols and encryption algorithms. This thesis briefly summarizes parts 4 and 3 of the ISO/IEC 14443 standard, which specify the initialization, selection, and transmission protocols in high-frequency RFID smart-card and reader communication. A formal security analysis was conducted to evaluate these protocols using a tool called Scyther. Then, an improved authentication protocol was proposed utilizing a commercially available feature, the Random Unique Identifier of the card (RID). The Scyther protocol verification results showed that implementing RID can prevent many RFID attacks such as, eavesdropping or replay attacks, and protect the cardholder's privacy.

PUBLICATIONS

A conference paper based on this thesis with the title "Enhancing the Security of ISO/IEC 14443-3 and 4 RFID Authentication Protocols through Formal Analysis" has been accepted at the IEEE COINS (Conference on Omni Layer Intelligent Systems), Germany, 2023.

ACKNOWLEDGEMENTS

I would like to thank Henrik Forsberg and Oskar Holmqvist from Axis Communications, Sweden for their valuable discussions and support.

CONTENTS

LIST OF FIGURES

LISTINGS

ACRONYMS

2K3DES Two-key Triple DES

3K3DES Three-key Triple DES

AES Advanced Encryption Standard

ATQA Answer to Request command

ATS Answer to Select command

CC Common Criteria

CEMA Correlative Electromagnetic Side Channel Attack

CIA Confidentiality, Integrity and Availability

CMAC Cipher-based Message Authentication Codes

CRC Cyclic Redundancy Check

CPS Cyber-Physical System

CVE Common Vulnerabilities and Exposures

DES Data Encryption Standard

DoS Denial of Service

EAL Evaluation Assurance Level

ECC Elliptic Curve Cryptography

ECDH Elliptic Curve Diffie-Hellman

IEC International Electrotechnical Commission

IDS Intrusion Detection System

IoT Internet of Things

ISO International Organization for Standardization

NIST National Institute of Standards and Technology

PCD Proximity Coupling Device

PICC Proximity Integrated Circuit Card

RATS Request for Answer to Select Acknowledge command

REQA Request command

RF Radio Frequency

RFID Radio Frequency Identification

RID Random Unique Identifier

RNG Random Number Generator

SAK Select Acknowledge command

TRNG True Random Number Generator

UID Unique Identifier

1 INTRODUCTION

End-to-end solutions, simply put, are products, services, or systems where the scoping, business analysis, development process, product testing, validating, and launching take place without any help from a third party [2]. End-to-end solutions generally are complex systems or services in functional form after developing them from beginning to end [3]. Cyber-Physical Systems (CPS) involve various interconnected systems that monitor and manipulate real objects and processes [4], and can also be developed as an end-to-end solution. By enabling advanced efficiency and connectivity of devices, systems, and services across numerous domains, Cyber-Physical Systems serve as the basis for the Internet of Things (IoT) and Industrial Internet [5].

Once an end-to-end solution is sold to a customer, despite high-security considerations through the design process and strong security configuration recommendations, the security of the installed system will largely depend on the underlying physical and IT security of the customer where the system is deployed. Therefore, evaluating intrusion detection points in the whole end-to-end system would have a vast scope and produce results that could be hard to replicate due to many customers and industries using the end-to-end solutions in different fields, settings, and applications. This thesis focuses on potential intrusions in the most exposed part of an end-to-end solution, particularly a Radio Frequency Identification (RFID) system. It formally analyzes the identification and mutual authentication protocol in a Radio Frequency Identification (RFID) System.

1.1 PROBLEM FORMULATION AND RESEARCH QUESTIONS

As established, evaluating intrusion detection points at all possible places within a cyber-physical end-to-end system would have an enormous scope; this master thesis focuses on the most exposed part of these systems, the RFID communication as it can be located outside of secure premises, and therefore could be accessible to anyone. Formal analysis was conducted on the security of the communication between an RFID reader and tag using an analyzer tool called Scyther to evaluate the card-reader communication as the first intrusion point in an end-to-end cyber-physical system. A review of previous academic literature and Common Vulnerabilities and Exposures was performed. To the best of my knowledge, this is the first work that formally analyzes the security of the identification and authentication protocol used by RFID readers and specific types of widely

used RFID tags, the MIFARE DESFire EV2 and EV3 cards. This formal security analysis is the novel contribution of this thesis and aims to answer the following research questions: *(i) How can RFID communication be evaluated as an intrusion detection point? (ii) Is the communication protocol between an RFID reader and a tag secure? (iii) Can the security of the protocol be improved?*

1.2 ORGANIZATION

The rest of this thesis is organized as follows. Chapter 2 presents a general overview of RFID systems, the ISO/IEC 14443 standard, a detailed description of parts 3 and 4 of this standard which contains identification and authentication protocols for RFID communications, and the analyzer tool called Scyther. A summary of related work on the security of RFID systems, MIFARE cards, and previous Scyther studies is also included in this chapter. Chapter 3 introduces the method and experiment set up for this master thesis and the Scyther input for the analysis. Chapter 4 summarizes the findings. Chapter 5 discusses future directions and reflects on the research question. Chapter 6 concludes this thesis.

2 BACKGROUND AND RELATED WORK

2.1 RFID - RADIO FREQUENCY IDENTIFICATION

RFID is short for Radio Frequency Identification, and it is a form of wireless communication. It uniquely identifies objects through the radio frequency portion of the electromagnetic spectrum [6]. An RFID system typically consists of an RFID reader that usually combines an antenna and a transceiver, and an RFID tag that contains a transponder. There usually is an application software or a database with a secure wired connection to the reader, which is responsible for data processing [7]. An RFID reader is a device that connects to a network and utilizes radio waves to send signals that trigger the tag and receives waves that can be translated to data from an activated tag [6]. The transponder (tag) consists of an integrated circuit (IC) and an antenna embedded in plastic [6]. The IC is responsible for responding to the reader and modulating and demodulating the radio frequency signals. At the same time, the antenna absorbs the incoming radio frequency waves from the reader and uses this absorbed energy to activate the IC [8]. RFID tags are usually constructed as keys, key fobs, smart labels, smart cards, and wristbands. The reader manages the radio frequency communication with the tags [8].

There are three main types of RFID systems depending on frequency. Low - Frequency systems range from 30 to 500 KHz, High - Frequency systems range from 3 to 30 MHz, and Ultra High - Frequency systems range from 300 to 960 MHz. Low - Frequency and High - Frequency radio communication use coil-shaped antennas, while Ultra High - Frequency communication utilizes traditional radio antennas. [8].

There are two types of RFID tags, Active RFID tag, which has its own power source, and Passive RFID tag, which receives power from the reading antenna [6]. It is essential to recognize that the communication between the reader and tag usually follows a Master-Slave relationship where the reader initiates communication and sends commands to the tag [9].

Adopting RFID technology has gained increasing popularity over time due to decreasing hardware costs. Common uses for RFID, include [6]: inventory management and inventory control, livestock tracking, vehicle tracking, transportation, access control, security, tap - and - go credit card payments, IoT deployments, and E - Passports.

A High-Frequency RFID system based on the ISO/IEC 14443 standard using passive RFID tags will be analyzed in this thesis.

2.1.1 *RFID Security Issues - Related Work*

According to Cheng and He [10], who conducted a survey on RFID attacks and defenses, the most common security requirements of a secure RFID system are the following; *(i) Mutual authentication* refers to the reader and the tag conducting two-way authentication in order to prevent spoofing and counterfeiting attacks. *(ii) Tag anonymity* refers to keeping the true identity of the tag's user confidential. Transmission between the card and the reader should therefore be encrypted. *(iii) Data integrity* refers to ensuring that the information has not been modified during transmission. *(iv) Backward security* which is strongly related to data integrity and refers to not being able to associate current and historical data with analyzing labels to obtain private information.

Authors in [10] and [11] mention the traditional classification of security threats according to the CIA triad (Confidentiality, Integrity, and Availability). The authors agree that if one of the principles of the CIA triad is not met, the security of a system is broken. They argue that this statement is true for RFID systems, too, as they are exposed to several types of attacks targeting the confidentiality, integrity, and availability of data stored on tags or the information exchanged between a reader and a tag.

Although different authors might categorize attacks targeting RFID systems in slightly different ways [10] [11] [1], they all agree that there are Physical attacks and Channel attacks. Authors in [10] also define System threats, while authors in [11] categorize Software attacks separately.

Physical attacks use physical means to attack an RFID tag or RFID communication. Examples of physical attacks are: Signal blocking, jamming, or electromagnetic interference blocking tag communications [10] [11]. Reverse engineering attacking tags that are not tamperproof to recover the encryption algorithm implemented in the card design [1] or get confidential information out of a tag [10] [11] by electronic, power, optical or communication protocol analysis [11]. RFID tag copying, tag removal [10] [11]. Physical destruction, for example, by applying pressure or exposing the tag to too high/low temperatures or certain chemicals [11].

Channel attacks refer to attacks related to the lack of security of the communication channel between the reader and the tag. For example: Eavesdropping attacks place a device between a tag and a reader to intercept communications between them [10] [11]. Replay attacks record eavesdropped signals with the intention of replaying them later [10] [11]. Relay attacks or Man-in-the-Middle attacks modify intercepted signals between the reader and the tag with the help of illegal devices (a mole and a proxy [12]) that do not need to be located in close proximity of the reader or the tag and can use other

forms of communication [10]. Side Channel attacks extract information from implementing cryptographic protocols by electromagnetic radiation, power consumption, or timing [1].

