Contributions to the security of EPC/RFID wireless technologies

Joaquin Garcia-alfaro

MEMOIRE
D'HABILITATION A DIRIGER DES RECHERCHES

présenté à

L'UNIVERSITE PIERRE ET MARIE CURIE

par

Joaquin GARCIA-ALFARO

25 NOVEMBRE 2013

Institut MINES-TELECOM / TELECOM SudParis / dép. RST

CONTRIBUTIONS TO THE SECURITY OF EPC/RFID WIRELESS TECHNOLOGIES

RAPPORTEURS / REFEREES :

Giuseppe Bianchi, Professeur, University of Roma, Italy
Olivier Festor, Professeur, Télécom Nancy, France
Javier Lopez, Professeur, University of Malaga, Spain

EXAMINATEURS / EXAMINERS :

Frédéric Cuppens, Professeur, Télécom Bretagne, France
Hervé Debar, Professeur, Télécom SudParis, France
Pierre Sens, Professeur, Université Pierre et Marie Curie, France (Président du Jury)

Table of Contents

Chapter 1

Introduction

This habilitation manuscript summarizes a selected list of research contributions in the area of *Wireless Security*. The selected list of contributions is by no means exhaustive. It shall be seen as a general outline to identify my research work methodology in the wider area of *Network Security*. In order to facilitate comprehension, citations of my own work are denoted using a plain, numerical style, e.g., [167]; while other literature efforts are cited using alpha-numerical style, e.g., [Pen55].

The contributions presented in the manuscript are situated in the field of *Network Security*, with a special emphasis on areas related to wireless security, management of policies, detection of attacks, analysis of vulnerabilities, selection of defense countermeasures and enforcement of access and usage control models. Results are grounded on the use of algorithmics, set theory, software modeling, combinatorics, cryptography, and graph theory.

More specifically, the selected contributions rely on current work around the improvement of RFID (Radio Frequency Identification) security and privacy aspects. The focus in on RFID technologies associated to electronic labels under the EPC (Electronic Product Code) standard (hereinafter simply referred as RFID or EPC tags). The EPC is an international

standard that proposes the use of RFID in the supply chain. EPC tags are minimalistic electronic devices that provide serial numbers to identify objects and people. They are designed to balance cost and functionality, and are becoming truly pervasive in wireless network applications, such as Mobile Wireless Ad Hoc Networks (MANETs), Wireless Sensor Networks (WSNs), and Vehicular Ad Hoc Networks (VANETs) [RDT09]. Tags are potentially the targets of attack against their security and this raises major concerns. The objective of this dissertation is to evaluate some of these concerns and report some of our research solutions handling them.

The remainder chapters of the manuscript are organized as follows. Chapter 2 introduces some preliminary state-of-the-art background on the Electronic Product Code (EPC) standard. It also analyzes security threats to the RFID system of the EPC technology and surveys relevant defense countermeasures for major threats. The following two chapters (Chapters 3 and 4) deepen the analysis on the security of Pseudorandom Number Generators (PRNGs) and provide novel designs to address predictability problems on previous efforts in the related literature. Following chapters introduce new series of defense countermeasures based on suitable PRNG designs holding the unpredictability property. Chapter 5 introduces a proactive threshold cryptosystem for EPC tags. Three privacy-preserving solutions grounded on an anonymous secret sharing scheme are presented. The solutions aim at handling the problem of distributing secrets between manufacturers and vendors of EPC labeled objects. Chapter 6 goes further and tackles the problem of flawed designs on protocols that aim at establishing some security properties. A key establishment protocol is presented and verified using an automatic framework for the formal verification of security schemes. Chapter 7 concludes and gives directions for future work.

Chapter 2

Background and Threat Analysis

In this chapter we introduce some preliminary state-of-the-art background on the RFID system of the Electronic Product Code (EPC) standard. We also analyze threats to the security of the exchange of information between RFID readers and tags. We analyze the set of threats according to the methodology proposed by the European Telecommunications Standards Institute (ETSI), and we rank these threats in order of relevance. The results of the analysis are intended for leading further research and developments of security of EPC-based technologies. We also study countermeasures for threats ranked at the critical or major level. We discuss the benefits and drawbacks associated with the surveyed solutions. Parts of this chapter have been previously published in [175, 176, 179, 180].

2.1 Low-cost RFID and the Electronic Product Code

Passive radio frequency identification (RFID) is a wireless communication technology that allows the automatic identification of objects, animals, and persons through radio waves. Passive RFID tags are electronic labels without self-power supply. They are energized by the electromagnetic field of radio frequency (RF) front-end devices (hereinafter referred as RFID readers). The radio spectrum used in RFID systems varies from low-frequency (LF)

and high-frequency (HF) bands (typically 125 kHz and 13.56 MHz) to ultra-high-frequency (UHF) bands (typically 868 MHz in Europe, 915 MHz in North America, and 950 MHz in Japan). Distances from which the RFID tags can be interrogated vary with the frequency band. It may vary from a few centimeters, while using LF and HF, to a few meters, while using UHF. Although no single technology is ideal for all applications [Wan06], most of the modern RFID systems seem to be moving toward increasing the integration of long-distance passive tags into self-organizing wireless applications. This is the case with the modern Electronic Product Code (EPC) Gen2 tags.

The EPC technology originates from the MIT's Auto-ID Center (now called the Auto-ID Labs). It had been further developed by different working groups at EPCglobal Inc. [EPC07]. It is a layered service-oriented architecture to link objects, information, and organizations via Internet technologies. At the lowest layer, an identification system based on passive RFID tags and readers provides the means to access and identify objects in motion. This system possesses two primary interfaces: the Class 1 Generation 2 UHF Air Interface Protocol Standard (Gen2 for short) and Low Level Reader Protocol (LLRP).The former defines the physical and logical requirements for RFID readers (or interrogators) and passive tags (or labels). The latter specifies the air interface and interactions between its instances.