System threats refer to attacks targeting the implemented authentication protocols and encryption algorithms' flaws, such as tracking activity patterns or movements based on unique identity information stored on a tag or password decoding by deciphering intercepted encrypted traffic [10].

Software attacks are related to software vulnerabilities found in tags or readers, for example, SQL injection into reader middle-ware, command injection or remote code execution on readers, and malware injection to tags [11].

Common countermeasures to minimize some of the threats above include RFID shields, such as card shielding sleeves or passport cases that do not allow RF waves to pass through, encryption, randomization of tag IDs, implementing strong input validation, limiting account privileges and disabling scripts on the backend system[8].

2.2 ISO/IEC 14443 STANDARD

ISO is the International Organization for Standardization [13] and IEC is the International Electrotechnical Commission [14]. The ISO / IEC 14443 standard is an international standard consisting of four parts that govern the operation of contactless smart cards. This standard applies specifically to smart cards that operate at 13.56 MHz and require close proximity (within 10cm) to a reader antenna [15]. The reader is often referred to as Proximity Coupling Device (PCD), and the transponder in the RFID tag as Proximity Integrated Circuit Card (PICC) [12]. The ISO/IEC 14443 standard has two versions depending on the type of proximity cards or objects (Type A or Type B). This thesis focuses on Type A. The first part (14443-1) of the standard specifies the physical characteristics of the PICC, the second part (14443-2) covers the characteristics of the field, such as the radio frequency power, the third part (14443-3) outlines the initialization and anti-collision routine for PICCs, and the fourth part (14443-4) defines the transmission protocols for contact-less environments and specifies the activation and deactivation sequence used during the transmission protocol [12]. Parts 3 and 4 of the standard will be described in detail as understanding the communication flow between the PCD and PICC is crucial to perform accurate security analysis.

2.2.1 *ISO/IEC 14443-3 and ISO/IEC 14443-4*

2.2.1.1 *Initialization and Selection*

ISO/IEC 14443-3 covers PICC initialization and anti-collision routine. When a card (PICC) is placed in the reader's (PCD) field, it powers up due to modulation pulses from the reader that ensure continuous power supply to the card. The card then enters an IDLE state while the reader periodically issues Request Commands (REQA). The PICC then replies with an Answer to Request (ATQA) to the PCD. This means that the reader has successfully detected a card. After detection, the anti-collision process is started by the reader. This process aims to place only one card into an activated state if multiple cards are in proximity to the reader and to acquire the PICC's Unique Identifier (UID) by issuing the Request UID command and the Select command. As a reply, the PICC sends its UID, a Select Acknowledge Command (SAK), and becomes ACTIVE.[1].

As part of ISO/IEC 14443-4, the reader sends a Request for Answer to Select command (RATS) to the card. The card then replies with an Answer to Select (ATS) message, which means that the card is fully selected by the reader [1].

2.2.1.2 *Mutual Authentication*

These steps are followed by a mutual authentication protocol called three-pass authentication between the card and the reader [16]. The authentication process aims to mitigate several attack types, such as eavesdropping or replay attacks [1]. The authentication protocol makes sure that both parties have the same symmetric cipher key, typically DES (Data Encryption Standard) [17] or AES (Advanced Encryption Standard) [18].

The steps of the authentication protocol are as follows: The PCD issues an authentication command and sends a challenge request to the PICC. The PICC generates a random number (B), encrypts it with a long-term shared secret key, and returns this encrypted random number to the PCD. The PCD decrypts this message with the shared key, rotates the decrypted message, generates another random number (A), concatenates this to the decrypted and rotated (B), and encrypts the concatenated information, which is then sent back to the PICC. The PICC decrypts the message to obtain the random number (A) and the rotated (B), compares the original (B) with the one sent by the PCD, rotates the decrypted (A), encrypts it, and sends it back to the PCD. The PCD then decrypts this message and compares the original (A) with the one sent by the PICC.

If original (B) and the received (B), and the original (A) and the received (A) match, both the PCD and the PICC are set to authen-

tication state, and encrypted data exchange (Secure Messaging) will begin [16].

2.3 MIFARE DESFIRE CARDS

MIFARE is a trademarked brand of the company NXP [19]. The MIFARE brand contains multiple product families used in public transport, smart cities, hospitality, secure access management, event ticketing, and micropayment installations, among many others [19]. One of these product families is the MIFARE DESFire family, which includes highly secure micro-controller-based integrated circuits and has been created considering global standards for "both RF interface and cryptographic methods" [20]. The DES in DESFire references the cryptographic engines that are used to secure the data during transmission. The DESFire family uses DES, 2K3DES (two-key triple DES), 3K3DES (three-key triple DES), and AES as cryptographic algorithms [20]. It is important to note that NIST (National Institute of Standards and Technology) issued an update in 2017 urging all users to migrate from DES/3DES to AES due to known exploited vulnerabilities of DES/3DES and announced the depreciation of 3DES by 2023 [21]. The FIRE in DESFire stands for Fast, Innovative, Reliable, and Secure as characteristics of these integrated circuits [22].

Both DESFire EV2 and EV3 include the following hardware components: a CPU with a 16-bit architecture, an MMU (Memory Management Unit), ROM (Read-Only Memory), RAM (Random Access Memory), and NVM (Non-volatile Memory). The AES/DES co-processor units support the Triple DES operations with a key length of 112 or 168 bits or AES operations with a key length of 128 bits. The TRNG (True Random Number Generator) provides true random numbers for authentication. A MIFARE DESFire Software component is stored in ROM, which provides the main functionality of the card during usage. They also include several analog components such as voltage and temperature sensors [23]. Both types of cards have advanced security features such as Proximity checks to minimize the risk of relay attacks, Secure Messaging, and Random UID (short: RID), a feature introduced for privacy protection. It guarantees that no information that can be linked to the cardholder is ever openly disclosed. [24]. It is essential to mention that the DESFire family supports Double Size UIDs, which are 7 bytes long. However, the RID is always limited to 4 bytes [25]. During the authentication protocol, 16 bytes are transmitted between the PICC and the PCD [26].

2.3.1 *MIFARE DESFire EV2*

The MIFARE DESFire EV2 is a contactless integrated circuit (typically a contactless card) that is compliant with all levels of the ISO/IEC

14443 standard. It supports the three-pass authentication protocol explained above. It offers a high level of security using 3DES or AES hardware cryptographic engine for protecting the confidentiality and integrity of transmitted data [22]. "During the authentication, the level of security of all further commands during the session are set" [22]. According to the official datasheet of MIFARE DESFire EV2, the encrypted data transfer, often also referred to as Secure Messaging, is ensured by choosing a crypto method (as mentioned above, during the authentication process) as well as calculating, encrypting, and attaching CRC (Cyclic Redundancy Check) and CMAC (Cryptic Message Authentication Code) to the stream [22]. The MIFARE DESFire EV2 has also achieved a Common Criteria EAL5+ security certification. Common Criteria (CC) is an international set of guidelines for computer security certification [27]. EAL5+ refers to the Evaluation Assurance Level, where 5+ means that a product and system have been semi-formally designed and tested. When a system is semi-formally designed and tested, developers or users demand high security that is independently assured during planned development. They require a rigorous development approach that avoids unreasonable costs associated with specialist security engineering techniques. [28].

2.3.2 *MIFARE DESFire EV3*

The MIFARE DESFire EV3 introduced new features and enhancements compared to the MIFARE DESFire EV2. It is Common Criteria EAL5+ certified and fully compliant with all levels of ISO/IEC 14443, similar to EV2. It offers multiple features making the user experience smoother and introduces some new security features like the Transaction Timer [29]. The Transaction Timer aims to prevent Man-in-the-middle attacks by preventing an attacker from keeping a card powered up and concluding a transaction after a card left a legitimate reader device [29].

2.3.3 *MIFARE Security Issues - Related Work*

Previous research has been conducted on different types of MIFARE cards. The authors of [30] reviewed the security of MIFARE Classic and used a tool called Proxmark3 for their security tests. Proxmark3 is a popular tool for performing RFID analysis as it comes with hardware components, high-level emulation, and software functionalities [31]. Other researchers [32] have used Proxmark3 to perform overclocking of a MIFARE DESFire EV1 and an EV2 card and concluded that overclocked cards respond to their distance-bounding requests quicker than expected, which allows adversaries to perform a relay attack from a distance over 40km, despite the existence of distance-

bounding protocol implemented in the cards. Similarly to [32], the paper [33] pointed out that some distance bounding protocols can be vulnerable to replay attacks and mentioned that NXP 's distance bounding protocol which is implemented in the DESFire cards, might resist certain types of relay attacks, but potentially not all. Authors of [11] also used Proxmark3 but on a more general level. They analyzed both High and Low-frequency tags and evaluated the security of HID Proximity cards, which are proximity cards from another brand. Researchers in [34] mention the MIFARE DESFire EV2 card as a "known to be safe tag" and recommend using it instead of the MIFARE Classic 1K tag, which has multiple vulnerabilities confirmed by Proxmark3. For example, adversaries could clone the Classic card by decrypting data sectors and reverse engineering.

2.3.4 *MIFARE Common Vulnerabilities and Exposures*

Common Vulnerabilities and Exposures (CVEs) are part of the CVE program that aims to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities. The CVE project is sponsored by the US Department of Homeland Security and Cybersecurity and Infrastructure Security Agency [35].

As of 23rd February 2023, the reported vulnerabilities related to MIFARE were the following: *CVE-2022-40363* is a vulnerability affecting a device called Flipper Zero which is a multi-tool device with built-in applications to transmit and receive frequencies like RFID or NFC. Flipper Zero firmware versions up to 0.65.2 have been confirmed to be vulnerable to Buffer overflow and Denial of Service attacks when attackers craft malicious NFC files with MIFARE Ultralight [36]. *CVE-2021-33881* is a vulnerability found on MIFARE Ultralight cards where an attacker can interrupt a write operation over RFID to bypass a Monotonic Counter protection mechanism which can have a different impact depending on the application of the card [37]. *CVE-2020-16097* is a vulnerability affecting controllers and readers of another manufacturer. It is possible to retrieve keys securing MIFARE Plus and DESDire cards on a controller using debug ports on a specific type of card reader [38]. *CVE-2019-9861* is a vulnerability related to MIFARE Classic. Due to weak cryptography, the MIFARE Classic tag can easily be read and cloned and could be used to deactivate an entire alarm system in an unauthorized way [39].

To the best of my knowledge, none of the disclosed vulnerabilities are directly related to the MIFARE DESFire EV2 and EV3 cards. Therefore as a unique contribution of this thesis, a formal security analysis was conducted on the communication between an RFID reader and EV2 card while performing selection and mutual authentication.

2.4 INTRODUCTION TO SCYTHER

Scyther is a security protocol verification tool that can identify problems originating from how a given protocol is constructed. It has been chosen because of its usability. Compared to other protocol analyzer tools, Scyther is easy to learn and has an understandable graphical user interface and a graphical output to clarify verification results [40]. The tool assumes that an adversary cannot learn anything from encrypted messages without a decryption key [41]. The input for Scyther is a security protocol description that specifies the intended security properties and evaluates these. Scyther has a graphical user interface, and by convention, it takes files with extension .spdl (Security Protocol Description Language) as input [41].

Scyther, by default, uses the Dolev-Yao adversary model, which contains honest participants and the adversary. "The honest participants follow the steps of the protocol without deviation. They can engage in multiple runs of the protocol simultaneously and with different parties" [42]. The network is assumed to be entirely under the adversary's control, who can record, delete, replay, reroute, reorder, and control the scheduling of messages. [42]

2.4.1 *Related Work - Scyther*

Scyther has previously been used to analyze security protocols related to IoT communications. For example, the paper from [43] conducts a formal analysis on LoRa (Long Range), a commonly accepted LPWAN (Low Power Wide Area Network) protocol, and it identifies weaknesses in the Over The Air Activation procedure of version 1.0 LoRaWAN (Long Range Wide Area Network) that were correctly addressed in the implementation of version 1.1 with no vulnerabilities found by Scyther. Authors of [44] analyzed the security of a hash-based mutual authentication protocol and the security of the Rabin public key cryptography-based mutual authentication in an RFID system. They found multiple vulnerabilities, such as adversaries being able to desynchronize tags and, therefore, tags being traceable by adversaries. To tackle the vulnerabilities, the authors proposed enhancements to these protocols and analyzed their enhancements using Scyther. In the work of [45], RFID mutual authentication protocols, such as Song protocol, in RFID-enabled supply chains, RFID cloud-based scheme, and multi-tag group reading scheme were reviewed, and Scyther was used to formally analyze proposed improvements and a new lightweight RFID mutual authentication protocol. Authors of [46] proposed a secure scheme for a smart home environment and analyzed the protocols included with Scyther.

2.4.2 *Security Claims - Scyther*

As mentioned, Scyther evaluates security protocols, which is achieved by executing and verifying security claims. This section explains the main security claims, secrecy, and authentication (including aliveness, agreement, and synchronization) in more detail. *Secrecy* means that the information is not revealed to an intruder even though the communication is performed over an untrusted channel. *Aliveness* refers to simple authentication properties which check if there is a communication partner when a role is executed. *Agreement* means that "after successful completion of the protocol, the parties agree on the values of all (or some) variables" [47]. *Synchronization* refers to executing runs for each role in the protocol simultaneously, naturally suggesting an ordering of the events and unmodified delivery of messages. [47]. All these claims can be defined in several ways, such as Weak Aliveness and Non-Injective synchronization. Scyther provides a separate verification option called *Characterize Roles*, which, instead of focusing on finding attacks, tries to verify the correctness of the protocol, meaning that for a correct protocol, there is precisely one explicit trace pattern for each role [47].

3 METHOD

After gaining insight into how end-to-end cyber-physical systems operate, the following has been concluded: Evaluating intrusion detection points at every possible point in an end-to-end cyber-physical system would result in a huge scope. Even though the highest security configurations are recommended to end users and the system's security is of utmost importance when designing products, evaluating intrusion detection points in an entire end-to-end system would hugely depend on the end user's physical and IT security. Therefore the findings or contribution of this thesis could not be replicated. Instead, I focused on the most exposed part of an end-to-end system. A formal security analysis was conducted on RFID communication, more specifically on the identification and three-pass authentication protocol between an RFID reader and a MIFARE DESFire EV2 card, as these could be accessible by anyone due to their physical location and could be exposed to several types of attacks as explained in Chapter 2. The Scyther tool was used to analyze and detect potential intruders trying to intercept or modify the card-reader communication.

3.1 EXPERIMENT SETUP

Prior to feeding the input to Scyther, it is crucial to understand the protocol one analyzes. In *Experiment 1*, Scyther was used to analyze the identification and three-pass authentication protocol in the card-reader communication. The communication details have been discussed in Chapter 2 section "ISO/IEC 14443-3 and ISO/IEC 14443-4". According to selection and authentication protocols used by the standard, the parameters were fed to Scyther. In *Experiment 2*, additional security features were added to the parameters of Experiment 1, such as the Random UID (RID), which requires an additional authentication step prior to sharing the actual UID of the card. Using RID prevents being able to associate information with the cardholder, as RID is always freshly generated. The standard Scyther version was used for both experiments with default settings. However, the number of runs was reduced to three due to time constraints. This will be explained in more detail in Chapter 4.

3.1.1 *Experiment 1 - Unencrypted UID and Three-Pass Authentication*

In Experiment 1 (Figure 1), the input code has been supplied to Scyther specifying the `Protocol` by adding the Card and Reader

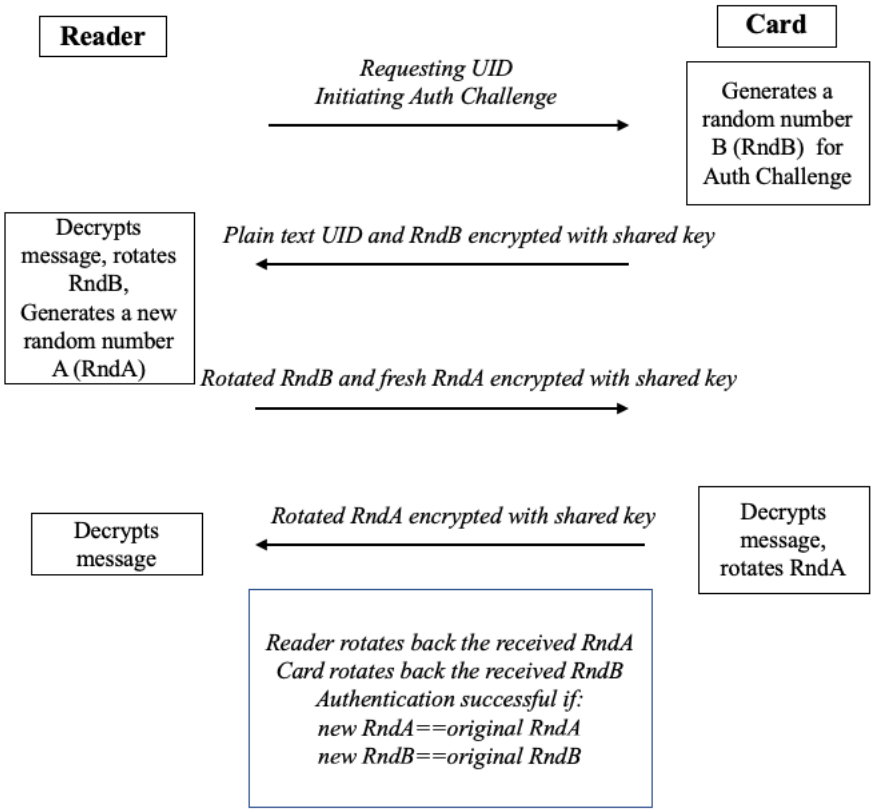

Figure 1: Experiment 1 - Transmission of plain text UID and three-pass authentication [1]

`Roles` because Scyther works with role-based descriptions of protocols [41]. Firstly, the transmission of the UID has been modeled. Transmission of the UID is necessary during the selection and anti-collision protocol, and all other standard commands, requests, and acknowledgments are ignored for modeling purposes. The UID has been defined as a constant on the card's end as it is hard-coded into the card, and it has been defined as a variable on the reader's end as it is a value that is received from the card. Values that are received from other roles are typically referenced as variables (`var`). `Nonce` is a frequently used standard type in Scyther. `Fresh` in Scyther means a freshly generated random value and `const` references a constant value. `Send` and `Receive` events have been added to each role which correspond to the messages exchanged during the communication. In this case, the UID is sent from the card in plain text.

As discussed in Chapter 2, once the UID is transmitted and the selection process is done, the reader issues an authentication challenge request command to the card. This step has been omitted from the below listing as no unique information is transmitted due to being a standard command; however, the rest of the steps exchanging the encrypted random numbers have been added to the model as send

and receive events. The rotation function has been globally declared as a `usertype` (user-defined type in Scyther) and has been used in the send and receive events to simulate the sending and receiving of the rotated random numbers. Encrypted messages with symmetric keys are within curly braces, and in general, `k(X,Y)` denotes the long-term symmetric key shared between X and Y [41]. In this case, the card and the reader share a secret key.