The next layer consists of a middleware composed of several services (such as filtering, fusion, aggregation, and correlation of events) that perform real-time processing of tag event data and collect the identifier of objects interrogated by RFID readers at different time points and locations. Data gathered by sensors, such as temperature and humidity, can also be aggregated at the middleware layer within tag events. The middleware forwards the complete set of events to a local repository where they are persistently stored (e.g., into a relational or XML database). The Reader Protocol (RP) and Reader Management (RM) interfaces define the interactions between a device capable of reading/writing RFID tags and the middleware. The middleware relies on a second interface called Application Level Event (ALE) for interaction with other applications (e.g., repository managers). At the top of the architecture, the EPC Information Services (EPCISs) offer the means to access the data stored in EPC network repositories. These EPCISs are implemented using

standard Web technologies such as the Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL). Two additional services are defined for accessing the EPCIS of a given EPC network by external applications: a lookup service binding object identifiers and EPCISs, called the Object Name Service (ONS) [173]; and a EPC discovery service (EPCDS) to perform searches with high-level semantics (i.e., similar to Web engines for Web page browsing).

Security attacks can target the different services of the EPC network architecture. They may succeed if weaknesses within the underlying technologies are not handled properly. The exchange of information between EPC tags and readers, for example, is carried out via wireless channels that do not posses basic security attributes such as authenticity, integrity, and availability. This situation allows attackers to misuse the RFID service of an EPC network and perform unauthorized activities such as eavesdropping, rogue scanning, cloning, location tracking, and tampering of data. The attacker motivation for performing these activities is potentially high. The attacker can obtain financial gains (e.g., offering services for corporate espionage purposes).

Mechanisms at the RFID level of the EPC architecture must be applied to mitigate security risks. The implementation of new security features in EPC tags faces several challenges, the main one being cost. The total cost of an EPC tag was estimated in [Sar01] to be less than 10 US dollar cents per unit. The goal is to maintain a low cost. Other challenges include compatibility regulations, power consumption, and performance requirements [SBCM07]. Detecting and responding to security threats are becoming major concerns of information security researchers. However, and before going further in these activities, an evaluation of the threats in terms of importance must be done. In the sequel, we present such an evaluation. Our analysis of the threats is based on a methodology proposed by the European Telecommunications Standards Institute (ETSI) [ETS03]. According to this methodology, we rank the threats in order of relevance. This assessment is intended to prioritize threats for future research on appropriate countermeasure mechanisms.

Remainder Outline: Section 2.2 outlines the methodology used for our analysis of threats. Section 2.3 presents the identified threats and their risk assessments. Section 2.4 surveys traditional security defenses for RFID solutions. Section 2.5 concludes the chapter.

2.2 Threat Analysis Methodology

We define a threat as the objective of an attacker to violate security properties of a target system, such as authenticity, integrity, and availability. We define the attacker as an agent that is exploiting a vulnerability of the targeted system to carry out the threat. The exploitation of the vulnerability is defined as the attack. The security officer of the target system must put in place countermeasures to reduce the risk of the undesirable activities associated with all the threats. Given the difficulty of implementing countermeasures for every possible threat against a system, it is crucial for security officers to identify threats with potentially high impact and ensure the presence of countermeasures. This is indeed the objective of the threat analysis.

The methodology we use in this chapter is based on a framework proposed by the ETSI [ETS03]. ETSI identifies three levels of threats: critical, major, and minor. Each level depends on estimated values for the likelihood of occurrence of the threat and its potential impact on a given system. The likelihood of a threat (cf. Figure 2.1(a)) is determined by the motivation for an attacker to carry out an attack associated to the threat versus the technical difficulties that must be resolved by the attacker to effectively implement the attack. The three levels of likelihood are: (1) *likely*, if the targeted system is almost assured of being victimized, given a high attacker motivation (e.g., financial gains as a result of selling private information or disrupting network services) and lack of technical difficulties (e.g., a precedent for the attack already exists); (2) *possible*, if the motivation for the attacker is moderate (e.g., limited financial gains) and technical difficulties are potentially solvable (e.g., the required theoretical and practical knowledge for implementing the attack is available); and (3) *unlikely*, in case there is little motivation for perpetrating the attack (e.g., few or no financial gains resulting from the attack) or if significant technical difficulties and obstacles must be overcome (e.g., theoretical or practical elements for perpetrating the attack are still missing).

The impact of a threat evaluates the potential consequences on the system when the threat is successfully carried out. The following three categories are identified: (1) *low*, if the consequences of the attack can be quickly repaired without suffering from financial losses;

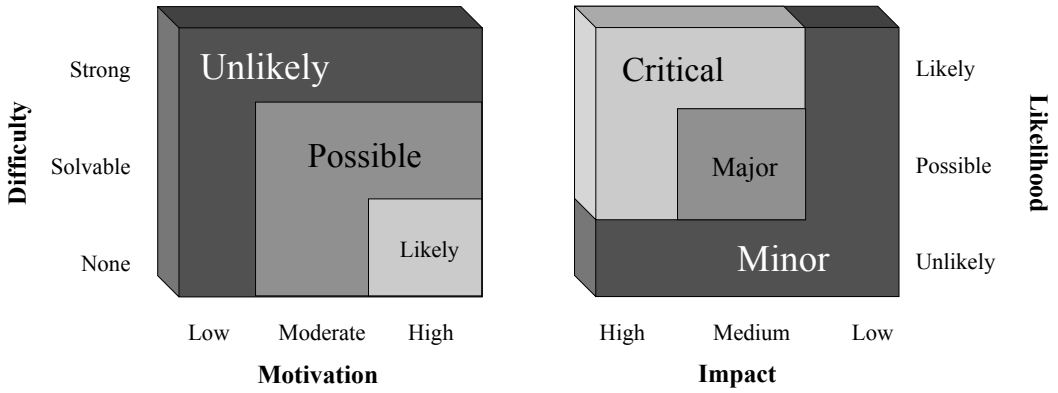


Figure 2.1: Likelihood and risk functions: (a) likelihood of a threat, (b) risk evaluation function.

(2) *medium*, if the consequences are limited in time but might result in few financial losses; and (3) *high*, if the attack results in substantial financial loss and/or law violations. The risk of a threat is ranked in [ETS03] as *minor*, if it is unlikely to happen and it has low or medium potential impact, or if it is possible but with low potential impact. A threat is ranked as *major* if it is likely but has low potential impact, if it is possible and has medium potential impact, or if it is unlikely but has high potential impact. A threat is ranked as *critical* if it is likely and has high or medium potential impact, or if it is possible and has high potential impact. Through our experience with the ETSI methodology, we have observed that several threats are overclassified as major, when they would better be ranked as minor. We have slightly adapted the risk function in order to focus on truly critical or major threats. Figure 2.1(b) presents the adapted risk function. A threat is ranked as major when its likelihood is possible and its potential impact is medium. A threat is ranked as minor when it is unlikely to happen or when its potential impact is low. Minor risk threats typically require no countermeasures. Major and critical threats need to be handled with appropriate countermeasures. Moreover, critical threats should be addressed with the highest priority.

2.3 Evaluation of Threats

The communication channel between the components of the RFID system of an EPC network, that is, tags and readers, is a potentially insecure wireless channel. It is fair to assume that most of the threats on EPC configurations are going to target this level. We analyze threats targeting basic security features such as authenticity, integrity, and availability during the exchange of data between an RFID tag and a reader. We assume that attackers may only act from outside when trying to exploit the wireless channel between tags and readers, for example, the lack of authentication between these elements. We therefore assume that attackers do not have physical access neither to the components of the system nor to the organization itself. The reason we ignore direct physical access is because we assume the presence of other security mechanisms in the organization (e.g., physical access control and surveillance of workers). Attackers, however, may have access to information about the system and its components or services. We summarize in Table 2.1 the results of our evaluation.

2.3.1 Authenticity Threats

The EPC Gen2 standard is designed to balance cost and functionality [Sar01]. However, security features on board Gen2 tags are minimal. They protect message integrity via 16-bit Cyclic Redundancy Codes (CRC) and generate 16-bit pseudorandom strings. Their memory, very limited, is separated into four independent blocks: reserved memory, EPC data, Tag Identification (TID), and user memory. The absence of strong authentication on the tags opens the door to malicious readers that can impersonate legal readers and perform eavesdropping attacks. Figure 2.2 shows a simplified description of the steps of the Gen2 protocol for product inventory. In Step 1, the reader queries the tag and selects one of the following options: select, inventory, or access [EPC07]. Figure 2.2 represents the execution of an inventory query. It assumes that a select operation has been completed in order to single out a specific tag from the population of tags. When the tag receives the inventory query, it returns a 16-bit random string denoted as RN16 in Step 2. This random

Threats	*Motivation*	*Difficulty*	*Likelihood*	*Impact*	*Risk*
Eavesdropping, rogue scanning	High	Solvable	Possible	High	Critical
Cloning of tags, location tracking	Moderate	Solvable	Possible	Medium	Major
Tampering of data	Moderate	Solvable	Possible	High	Critical
Destruction of data, denial of service	Moderate	Solvable	Possible	Medium	Major
Malware	Moderate	Strong	Unlikely	Medium	Minor

Table 2.1: Evaluation of Threats.

string is temporarily stored in the tag memory. The reader replies to the tag in Step 3 with a copy of the random string, as an acknowledgment. If the echoed string matches the copy of the RN16 stored in the tag memory, the tag enters the acknowledged state and returns the EPC identifier.

Observe that any compatible Gen2 reader can access the EPC. The traffic between tags and readers flows through non-authenticated wireless channels. Illegitimate collection of traffic might be slightly protected by reducing the transmission power or by sheltering the area. It is, although, theoretically possible to conduct eavesdropping attacks. We define forward eavesdropping as the passive collection of queries and commands sent from readers to tags; and backward eavesdropping as the passive collection of responses sent from tags to readers. Although the range for backward eavesdropping could be only of a few meters [EPC07], and probably irrelevant for a real eavesdropping attack, the distance at which an attacker can eavesdrop the signal of an EPC reader can be much longer. In ideal conditions, for example, readers configured to transmit at maximum output power, the signal could be received from tens of kilometers away. Analysis attacks inferring sensitive information from forward eavesdropping, for example, analysis of the pseudorandom sequences generated by the tags (denoted as RN16 in Figure 2.2), are hence possible. Replay attacks enabled by this inferred data are also possible. The absence of a strong authentication process also enables scanning attacks. Although, the distance at which an attacker can perform scanning is considerably shorter than the distance for forward eavesdropping.

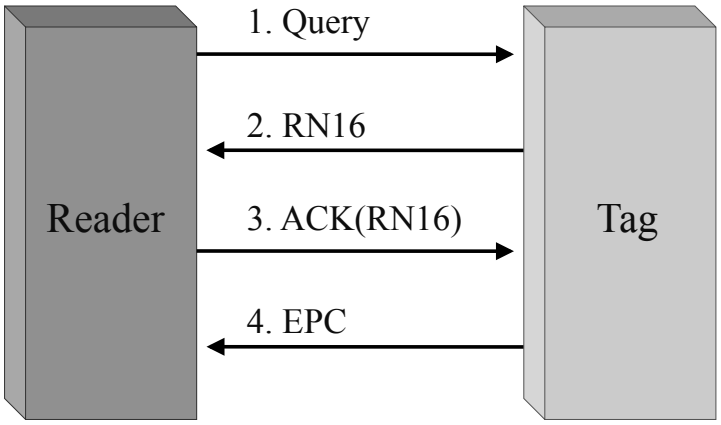


Figure 2.2: Inventory protocol of a Gen2 tag.