The `claims` specify what aspects of the protocol need to be checked by Scyther. There is an option to auto-verify possible claims, automatically adding authentication and secrecy claims at the end of each role [48]. The auto-verify option was used during both experiments.

Listing 1: Experiment 1 - Scyther - UID no encryption

```
usertype Rotated; //globally defining the rotation function
protocol RFIDUID(Card,Reader)
{
    role Card
    {
        const UID: Nonce;
        fresh RndB: Nonce;
        var RndA: Nonce;
        // Send unencrypted UID to the reader along with random
            number B for authentication challenge
        send_1(Card,Reader,(UID,{RndB}k(Card,Reader)));
        // Receive rotated random number B and a freshly
            generated random number A from reader
        recv_2(Reader,Card,{Rotated(RndB),RndA}k(Card,Reader));
        // Send rotated number A back to reader
        send_3(Card,Reader,{Rotated(RndA)}k(Card,Reader));
        // Rotate RndB back. If new RndB==original RndB,
            Authentication successful
        }
    role Reader
    {
        var UID : Nonce;
        var RndB: Nonce;
        fresh RndA: Nonce;
        // Reader waits for the card to send its UID
        // Receive the response UID from the card and random
            number B
        recv_1(Card,Reader,(UID,{RndB}k(Card,Reader)));
        // Send rotated random number B and a freshly generated
            random number A to card
        send_2(Reader,Card,{Rotated(RndB),RndA}k(Card,Reader));
        // Receive rotated number A from card
        recv_3(Card,Reader,{Rotated(RndA)}k(Card,Reader));
        // Rotate RndA back. If new RndA==original RndA,
            Authentication successful
    }
}
```

3.1.2 *Experiment 2 - Implementing Random UID*

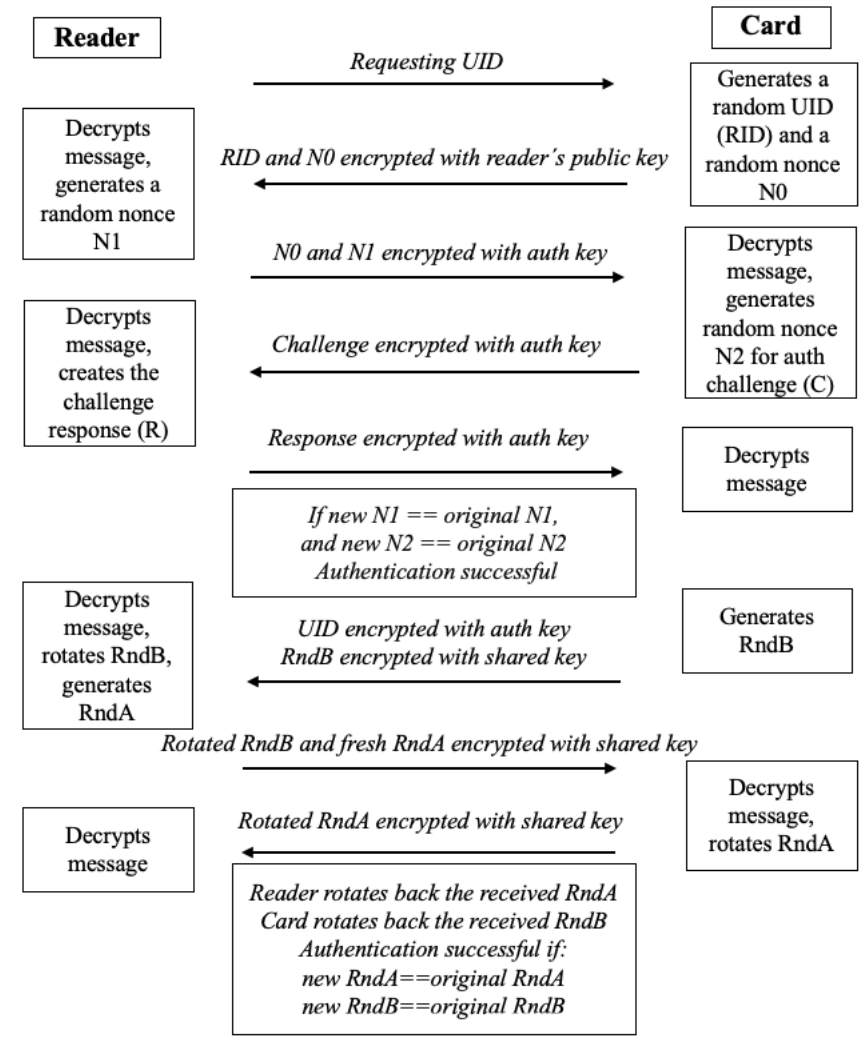

Figure 2: Experiment 2 - Transmission of RID, encrypted UID and three-pass authentication (the proposed solution)

In Experiment 2 (Figure 2), a long-term shared authentication key has been declared globally as `usertype`. This key is only used when the RID is activated and is not used to establish Secure Messaging. Its purpose is to let the card and reader mutually authenticate each other by proving they possess the right shared key before the actual UID can be transmitted. It is a standard convention in Scyther to define `macros`, which are used to abbreviate complex messages or repeated elements [41]. Two macros have been declared globally, C, which stands for a challenge, and R, which stands for a response. Then, as shown in Experiment 1, the protocol and roles are defined. The random UID is defined as a fresh nonce, while the UID is a constant. Random numbers N0, N1, and N2 are fresh/variable nonces. The Random ID and a freshly generated random number encrypted with the

reader's public key are sent from the card to the reader. The public-key infrastructure is predefined in Scyther where the long-term private key of an agent is referenced as `sk(X)` and the corresponding public key is `pk(X)`. An encrypted message with X's public key can only be decrypted by an agent who knows the secret key of X [41]. Then, the flow of messages used for the first mutual authentication in Listing 2 follows the logic of [49]: "Symmetric cryptography provides a degree of authentication because data encrypted with one symmetric key cannot be decrypted with any other symmetric key. Therefore, as long as the symmetric key is kept secret by the two parties using it to encrypt communications, each party can be sure that it is communicating with the other as long as the decrypted messages continue to make sense." Once the card and reader have authenticated each other, the UID is transmitted in an encrypted channel. This is followed by the same three-pass authentication protocol used in Experiment 1 to establish Secure Messaging. Scyther's auto-verify option was used to verify automatic secrecy and authentication claims for both roles.

Listing 2: Experiment 2 - Scyther - Random UID with public key

```
usertype kAuth; //long-term shared key between card and reader
    used for authentication purpose only when RID is activated
macro C = {N1,N2}kAuth(Card,Reader); // Challenge: N1 and N2
    concatinated and encrypted using the secret key
macro R = {N2,N1}kAuth(Card,Reader); // Response: N2 and N1
    concatinated and encrypted using the secret key
usertype Rotated; //references the same rotation function as in
    Experiment 1

protocol RFIDrandomID(Card,Reader)
{
    role Card
    {
        fresh RID : Nonce;
        fresh N0 : Nonce;
        fresh N2 : Nonce;
        var N1 : Nonce;
        const UID: Nonce;
        fresh RndB: Nonce;
        var RndA: Nonce;

        // Send the card's freshly generated RID and a nonce N0
            encrypted with the reader's public key to the reader
        send_1(Card,Reader,{RID,N0}pk(Reader));
        // Receive N0 and the reader's nonce N1
        recv_2(Reader,Card, {N0,N1}kAuth(Card,Reader));
        // Send C to the reader
        send_3(Card,Reader,C);
        // Receive R from the reader
        recv_4(Reader,Card,R);
        //Card decrypts R
```

```
        //if new N2==original N2, Authentication successful
        // Send UID to the reader encrypted with auth key, and
            random number B (same as in Experiment 1)
        send_5(Card,Reader,({UID}kAuth(Card,Reader),{RndB}k(Card,
            Reader)));
        // Same three pass authentication as in Experiment 1
        recv_6(Reader,Card,{Rotated(RndB),RndA}k(Card,Reader));
        send_7(Card,Reader,{Rotated(RndA)}k(Card,Reader));
        }
    role Reader
    {
        var RID: Nonce;
        var N0 : Nonce;
        fresh N1 : Nonce;
        var N2 : Nonce;
        var UID : Nonce;
        var RndB: Nonce;
        fresh RndA: Nonce;

        // Reader waits for the card to send its RID
        // Receive the card's RID and nonce N0 encrypted with the
             reader's public key
        recv_1(Card,Reader,{RID,N0}pk(Reader)); //this can be
            decrypted with the reader's private key
        // Send received N0, fresh random nonce N1 to the card
        send_2(Reader,Card, {N0,N1}kAuth(Card, Reader));
        // Receive C from the card
        recv_3(Card,Reader, C);
        //Reader decrypts C
        //if new N1 == original N1, Authentication successful
        // Send R to the card
        send_4(Reader,Card,R);
        // Receive the response UID from the card and random
            number B
        recv_5(Card,Reader,({UID}kAuth(Card,Reader),{RndB}k(Card,
            Reader)));
        // Same three pass authentication as in Experiment 1
        send_6(Reader,Card,{Rotated(RndB),RndA}k(Card,Reader));
        recv_7(Card,Reader,{Rotated(RndA)}k(Card,Reader));
    }
}
```

4 RESULTS

As discussed in Chapter 2, in Scyther, it is assumed that the cryptography is perfect, meaning that a message can only be decrypted by someone who has the correct key, and schemes cannot be cracked. Furthermore, the messages are considered abstract terms, and the intruder can only learn the entire content of a message if he has the right key or learns nothing. Analysis based on modeling these abstractions instead of modeling all cryptographic details is referred to as "black box" analysis [47]. Before analyzing the experiments' results, it is essential to note that Scyther was designed to analyze basic security concepts instead of cryptographic details.