The use of special hardware (e.g., highly sensitive receivers and high gain antennas) could enable rogue scanning attacks.

We can conclude that outsiders equipped with Gen2 compatible readers and special hardware can theoretically eavesdrop the communication between readers and tags; or scan objects in motion if they successfully manage to place their readers at appropriate distances. According to [EPC07], the information stored on an EPC tag is limited to an identification number. No additional data beyond the number itself is conveyed in the EPC. Additional information associated with the code must be retrieved from an EPCIS. However, an attacker accessing these data may determine types and quantities of items in a supply chain and sell the information to competitors or thieves. An attacker can obtain information from the EPC, that is, the manufacturer and product number. This information may be used for corporate espionage purposes by competitors, or for other attacks against other services of the EPC infrastructure. Clearly, the motivation of an attacker to carry out this threat must be rated as high, since attackers can sell their services to competitors, thieves, or any other individual looking for the objects tagged in the organization. The difficulties for performing both eavesdropping and rogue scanning, as shown by the example depicted by Figure 2.2, are solvable. This level of motivation and degree of difficulty lead to a likelihood that

is possible. Regarding the potential impact of these threats (e.g., disclosure of information considered by the organization as confidential or trade secrets), it is high, since it may have serious consequences for an organization if an attacker offers the malicious service to competitors or to thieves. These threats are assessed as critical and need to be handled by appropriate countermeasures.

Using the codes eavesdropped or scanned by unauthorized readers, an attacker may successfully clone the tags by conducting, for example, skimming attacks. Indeed, an attacker can simply dump data and responses from a given tag, and program it into a different device. The objective of the attacker for performing the cloning of tags is the possibility for counterfeiting. The attacker may create fake EPC tags that contain data and responses of real tags and sell these counterfeit tags for profit. The forgery of legal tags can be performed without physical access to the organization. We rank the motivation of attackers to carry out the attacks associated with this threat as moderate since they can obtain some financial gain by offering this service to third parties. Current EPC specifications do not include any mechanism for Gen2 compatible readers to verify if they communicate with genuine or fraudulent tags. We thus rate the difficulties associated to this threat as solvable. This level of motivation and degree of difficulty lead to a likelihood that is possible. Regarding the potential impact of this threat, it is medium and thus the threat is assessed as major.

The lack of a strong authentication process in Gen2 tags also has consequences to the privacy of tagged object bearers. Indeed, interrogations of Gen2 tags give attackers unique opportunities for the collection of personal information (and without the consent of the bearer). This can have serious consequences, such as location tracking or surveillance of the object bearers. An attacker can distinguish any given tag by just taking into account the EPC number. Following a reasoning similar to the one used for the cloning threat leads to ranking the risk of the location tracking threat as major. This threat, as well as the cloning threat, must be handled by appropriate countermeasures.

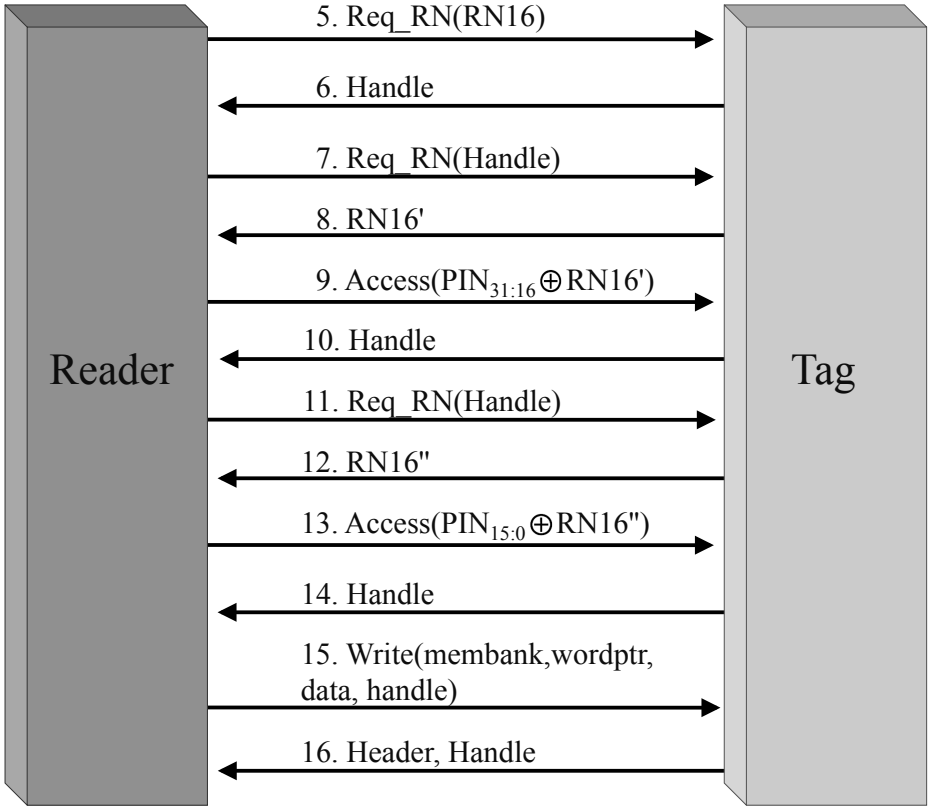


Figure 2.3: Writing protocol of a Gen2 tag.

2.3.2 Integrity and Availability Threats

Gen2 tags are required to be writable [EPC07]. They must also implement an access control routine, based on the use of 32-bit passwords, to protect the tags from unauthorized activation of the writing process. Other operations, also protected by 32-bit passwords, can be used in order to permanently lock or disable this operation. Although the writable feature of Gen2 tags is very interesting, it is also one of the least exploited features in current EPC scenarios (due probably to the lack of a strong authentication process, as reported in the previous section). Writable tags are hence locked in most of today's EPC applications. This option will, however, be extremely important in future EPC applications, especially

on those self-organizing-based scenarios, where the addition of complementary information into the memory of the tags will require the unlocking of the writing process (e.g., to store routing parameters, locations, or time stamps). It is therefore important to analyze the risk of a tampering attack to the data stored by Gen2 tags, if they can be accessed in write mode from a wireless channel that does not guarantee strong authentication.

Figure 2.3 presents a simplified description of protocol steps for requesting and accessing the writing process that modifies the memory of a Gen2 tag.We assume that a select operation has been completed, in order to single out a specific tag from the population of tags. It is also assumed that an inventory query has been completed and that the reader has a valid RN16 identifier (cf. Figure 2.2, Steps 2 and 3) to communicate and request further operations from the tag. Using this random sequence (cf. Figure 2.2, Step 5), the reader requests a new descriptor (denoted as Handle in the following steps). This descriptor is a new random sequence of 16 bits that is used by the reader and tag. Indeed, any command requested by the reader must include this random sequence as a parameter in the command. All the acknowledgments sent by the tag to the reader must also include this random sequence. Once the reader obtains the Handle descriptor in Step 6, it acknowledges by sending it back to the tag as a parameter of its query (cf. Step 7). To request the execution of the writing process, the reader needs first to be granted access by supplying the 32-bit password that protects the writing routine. This password is actually composed of two 16-bit sequences, denoted in Figure 2.3 as $\text{PIN}_{31:16}$ and $\text{PIN}_{15:0}$.

To protect the communication of the password, the reader obtains in Steps 8 and 12, two random sequences of 16 bits, denoted in as RN16' and RN16". These two random sequences RN16' and RN16" are used by the reader to blind the communication of the password toward the tag. In Step 9, the reader blinds the first 16 bits of the password by applying an XOR operation (denoted by the symbol $\oplus$ in Figure 2.3) with the sequence RN16'. It sends the result to the tag, which acknowledges the reception in Step 10. Similarly, the reader blinds the remaining 16 bits of the password by applying an XOR operation with the sequence RN16", and sends the result to the tag in Step 13. The tag acknowledges the reception in Step 14 by sending a new Handle to the reader. By using the latter, the reader requests the writing operation in Step 15, which is executed and acknowledged by the tag

in Step 16. Notice that an attacker can find the 32-bit password that protects the writing routine. It suffices to intercept sequences RN16' and RN16", in Steps 8 and 12, and to apply the XOR operation to the contents of Steps 9 and 13. Other techniques to retrieve similar passwords have also been reported in the literature. For example, in [Ore07] the authors present a mechanism to retrieve passwords by simply analyzing the radio signals sent from readers to tags. Although the proof-of-concept implementation of this technique is only available for Gen1 tags [EPC07], the authors state that Gen2 tags are equally vulnerable. The technical difficulties for setting up attacks to retrieve the password are therefore ranked as solvable. The likelihood of the tampering threat is classified as possible. Regarding the impact, it is ranked, because of some extreme scenarios, as high. For example, in the context of a pharmaceutical supply chain, corrupting data in the memory of EPC tags can be very dangerous: the supply of medicines with wrong information, or delivered to the wrong patients, can lead to situations where a sick person could take the wrong drugs. In these circumstances, the combination of likelihood and impact of the tampering threat lead to critical risk. The threat needs therefore to be addressed by appropriate countermeasures.

The aforementioned attacks enabled by retrieving the passwords, that protect both writing and self-destruction routines of Gen2 tags [EPC07], can be used as models to analyze the risk of threats like destruction of data or denial of service [HTKC06]. Tag information can also be destroyed by devices that send strong electromagnetic pulses. Devices, such as the RFID-zapper [MM05], have been presented in the literature. We can also include here denial of service attacks consisting of jamming channels or flooding channels between tags and readers by sending a large number of requests and responses. For performing a jamming attack, the attacker uses powerful transmitters to generate noise in the range of frequencies used by readers and tags. In any case, the technical difficulties are ranked as solvable. The motivation of attackers to carry out these threats is rated as moderate, since they can obtain some financial gain by offering their malicious services. The likelihood of these two threats is hence classified as possible. However, since the impact of these threats represents to the victim temporal disruption of its operations rather than great financial losses, we rate the impact as medium, and so the risk of these two threats as major.

The final threat analyzed in this section, related to attacks to the integrity and availability of the back-end servers connected to RFID readers, was initially reported in [RCT06]. Rieback et al. uncover the possibility of using malware to attack back-end databases. Their approach classifies such malware into three categories: (1) exploits, (2) worms, and (3) viruses. The exploits are attacks carried out within the information stored into RFID tags. They target the security of middleware services connecting readers to back-end databases. Worms and viruses are attacks that spread themselves over new RFID tags by using network connections (in the case of worms) or connectionless self-replication strategies (in the case of viruses). The malware reported in [RCT06] exploits the trust relationship between backend databases and the information sent by readers-obtained in turn from malicious tags. Rieback et al. consider that even if there is a very tiny window for storing information into an RFID tag, traditional attacks against information systems (e.g., buffer overflows and SQL injection attacks) might be condensed into a small string of bits harmful enough to break the security of a system. The authors present a proof-of-concept that uses tags carrying an SQL injection attack that compromises the security of the back-end layer of an ad hoc RFID setup. The work presented in [RCT06] is interesting and relevant. However, we think that the likelihood for those threats must be rated as unlikely, since no real-world vulnerabilities on the filtering and collection of middleware services specified by EPCglobal can be exploited at the moment. We conclude that even if the impact is potentially serious, due to its unlikely degree of likelihood, the threat is assessed as minor.