Furthermore, as mentioned in Chapter 3, the default Scyther settings have been used during all experiments. However, the "Maximum number of runs" was set to three manually. According to Cremers, the creator of Scyther, [47], when a protocol is executed, each role can be executed several times, and a single execution of a role is referred to as a *run*. More runs would have resulted in extended execution times. Therefore using three runs was preferred.

4.1 EXPERIMENT 1

In Experiment 1, Scyther auto-verify claims for the unencrypted transmission of UID and the three-pass authentication were analyzed.

As shown in Figure 3, the secrecy claims for the UID on both the card's and the reader's end were falsified. This is due to no encryption applied. The secrecy claim is defined by [47]. It refers to certain information not being revealed to an intruder, even though the communication is done over an untrusted network. The UID term should not be known to an intruder and should be a "secret". The falsification of claims constitutes an attack, meaning that the term UID could be known to an intruder. This is especially dangerous because an intruder can read the UID, therefore track UID movements, and is also able to redirect messages containing the UID, which could lead to several RFID attacks mentioned in Chapter 2 such as eavesdropping, replay attacks, spoofing or even potentially cloning. In Appendix A, a detailed figure shows the flow of an example attack. In a potential real-life scenario, performing such attacks would require in-depth knowledge of the communication flows, additional devices to sniff and replay messages, as well as low-security requirements from the tag/reader manufacturer's end, but as discussed in Chapter 2, such attacks are not impossible. Recommended countermeasures to pre-

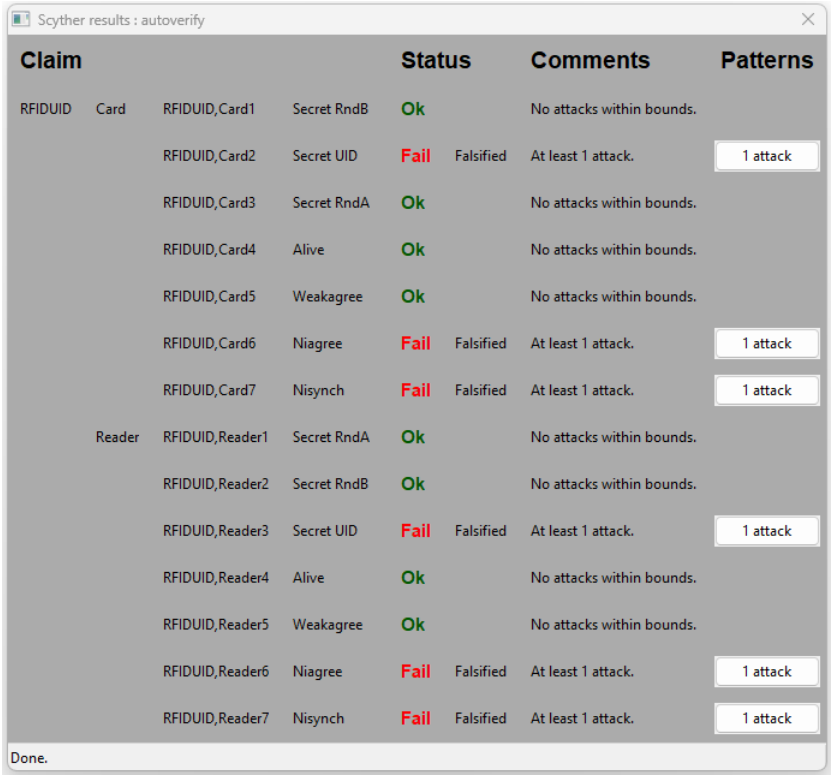

Figure 3: Verification result - Experiment 1 - UID no encryption - Scyther

vent attacks targeting the transponder (card) data are based on cryptography, such as mutual authentication and encrypted data transfer between a card and a reader [50].

Performing mutual authentication between a card and a reader is one of the most important countermeasures to mitigate eavesdropping and replay attacks. In the case of the EV2 cards, it also establishes the encryption method for Secure Messaging (AES recommended) to exchange user or application-specific information.

Secrecy of random numbers generated during the authentication protocol was verified with the status "OK" and comments "No attack within bounds". OK means that the claims are not falsified and are correct, while "No attack within bounds" refers to no attack found within the bounded statespace [41], which in this case was three runs. In other words, it means "no attacks that involve three runs or less" [41].

Aliveness is defined by [47], and it means that during the execution of the role until the claim event, the trusted communicating parties executed their events. In other words, "the intended partners are alive" [47]. For example, in this experiment, if the reader receives a message, it is sure that the card has been active and vice-versa. In the Dolev-Yao intruder model, they can only be sure that an intruder did not generate the message if it is signed with a secret key [47]. For

all claims, "No attacks within bounds" refers to the same bounded statespace as the secrecy claim.

Agreement refers to an extensional security property that aims to consider the effect the protocol achieves [47]. It checks whether the communicating parties agree on the values of the variables after the execution of the protocol. "In a two-party protocol, if two parties agree on the values of all variables, then they agree on the contents of all messages exchanged" [47]. Non-injective agreement (Niagree) verifies that all communication events preceding a claim occurred before the claim, while weak agreement (Weakagree) performs a minimum agreement check. As shown in Figure 3, Weakagree claims were verified for both roles in this experiment, with no attacks found within bounds. However, the Niagree claims were falsified. This is due to an intruder being able to replay previously captured messages, such as the exposed UID, and impersonate a card, as no measures ensure the agreement on values between the agents and the occurrence of events before the claim.

Non-injective synchronization (Nisynch) verifies if everything intended to happen in the protocol description also actually happened [47]. Synchronization is a form of authentication where certain conditions need to hold to verify that the communication occurred correctly. These conditions are: correct order of events, messages communicated correctly (meaning that the content of a received message is the same as the content of the corresponding sent message), and run events correspond to the correct send and receive events from the protocol and that they are part of the correct runs. The Nisynch claim was falsified. Similarly to Niagree, there are no measures implemented in the communication that ensure that messages are received in the correct order or that the correct messages are received. It is also important to note a significant difference between agreement and synchronization. Synchronization requires the send and receive events to be executed in the correct order, while agreement does not check the expected order of the messages.

4.2 EXPERIMENT 2

As Experiment 1 highlighted, exposing the card's physical UID can have serious consequences where the intruder can disturb the synchronization of messages and redirect the unencrypted UID. Therefore, in Experiment 2, an important security feature of DESFire cards has been modeled, the implementation of random UID (RID). The main idea behind using RID is to prevent tracking the owner of the card, as well as prevent replay attacks.

Adding a fresh nonce while sending the RID and using public key encryption protect the secrecy of the RID and ensure that all

Scyther results : autoverify

Claim				Status	Comments
RFIDrandomID	Card	RFIDrandomID,Card1	Secret N0	Ok	No attacks within bounds.
		RFIDrandomID,Card2	Secret RndB	Ok	No attacks within bounds.
		RFIDrandomID,Card3	Secret UID	Ok	No attacks within bounds.
		RFIDrandomID,Card4	Secret N2	Ok	No attacks within bounds.
		RFIDrandomID,Card5	Secret RID	Ok	No attacks within bounds.
		RFIDrandomID,Card6	Secret RndA	Ok	No attacks within bounds.
		RFIDrandomID,Card7	Secret N1	Ok	No attacks within bounds.
		RFIDrandomID,Card8	Alive	Ok	No attacks within bounds.
		RFIDrandomID,Card9	Weakagree	Ok	No attacks within bounds.
		RFIDrandomID,Card10	Niagree	Ok	No attacks within bounds.
		RFIDrandomID,Card11	Nisynch	Ok	No attacks within bounds.
	Reader	RFIDrandomID,Reader1	Secret RndA	Ok	No attacks within bounds.
		RFIDrandomID,Reader2	Secret N1	Ok	No attacks within bounds.
		RFIDrandomID,Reader3	Secret N0	Ok	No attacks within bounds.
		RFIDrandomID,Reader4	Secret RndB	Ok	No attacks within bounds.
		RFIDrandomID,Reader5	Secret UID	Ok	No attacks within bounds.
		RFIDrandomID,Reader6	Secret N2	Ok	No attacks within bounds.
		RFIDrandomID,Reader7	Secret RID	Ok	No attacks within bounds.
		RFIDrandomID,Reader8	Alive	Ok	No attacks within bounds.
		RFIDrandomID,Reader9	Weakagree	Ok	No attacks within bounds.
		RFIDrandomID,Reader10	Niagree	Ok	No attacks within bounds.
		RFIDrandomID,Reader11	Nisynch	Ok	No attacks within bounds.

Done.

Figure 4: Verification result - Experiment 2 - Random ID - Scyther

exchanged messages occur during the communication and in the correct order.

As shown in Figure 4, all auto-verify claims were verified with "No attacks within bounds" for Experiment 2, which proves that implementation of RID could be a very important countermeasure to prevent many RFID attacks mentioned in Chapter 2 and is a significant improvement when it comes to protecting the privacy of the cardholder.

As highlighted in Chapter 2, there is an additional verification option in Scyther called Characterize roles. It verifies whether the communicating parties are reachable by checking the correctness of the protocol. Figure 5 shows that both the card and the reader are reachable, and there is precisely one trace pattern for both roles.

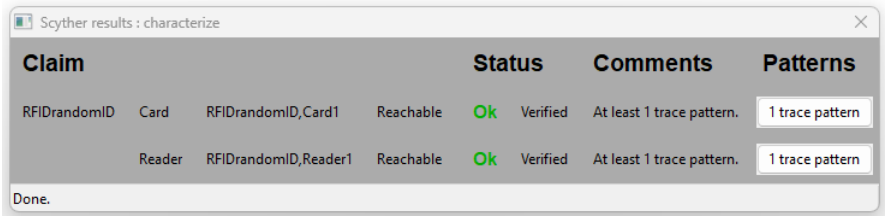

Figure 5: Verification result - Characterize Roles - Experiment 2 - Random ID - Scyther

5 DISCUSSION

5.1 CHALLENGES, DISCUSSION AND FUTURE WORK

While Scyther is a tool that has successfully been used in theoretical research to analyze and design protocols [48], there are some open challenges regarding the tool and the proposed protocol, which will be discussed in this section.

5.1.1 *Cryptographic Primitives*

One of the main limitations of automatic protocol verification in Scyther is that an assumption is made that the cryptography is flawless [47]. In reality, this may not be the case. For example, side channel attacks can be performed to crack keys that are generally considered to be secure, such as AES keys [51].

5.1.1.1 *Side Channel Attacks*

As highlighted in Chapter 2, the DESFire EV2 and 3 cards are considered safe tags. Both utilize AES to protect the confidentiality and integrity of transmitted data. In general, AES is considered safe against brute force attacks and would potentially require quantum computing to crack a 128-bit key [52]. However, fast side-channel attacks to crack AES keys, such as Electromagnetic (EM) Side Channel attacks [53] or Correlative Electromagnetic Side Channel attacks (CEMA) [54], continue to emerge. EM radiation serves as the foundation for a variety of wireless communication technologies. However, it is widely known that electronic devices emit EM radiation unintentionally on frequencies (for example, when performing cryptographic calculations) that are not intended as a byproduct of their internal functioning [53]. The experiment by [54] overcomes shortcomings of traditional side-channel attacks and significantly reduces the cracking time. The authors highlighted that their experiment worked on embedded encryption chips and that cracking hardware encryption chips is still facing difficulties. It is crucial to consider these attacks, research them and implement countermeasures by design in wireless communication as technology advances.

5.1.2 *TRNG - True Random Number Generator*

Due to the requirement of random numbers in the proposed solution, one can wonder about the true randomness of these random values. As mentioned in Chapter 2, the DESFire cards have a True Random Number Generator (TRNG) hardware component. TRNGs utilize a wide range of entropy sources as their primary component while exhibiting several common attributes. They do not depend on seeding to produce randomness and derive entropy from a natural phenomenon [26]. TRNGs must be tested as they play an essential role in cryptographic operations. Common standards for testing Random Number Generators (RNG) are specified by the National Institute of Standards and Technology (NIST) and Common Criteria (CC). SP800-22 (Special Publication by NIST) details an extensive test battery suitable for use over TRNG. At the same time, CC employs a broader verification scheme for the security of the whole system, which includes RNG testing [26]. The authors of [26] experimented with DESFire EV1 cards where 64MB of data was collected during the card-reader authentication protocol. The collected data was then tested in the NIST SP800-22 framework, and several other statistical tests were performed. The experiments highlighted poor X^2 results which meant that "the values in the tested sequences are not uniformly distributed: there is a bias towards some byte values and away from others" [26]. The absence of a uniform distribution of bytes does not necessarily signify a lack of randomness. While it is not an ideal indication of randomness, it is feasible for a genuine source of randomness to generate a slightly skewed sequence. Researchers investigated this issue, and the concerns were disclosed to NXP [26], [55].

The authors also confirmed that further investigation into the EV2 has revealed no problems with its Random Number Generator.

5.1.3 *Secure Key Exchange*

The proposed approach presumes securely exchanging symmetric keys between the card and the reader prior to reading the card during mutual authentication. Several researchers have suggested Diffie-Hellman-based elliptic curve key exchange for IoT applications [56] [57] [58]. Elliptic curve cryptography is considered more secure and efficient than traditional public-key cryptography methods such as RSA. It offers the same level of security with shorter key lengths, making it ideal for use in constrained environments such as mobile devices and embedded systems [58]. Elliptic-Curve Diffie-Hellman (ECDH) key exchange allows two parties with no former contact to establish a shared secret key over an insecure channel using the mathematical properties of elliptic curves [58]. Implementing ECDH could

be researched as part of initially writing a card to establish the shared secret keys used when reading cards.

5.1.4 *Future Work*

Along with measures preventing side-channel attacks and implementing secure key exchange, the real-life testing, evaluation, implementation, and engineering of the proposed protocol in Experiment 2 utilizing RID could be an area of future research.

The experiments in this thesis focused on preventing attacks by improving the security of RFID communication instead of detection. Intrusion detection frameworks have been proposed by previous research and can be used for monitoring securely designed systems for intrusions. A survey [59] provides an excellent overview of recent approaches for intrusion detection in the Internet of Things (IoT). These intrusion detection techniques include Machine Learning, Deep Learning, Blockchain-based solutions, and Intrusion detection systems (IDS) for specific types of attacks, for example, Denial of Service (DoS). To mention some of the other newer approaches to intrusion detection in different IoT systems, [60] proposed an Anomalous intrusion detection protocol for wireless sensor networks. In contrast, [61] introduced a method to detect and isolate cyber attacks in industrial control systems. However, as summarized by [59], the manufacturers of smart devices used in IoT networks primarily focus on user-friendliness, low energy consumption, and lower computational costs. This leaves the security of these devices behind and introduces security vulnerabilities. Therefore, along with detection measures, manufacturers and researchers should shift the focus to implementing security by design in resource-sensitive environments.

5.1.4.1 *Computational Challenges*

[62] compares several lightweight and ultra-lightweight RFID authentication schemes and their potential vulnerabilities to different types of attacks. The compared authentication schemes include protocols utilizing XORing, rotation, concatenation [63], [64], [65], and protocols utilizing elliptic curve cryptography and hash functions [66], [67], [68]. According to [62], most compared schemes including hashing and elliptic curve cryptography require storage/memory between 288-672 bits, and the communication overhead is between 192-1440 bits to perform the required operations. Simpler operations such as XORing or rotating require less resources than hash functions or elliptic curve cryptography [62]. Regardless of the operations performed, most authentication schemes were vulnerable to desynchronization, tracking, or replay attacks [62]. These attacks, though, could be prevented by utilizing RID. As shown in Experiment 2, implementing RID requires additional communication steps, which could increase

the memory requirements and communication overhead. According to the DESFire EV3 datasheet [29], the communication buffer for EV2 is 128 bytes (1024 bits), and for EV3 is 256 bytes (2048 bits). The size of these memory buffers is suitable to perform most authentication protocols in [62]. However, the requirements should be measured for implementing RID as well. [69] compares the technical specifications of several types of available smart cards, including the MIFARE DESFire family. The authors in [69] performed multiple cryptography operations with these cards, such as generating hashes or encrypting messages with symmetric keys, and measured the average time of operations taken in ms. According to their performance assessment, asymmetric cryptography operations take significantly longer than symmetric operations [69]. However, asymmetric cryptography is also considered feasible, as most modern RFID smart cards support hash functions and asymmetric cryptography operations, including elliptic curve cryptography [69]. The average time of operations is an interesting research direction that could be explored in the future for the protocol utilizing RID to compare it with existing protocols and confirm feasibility. Potential optimization measures could also be considered as a future direction. [70] propose a scalable and secure hash-based RFID protocol that utilizes a time-memory trade-off. The authors in [70] show that hash functions could reduce the amount of computations in the system and enhance forward privacy and secrecy. Elliptic curve cryptography could be used to optimize the public-key cryptography proposed in Experiment 2. In addition, authors in [71] prove that their elliptic curve cryptography-based RFID authentication scheme is a feasible option to protect against several RFID attacks, such as tracking, cloning, or replay attacks. They also conducted a performance analysis to measure memory requirements to store the private and public keys and computational costs and overhead [71], which highlighted that while elliptic curve cryptography is preferred due to small key sizes and more efficient computations than other public-key algorithms, they also concluded that more secure elliptic curve cryptography based protocols have higher computational costs and memory requirements than previous elliptic curve based approaches such as [72], [73], [74]. Previous studies in this section show that elliptic curve cryptography could be feasible for RFID systems and that hashing could optimize performance. A thorough energy and computational analysis should be carried out to measure and improve the performance of the proposed protocol in Experiment 2 as an area of future research.

5.2 REFLECTION ON RESEARCH QUESTIONS

In Chapter 1, multiple research questions were raised, and throughout this thesis, the aim was to answer them. In this section, a reflection on these questions is provided.

How can RFID communication be evaluated as an intrusion detection point? Chapter 2 highlighted that intruders have several ways to attack RFID systems. Previous research has concluded that most cyber threats targeting RFID systems can be associated with the lack of security of the communication channel between the reader and the tag or flaws in the implemented authentication protocols and encryption algorithms. The security of the communication can be evaluated and was, in fact, evaluated in this thesis using "black box" abstract modeling of the messages in the communication using Scyther.

Is the communication protocol between an RFID reader and tag secure? Experiment 1 revealed that the construction of the selection and three-pass authentication protocols where the card's UID is transmitted unencrypted has flaws that allow intruders to read the UID, track UID movements, and redirect messages containing the UID.