2.4 Survey of RFID Defense Countermeasures

Research on defense countermeasures for RFID technologies can be divided in two main categories: (1) hardware-based security primitives for RFID tags, and (2) security protocols using the hardware-based primitives. We review in this section a non-exhaustive list of contributions in both categories.

2.4.1 Hardware-Based Primitives

According to research presented in [Sar01], the cost of passive EPC tags should not exceed 5 US dollar cents to successfully enable their deployment on worldwide scale. Of these 5 cents, only 1 or 2 should be used for the manufacturing of the integrated circuit (IC). Another challenge is that the available layout area for the implementation of the IC is in the order of 0.25 mm2 which, considering current complementary metal-oxide-semiconductor (CMOS) technology, corresponds to a theoretical number of logic gates from 2000 to 4000. Not all the barriers identified in [Sar01] have been removed. Today, the EPC technology is more expensive than what it was originally anticipated (around 10 US dollar cents per unit in large quantities). The inclusion of additional features, especially for security purposes, may increase the total end-cost of tags up to 15 cents per unit or more. Although Moore's Law says that the cost of ICs will continue to decrease, cost of analogue devices (i.e., RF front-end of tags) is relatively stable and will remain a constraint [CR08]. The inclusion of new elements must therefore be clearly justified.

Since EPC tags are powered from the weak energy captured from a reader's electromagnetic waves, their current consumption also needs to be taken into account. This consumption varies according to the operation that is being performed (e.g., responding to a query or writing data into the memory) and other parameters such as the transmission rate, response delay, and memory technology. Most of the operations performed within the modern EPC tags consume about 5-10 μA (microamperes). Some special operations, such as write accesses, may consume more. The current consumption of new security primitives must be within this range to allow low-cost tag production. They must also work at the data rate of EPC applications. For example, some supply chain applications demand an average reading speed of about 200 tags/s. This leads to a data transmission rate from tag to reader, of about 640 kbps; and from reader to tag of about 120 kbps. Delays associated to new security mechanisms (e.g., time to perform encryption or random number generation) may also affect the global performance. Delays must hence be taken into account and minimized. We refer the reader to [CR08] for a more detailed description of aspects that must be considered during the design of new primitives.

Several security proposals aim at including cryptographic primitives on low-cost EPC tags. However, not all the proposals meet the aforementioned constraints or guarantee secure designs. Existing implementations of one-way hash functions, such as MD4, MD5, and SHA-128/SHA-256, exceed cost constraints due to the required number of gates. According to [FR06], this amounts of having from 7000 to over 10,000 logic gates. The use of cellular automata (CA) theory for the implementation of one-way functions [Wol86] and encryption engines [SSC+02] has been investigated for the implementation of cryptographic primitives on low-cost RFID tags. However, it has been proved that these implementations are insecure [Bar90, BMP97].

The use of linear feedback shift registers and nonlinear feedback shift registers as underlying mechanisms for the implementation of low-cost one-way hash functions and pseudorandom number generators (PRNG), without appropriate measures that might make increase the cost, also lead to insecure implementations [MT72, MOV01]. Light-weight hardware implementations of standard block ciphers to implement one-way hash functions have been discussed. The use of elliptic curve cryptosystems (ECC) [MOV01] for the implementation of primitives for RFID tags has been discussed in [Wol05]. Its use of small key sizes is seen as very promising for providing an adequate level of computational security at a relatively low cost [CR08]. An ECC implementation for low-cost RFID tags can be found in [BGK+06]. In [FDW04], Feldhofer et al. present a 128-bit implementation of the advanced encryption standard (AES) [DR02] on an IC of about 3,500 gates with a current consumption of less than 9 μA at a frequency of 100 kHz. The encryption of each block of 128 bits requires about 1000 clock cycles. Although, it considerably simplifies previous implementations of the AES cryptosystem, for example, proposals presented in [MAD03, VSK03] that require between 10,000 and over 100,000 gates, respectively, the design is still considered too complex for basic EPC setups [Jue06].

More suitable encryption engine implementations can be found in [Isr06, MSQ07]. The first reference presents the implementation of the tiny encryption algorithm (TEA) [WN95]. It is implemented on an IC of about 3,000 gates with a current consumption of about 7 μA.

It fits the timing requirements of basic EPC setups where hundreds of tags must simultaneously be accessed by the same reader. For meeting the constraints, the implementation relies on very simple operators such as XOR, ADD, and SHIFT. The authors of TEA [WN95] claim that, despite its simplicity and ease of implementation, the complexity of the algorithm is equivalent to data encryption standard (DES) [MOV01]. Variants of the basic algorithm, such as eXtended TEA (XTEA), are however necessary for implementing one-way hash functions. Mace et al. discuss in [MSQ07] some of the vulnerabilities of TEA, such as linear and differential cryptanalysis attacks, and present scalable encryption algorithm (SEA). Given the relatively recent invention of these algorithms, their strength is not clear [CR08].

There are other hardware-based security enhancements for RFID technologies not relying on the implementation of cryptographic primitives. Many signal- and power-based defenses, such as shielding of tags, use of noise and third-party blocker devices have been surveyed in [Jue06]. The use of distance measurements to detect rogue readers has been discussed. In [FRJ05], for example, Fishkin et al. propose the inclusion of low-cost circuitry on tags to use the signal-to-noise ratio of readers as a metric for trust. In [Han07], a similar assumption is used in order to claim that a reader can be authorized to read a tag contents according to its physical distance. The use of trust [SDFMBD07] and trusted computing [MSW05] with similar purposes has also been discussed. For example, Molnar et al. describe in [MSW05] a mechanism consisting of trusted platform modules (TPMs) to enforce privacy policies within the RFID tags. A trusted entity called trusted center (TC) decides whether readers are allowed or not to access tags. Finally, the use of radio fingerprinting to detect characteristic properties of transmitted signals and design authentication procedures has been investigated. The authors in [CR08] consider, however, that this technique is difficult to develop on RFID applications and that the benefits of using it, with respect to performance, cost, and required implementation surface on tags, are unclear. Avoine and Oechslin also debate in [AO05a] the prevention of the traceability via radio fingerprinting. They conclude that obtaining radio fingerprint of tags is expensive and difficult. The myriad of tags in circulation in future RFID scenarios makes impracticable the individual distinction of them.

Physically unclonable functions (PUFs) and physical obfuscated keys (POKs) are promising for the implementation of new security primitives in low-cost EPC tags. They can be used to handle the authentication threat, as well as the cloning and location tracking threats. Half way between cryptography-based enhancements and physical protection defenses, the ideas behind PUFs and POKs originated in [Pap01] with the conception of optical mechanisms for the construction of physical one-way functions (POWFs). Their use to securely store unique secret keys, in the form of fabrication variations, was proposed as a silicon prototype in [Gas03, GCvDD02]. These ideas were later improved in [LLG+05]. A coating PUF proposed in [ST05] is implemented with less than 1000 gates. These designs exploit the random variations in delays of wires and logic gates of an IC. For example, the silicon PUF presented in [GCvDD02] receives input data, as a challenge, and launches a race condition within the IC: two signals propagate along different paths and are compared to determine which one comes first. To decide which signal comes first, a controller, implemented as a latch, produces a binary value.

On a similar vein, Holcomb et al. [HBF07] propose using the SRAM based on CMOS circuitry to generate physical fingerprints. The key idea is the use of SRAM start-up values as origin of randomness. The use of 256 bytes of Static Random Access Memory (SRAM) can yield 100 bits of true randomness each time that the memory is powered up. While sound in theory, this technique has as important drawback the limitation of memory space of current low-cost tags. The implementation of PUF-based circuits seems to have clear advantages at a cost of less than 1,000 logic gates [ST05]. This technology provides a cost effective and reliable solution that meets the constraints and requirements. Drawbacks, such as the effects of environmental conditions and of power supply voltage [REC04b], must be taken into account. The difficulty of successfully modeling the circuits and their reliability have also raised some concerns. Bolotnyy and Robins [BR07] address some of these issues. Some attacks on PUF- and POK-based protocols are outlined in [LLG+05]. The execution and reinterpretation of existing protocols via new PUF and POK designs (using, essentially, challenge-response protocols) are outlined next.

2.4.2 Security Protocols

We review algorithmic solutions and software protocols for handling the threats uncovered in Section 2.3. The solutions rely on the implementation and use of hardware-based primitives discussed in Section 2.4.1.

Message Authentication Code (MAC)-based security protocols for wireless applications is a typical solution discussed in the literature (e.g., [BR07, TUI+01, WKHW02]). In [TUI+01], Takaragi et al. present a very simple MAC-based approach. It uses a static unrewritable 128-bit identifier stored, at manufacturing time, in every tag. This static identifier is not modifiable once the shipment is made. To build up this identifier, the manufacturer uses a unique secret key for each tag and a keyed hash function that accepts as input the secret key and a specific message. All this information (i.e., secret key, hash function, and specific message) is communicated by the manufacturer to the client. By sharing this information among readers and tags, integrity and authenticity of exchanged messages is verified. It therefore reduces the risk of threats to authenticity and integrity by increasing the technical difficulties of performing attacks. However, due to the use of static identifiers embedded in the tags at manufacturing time, the location tracking issue is not solved. Moreover, brute force attacks can break the secrets shared between readers and tags.

The use of public key cryptography and digital signatures is discussed in [JP03]. The authors address the protection of banknotes embedding the RFID tags. Their approach includes the possibility of deploying cryptographic protocols in RFID applications, but avoids the need to embed cryptographic primitives within the tags. The scheme consists of a public-key cryptosystem used by a central bank aiming to avoid banknote forgery and a law enforcement agency that aims at tracking banknotes. Both authorities, that is, central bank and law enforcement agency, hold an independent pair of public and private keys associated to each banknote. The central bank authority assigns a unique serial number to each banknote. The central bank authority, using its private key, signs the unique serial number. The unique serial number of the banknote and its corresponding digital signature are printed on the banknote as optical data. In addition, the law enforcement agency encrypts with its public key the digital signature, unique serial number, and a random number.

The resulting ciphertext is stored into a memory cell of the RFID tag. This memory cell is keyed-protected. The tag only grants write access to this memory cell if it receives an access key derived from the optical data. The random number used to create the ciphertext is also stored into a separated memory cell of the tag. This second memory cell is also keyed-protected. The tag only grants read or write access to this memory cell if it receives an access key derived from the optical data.

Now, a merchant that receives a banknote must verify first the digital signature, printed in the banknote as optical data, using the public key of the central bank. Second, the merchant must also verify the validity of the ciphertext stored in the banknote's tag.To do so, the merchant encrypts the digital signature, serial number, and random number stored in the tag's memory, using the public key of the law enforcement agency and the optical data. If one of these two verification processes fails, the authorities must be warned. To avoid using the same ciphertext on every interaction, Juels and Pappu propose the use of a reencryption process that can be performed by the merchant without the necessity of accessing the private keys of the law enforcement authority. Indeed, based on the algebraic properties of the ElGamal cryptosystem [MOV01], the initial ciphertext can be transformed into a new unlinkable ciphertext only using the public key of the law enforcement authority [Jue06]. This reencryption process is performed outside the tags. Integrity issues of this approach are discussed and fixed in [ZK05]. However, the whole process and requirements for implementing the approach in [JP03, ZK05] are too complex and expensive for use in EPC supply chain applications.