Can the security of the protocol be improved? In Experiment 2, the implementation of Random UID was proposed and modeled in Scyther. This included adding fresh nonces to messages and using public key cryptography to ensure that all messages occur as intended during the communication and in the correct order. Theoretically, this gives no opportunity for intruders to intercept, replay or de-synchronize messages under the Dolev-Yao adversary model. The Scyther verification results indicated "No attacks found within bounds" for the proposed protocol.

6 CONCLUSION

6.1 CONCLUSION

A formal security analysis was performed to evaluate the security of the RFID communication, as it can often be located and occur outside of secure premises and be exposed to intruders. Previous research highlighted that most RFID attacks result from the lack of security of the communication channel or flaws in the implemented authentication protocols. Therefore, these protocols were modeled in Scyther.

Two experiments were performed to formally analyze the RFID selection and authentication protocols. In the first experiment, the UID was transmitted without encryption, which could lead to RFID attacks such as eavesdropping, replay attacks, spoofing, or even cloning. In the second experiment, a commercially available security feature, Random UID, was modeled with additional asymmetric encryption and a further mutual authentication step. According to the verification results of Scyther, no attacks were found on the logic and construction of the protocol using RID. However, as an area of future work, the real-life implementation and optimization of the proposed protocol could be explored, and accurate measurements and a detailed computational analysis could be performed.

APPENDIX - SCYTHER - ATTACK EXAMPLE

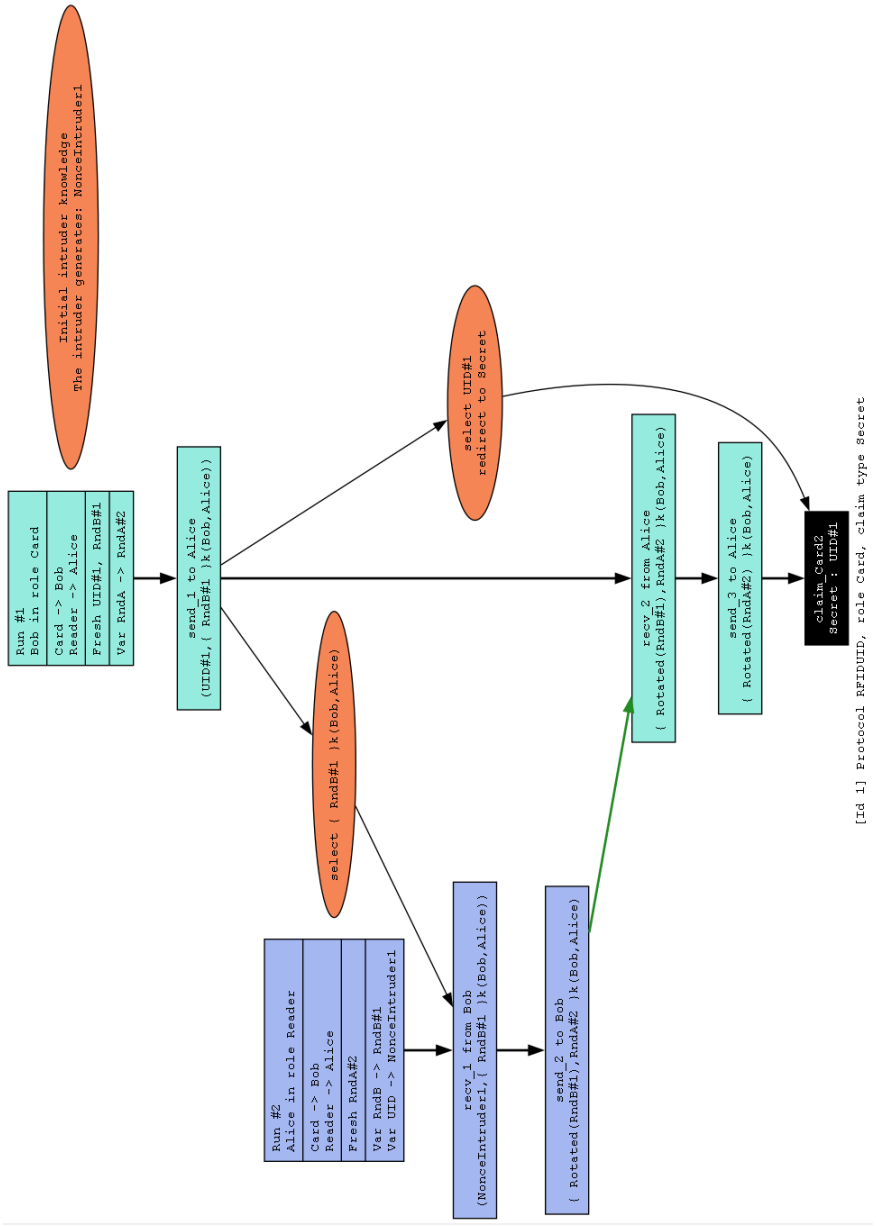

Figure 6: Attack - Experiment 1 - UID no encryption Secrecy- Scyther

BIBLIOGRAPHY

[1] Yassine NAIJA. Secured digital architectures for low cost full-fledged hf rfid tags - national engineering school of sousse.

[2] Chisel Labs. What is end-to-end? URL: https://chisellabs.com/glossary/what-is-end-to-end/.

[3] Will Kenton. What is end-to-end? a full process, from start to finish. URL: https://www.investopedia.com/terms/e/end-to-end.asp.

[4] Jean-Paul A Yaacoub, Ola Salman, Hassan N Noura, Nesrine Kaaniche, Ali Chehab, and Mohamad Malli. Cyber-physical systems security: Limitations, issues and future trends. *Microprocessors and microsystems*, 77:103201, 2020.

[5] Vanderbilt Engineering Graduate Admissions Team. What is the difference between cps and iot? URL: https://blog.engineering.vanderbilt.edu/what-is-the-difference-between-cps-and-iot.

[6] Sharon Shea Sarah Amsler. Rfid (radio frequency identification). URL: https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-identification.

[7] Xiaolin Jia, Quanyuan Feng, Taihua Fan, and Quanshui Lei. Rfid technology and its applications in internet of things (iot). In *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pages 1282–1285, 2012. doi: 10.1109/CECNet.2012.6201508.

[8] Amit Grover and Hal Berghel. A survey of rfid deployment and security issues. *Journal of information processing systems*, 7(4):561–580, 2011.

[9] Timo Kasper, Ingo von Maurich, David Oswald, and Christof Paar. Cloning cryptographic rfid cards for 25. In *5th Benelux workshop on information and system security. Nijmegen, Netherlands*, 2010.

[10] Hong Li, YongHui Chen, and ZhangQing He. The survey of rfid attacks and defenses. In *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4. ieee, 2012.

[11] Tiago M Fernández-Caramés, Paula Fraga-Lamas, Manuel Suárez-Albela, and Luis Castedo. Reverse engineering and security evaluation of commercial tags for rfid-based iot applications. *Sensors*, 17(1):28, 2016.

[12] Wolfgang Issovits and Michael Hutter. Weaknesses of the iso/iec 14443 protocol regarding relay attacks. In *2011 IEEE International Conference on RFID-Technologies and Applications*, pages 335–342. IEEE, 2011.

[13] ISO. Iso home page. URL: https://www.iso.org/home.html.

[14] IEC. What we do. URL: https://iec.ch/what-we-do.

[15] nfc tools. Iso14443. URL: https://nfc-tools.github.io/resources/standards/iso14443/.

[16] David Coelho. Mifare desfire - an introduction. URL: https://www.linkedin.com/pulse/mifare-desfire-introduction-david-coelho/.

[17] IBM Corporation. Des key types. URL: https://www.ibm.com/docs/en/zos/2.2.0?topic=keys-des-key-types.

[18] IBM Corporation. Aes key types. URL: https://www.ibm.com/docs/en/zos/2.2.0?topic=keys-aes-key-types.

[19] NXP Semiconductors. Mifare. URL: https://www.nxp.com/products/rfid-nfc/mifare-hf:MC_53422.

[20] NXP Semiconductors. Mifare desfire. URL: https://www.nxp.com/products/rfid-nfc/mifare-hf/mifare-desfire:MC_53450.

[21] NIST. Update to current use and deprecation of tdea. URL: hhttps://csrc.nist.gov/News/2017/Update-to-Current-Use-and-Deprecation-of-TDEA.

[22] NXP Semiconductors. Mf3d(h)x2 mifare desfire ev2 contactless multi-application ic. URL: https://www.nxp.com/docs/en/data-sheet/MF3DX2_MF3DHX2_SDS.pdf.

[23] NXP. Mifare desfire ev2 security target lite. URL: https://www.ssi.gouv.fr/uploads/2016/05/mf3dx2_mifare_desfireev2_securitytargetlite_v1.5.pdf.

[24] HID. Mifare desfire ev3 application note. URL: https://www.hidglobal.com/sites/default/files/documentlibrary/plt-05618-a.0-mifare-desfire-ev3-application-note.pdf.

[25] NXP Semiconductors. Mifare product and handling of uids. URL: https://www.nxp.com/docs/en/application-note/AN10927.pdf.

[26] Darren Hurley-Smith and Julio Hernandez-Castro. Challenges in certifying small-scale (iot) hardware random number generators. *Security of Ubiquitous Computing Systems: Selected Topics*, pages 165–181, 2021.

[27] Katie Moss Jefcoat. What is common criteria certification, and why is it important? URL: https://www.blancco.com/resources/blog-what-is-common-criteria-certification-why-is-it_important.

[28] Trustonic. What eal5+ is, and why receiving it is such a significant achievement. URL: https://www.trustonic.com/opinion/what-eal5-is-and-why-receiving-it-is-such-a-significant_achievement/.

[29] NXP Semiconductors. Mf3d(h)x3 mifare desfire ev3 contactless multi-application ic. URL: https://www.nxp.com/docs/en/data-sheet/MF3DHx3_SDS.pdf.

[30] Santiago Figueroa Lorenzo, Javier Añorga Benito, Pablo García Cardarelli, Jon Alberdi Garaia, and Saioa Arrizabalaga Juaristi. A comprehensive review of rfid and bluetooth security: Practical analysis. *Technologies*, 7(1):15, 2019.

[31] Proxmark. Proxmark3. URL: https://proxmark.com/proxmark-3-hardware/proxmark-3.

[32] Dominic Celiano. *Overclocking Proximity Checks in Contactless Smartcards*. PhD thesis, Masterâs thesis, University of Cambridge, 2018.

[33] Gildas Avoine, Ioana Boureanu, David Gérault, Gerhard P Hancke, Pascal Lafourcade, and Cristina Onete. From relay attacks to distance-bounding protocols. *Security of Ubiquitous Computing Systems: Selected Topics*, pages 113–130, 2021.

[34] Antonio Muñoz, Carmen Fernández-Gago, and Roberto López-Villa. A test environment for wireless hacking in domestic iot scenarios. *Mobile Networks and Applications*, pages 1–10, 2022.

[35] CVE. Cve program mission. URL: https://www.cve.org.

[36] VVX7. Your amiiboâs haunted. URL: https://vvx7.io/posts/2022/09/your-amiibos-haunted/.

[37] NIST. Cve-2021-33881 detail. URL: https://nvd.nist.gov/vuln/detail/CVE-2021-33881.

[38] Gallagher Group. Cve-2020-16097. URL: https://security.gallagher.com/en-GB/Security-Advisories/CVE-2020-16097.

[39] Matthias Deeg. [syss-2019-005]: Abus secvest - proximity key - cryptographic issues (cwe-310). URL: https://seclists.org/fulldisclosure/2019/May/3.

[40] Katharina Pfeffer. Formal verification of a lte security protocol for dual-connectivity: An evaluation of automatic model checking tools, 2014.

[41] Cas Cremers. Scyther user manual. URL: https://github.com/cascremers/scyther/blob/master/gui/scyther-manual.pdf.

[42] Jonathan Herzog. A computational interpretation of dolev–yao adversaries. *Theoretical Computer Science*, 340(1):57–81, 2005.

[43] Mohamed Eldefrawy, Ismail Butun, Nuno Pereira, and Mikael Gidlund. Formal security analysis of lorawan. *Computer Networks*, 148:328–339, 2019.

[44] Mehdi Hosseinzadeh, Omed Hassan Ahmed, Sarkar Hasan Ahmed, Cuong Trinh, Nasour Bagheri, Saru Kumari, Jan Lansky, and Bao Huynh. An enhanced authentication protocol for rfid systems. *IEEE Access*, 8:126977–126987, 2020.

[45] Sarah Abu Ghazalah. *Mutual Authentication Protocols for RFID Special Schemes*. PhD thesis, Royal Holloway, University of London, Egham, UK, 2016.

[46] Xiong Wang, Yuan Teng, Yaping Chi, and Hongbo Hu. A robust and anonymous three-factor authentication scheme based ecc for smart home environments. *Symmetry*, 14(11):2394, 2022.

[47] Casimier Joseph Franciscus Cremers et al. *Scyther: Semantics and verification of security protocols*. Eindhoven university of Technology Eindhoven, Netherlands, 2006.

[48] Cas JF Cremers. The scyther tool: Verification, falsification, and analysis of security protocols: Tool paper. In *Computer Aided Verification: 20th International Conference, CAV 2008 Princeton, NJ, USA, July 7-14, 2008 Proceedings 20*, pages 414–418. Springer, 2008.

[49] IBM Corporation. Symmetric cryptography. URL: https://www.ibm.com/docs/en/ztpf/2020?topic=concepts-symmetric-cryptography.

[50] Klaus Finkenzeller. Known attacks on rfid systems, possible countermeasures and upcoming standardisation activities. In *5th European Workshop on RFID Systems and Technologies*, pages 1–31, 2009.

[51] Nicky Mouha and Morris Dworkin. Review of the advanced encryption standard. *Technical report, National Institute of Standards and Technology*, 2021.

[52] Govindraj Basatwar. Understanding aes-128 encryption and its significance in the current threat landscape. URL: https://www.appsealing.com/aes-128-encryption/.

[53] Asanka Sayakkara, Nhien-An Le-Khac, and Mark Scanlon. A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics. *Digital Investigation*, 29:43–54, 2019.

[54] Wen-hai Zhou and Fan-tong Kong. Electromagnetic side channel attack against embedded encryption chips. In *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, pages 140–144, 2019. doi:10.1109/ICCT46805.2019.8947185.

[55] Darren Hurley-Smith and Julio Hernandez-Castro. Certifiably biased: An in-depth analysis of a common criteria eal4+ certified trng. *IEEE Transactions on Information Forensics and Security*, 13(4):1031–1041, 2017.

[56] HPTM Jayawardana and RL Dangalla. Hybrid encryption protocol for rfid data security. In *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pages 1209–1212. IEEE, 2020.

[57] Mustapha Benssalah, Izza Sarah, and Karim Drouiche. An efficient rfid authentication scheme based on elliptic curve cryptography for internet of things. *Wireless Personal Communications*, 117(3):2513–2539, 2021.

[58] Ravi Kishore Kodali and Ashwitha Naikoti. Ecdh based security model for iot using esp8266. In *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pages 629–633, 2016. doi:10.1109/ICCICCT.2016.7988026.

[59] Shakir Zaman, Haseeb Tauqeer, Wakeel Ahmad, Syed M Adnan Shah, and Muhammad Ilyas. Implementation of intrusion detection system in the internet of things: A survey. In *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pages 1–6. IEEE, 2020.

[60] Rajkumar Krishnan, R Santhana Krishnan, Y Harold Robinson, E Golden Julie, Hoang Viet Long, A Sangeetha, M Subramanian, and Raghvendra Kumar. An intrusion detection and prevention protocol for internet of things based wireless sensor networks. *Wireless Personal Communications*, 124(4):3461–3483, 2022.

[61] Anna Sztyber-Betley, Michał Syfert, Jan Maciej Kościelny, and Zuzanna Górecka. Controller cyber-attack detection and isolation. *Sensors*, 23(5):2778, 2023.

[62] Atul Kumar, Ankit Kumar Jain, and Mohit Dua. A comprehensive taxonomy of security and privacy issues in rfid. *Complex & Intelligent Systems*, 7:1327–1347, 2021.

[63] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan ME Tapiador, and Arturo Ribagorda. Advances in ultralightweight cryptography for low-cost rfid tags: Gossamer protocol. In *Information Security Applications: 9th International Workshop, WISA 2008, Jeju Island, Korea, September 23-25, 2008, Revised Selected Papers 9*, pages 56–68. Springer, 2009.

[64] Xu Zhuang, Yan Zhu, and Chin-Chen Chang. A new ultralightweight rfid protocol for low-cost tags: R 2 ap. *Wireless Personal Communications*, 79(3):1787–1802, 2014.

[65] Umar Mujahid, Muhammad Najam-ul Islam, and Shahzad Sarwar. A new ultralightweight rfid authentication protocol for passive low cost tags: Kmap. *Wireless Personal Communications*, 94:725–744, 2017.

[66] Zhenguo Zhao. A secure rfid authentication protocol for healthcare environments using elliptic curve cryptosystem. *Journal of medical systems*, 38:1–7, 2014.

[67] Eun-Kyung Ryu, Dae-Soo Kim, and Kee-Young Yoo. On elliptic curve based untraceable rfid authentication protocols. In *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pages 147–153, 2015.

[68] Han Shen, Jian Shen, Muhammad Khurram Khan, and Jong-Hyouk Lee. Efficient rfid authentication using elliptic curve cryptography for the internet of things. *Wireless personal communications*, 96:5253–5266, 2017.

[69] Lukas Malina, Petr Dzurenda, Jan Hajny, and Zdenek Martinasek. Assessment of cryptography support and security on programmable smart cards. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–5, 2018. doi:10.1109/TSP.2018.8441334.

[70] Gildas Avoine and Philippe Oechslin. A scalable and provably secure hash-based rfid protocol. In *Third IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 110–114. IEEE, 2005.

[71] Yi-Pin Liao and Chih-Ming Hsiao. A secure ecc-based rfid authentication scheme integrated with id-verifier transfer protocol. *Ad hoc networks*, 18:133–146, 2014.

[72] Xinglei Zhang, Linsen Li, Yue Wu, and Quanhai Zhang. An ecdlp-based randomized key rfid authentication protocol. In *2011 international conference on network computing and information security*, volume 2, pages 146–149. IEEE, 2011.

[73] Pim Tuyls and Lejla Batina. Rfid-tags for anti-counterfeiting. In *Topics in Cryptology–CT-RSA 2006: The Cryptographersâ Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2005. Proceedings*, pages 115–131. Springer, 2006.

[74] Lejla Batina, Jorge Guajardo, Tim Kerins, Nele Mentens, Pim Tuyls, and Ingrid Verbauwhede. Public-key cryptography for rfid-tags. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*, pages 217–222. IEEE, 2007.

.

PO Box 823, SE-301 18 Halmstad
Phone: +35 46 16 71 00
E-mail: registrator@hh.se
www.hh.se