Mutual authentication protocols among tags and readers are discussed in [KHK+03, Jue04]. The work presented by Kinosita et al. in [KHK+03] consists of an anonymous ID scheme, in which a tag contains only a pseudonym that is periodically rewritten. Pseudonyms are used instead of real identifiers (e.g., instead of the EPC codes). Similarly, the approach of Juels entitled minimalist cryptography for low-cost RFID tags [Jue04] suggests a very lightweight protocol for mutual authentication between tags and readers based on one-time authenticators. Both solutions rely on the use of pseudonyms and keys stored within tags and back-end servers. Each tag contains a small collection of pseudonyms, according to the available memory of the tag. A throttling process is used to rotate the pseudonyms.

Each time the tag is interrogated by a reader, a different pseudonym is used in the response. Authorized readers have access to the complete list of pseudonyms set for each tag and can correlate the responses they receive. Without the knowledge of this list, unauthorized readers are unable to infer any information about the several occurrences of the same tag. The process also forces tags to slow down their data transmissions when queried too frequently, as a defense to potential bruteforce attacks. The memory space on current low-cost tags is the main limitation of this approach. Although enhancements can be used to update the list of pseudonyms, communication costs, and integrity threats will remain as main drawbacks.

The use of hash-lock schemes for addressing authentication issues is another possibility. A design can be found in [WSRE03a]. Weis et al. propose a way to lock tags without storing access keys in them. Only hashes of keys must be known by the tags. Keys must be also stored on back-end servers and be accessible by authorized readers. Most authentication threats are therefore mitigated by locking tags. Cloning and tracking threats are handled by avoiding the use of real identifiers once tags are locked. In [HM04], Henrici and Müller extend the hash-lock scheme and address some weaknesses in [WSRE03a] to increase traceability and location resistance. A similar hash-based protocol is presented in [AO05b] in order to deal with those limitations by using time stamps. Other similar hash-based protocols for handling authentication threats can be found in [LHLL05, CLL05, LAK06]. All these protocols rely on synchronized secrets residing in the tags and back-end servers. They require a one-way hash function implemented within the tags. The requirement of reliable hash primitives implemented at the tag level is the main drawback. Workload on back-end servers is also considerably high and can make difficult the deployment in real-world EPC supply chain applications. The Yet Another Trivial RFID Authentication Protocol (YATRAP) protocol presented in [Tsu06] reduces the cost of computation by combining precomputed hash-tables for tag verification processes, use of time stamps, and generation of pseudorandom numbers. The protocol is, however, vulnerable to availability attacks when temporal desynchronizations between tags and readers occur. Some limitations are addressed in [CvLB06]. Chatmon et al. define new protocols for anonymous authentication. These improvements notably increase the degree of workload on servers and are highly complex for use in supply chain applications.

Challenge-response protocols for low-cost EPC tags using physical unclonable functions (PUFs) and POKs have recently gained importance. An approach presented in [REC04a], based on PUFs proposed in [Gas03, GCvDD02], consists of a challenge-response scheme that probabilistically ensures unique identification of RFID tags. A back-end system must learn challenge-response pairs for each PUF/tag. It then uses these challenges (hundreds of them) at a time, to identify and authenticate tags. Unique identification of tags is only ensured probabilistically. The exposition of tag identifiers to eavesdroppers and lack of randomness in tag responses, make the approach vulnerable to the location tracking threat. Moreover, the great number of challenges that are necessary in the identification process increases the tag response delay and power consumption. Hence, this approach might not meet the constraints and requirements of the EPC technology.

An alternative approach is presented in [TB06]. Tuyls and Batina discuss an off-line PUF-based mechanism for verifying the authenticity of tags through the PUF technology presented in [ST05]. Similar to the results presented in [Jue04, JW05], where readers and tags define ad hoc secrets, the PUF-based approach uses the internal physical structure of tags to generate unique keys. A key extraction algorithm from noisy binary data is presented in [TB06]. The usage of PUF-based keys simplifies the process of verifying tag authenticity. The combination of unique keys generated on-board together with the use of signatures avoid leaking of a single identifier and increases the technical difficulties for an attacker to carry out the location tracking threat. The main drawback is the need of large storage space and reliable searching processes on back-end servers to link readers with PUF/tag identifiers. The use of public key and digital signatures, based on Elliptic Curve Cryptography (ECC), is another important constraint.

Following the trend of combining PUFs together with traditional cryptographic primitives and encryption engines, a modification of the tree-based hash protocols proposed in [MW04] is presented in [BCI07]. Using the notion of POKs introduced in [Gas03] (i.e., application of a fixed hard-wired challenge to the PUF to obtain a unique secret), the authors guarantee the existence of internal keys in basic tree-based hash protocols, now physically obfuscated. They cannot be cloned by unauthorized parties. The use of an AES engine, such as the one presented in [DR02], is proposed. On the other hand, Bolotnyy and

Robins present in [BR07] a complete set of adapted MAC protocols, based on PUFs, trying to simplify the challenge-response communication scheme of previous proposals and to eliminate requirement of traditional cryptographic primitives. Each tag generates multiple identifiers based on embedded PUFs. Their approach only addresses static identification. It is vulnerable to the location tracking threat identified in Section 2.3. It does not solve the requirement of huge lists of challenge-response pairs for each PUF/tag that must be stored on back-end servers connected to the readers. Indeed, each given pair is of single use to prevent replay attacks.

Towards Secret-sharing Schemes

Secret-sharing schemes have been presented as efficient algorithmic solutions to balance security and low-cost on-tag cryptographic processes. Juels et al. present in [JPP08] a defense countermeasure to authenticity threats in EPC supply chain applications. Two different models are discussed: dispersion of secrets across space and dispersion of secrets across time. Both models are based on a secret-sharing strategy, where a secret used to encrypt EPCs is split in multiple shares and distributed among multiple parties. In order to obtain the EPC of a tag, a party must collect a minimum number of shares distributed among all the other parties. Authentication is therefore achieved though the dispersion of secrets. The dispersion helps to improve the authentication process between readers and tags, as tags move through a supply chain. Assuming that a given number of shares is necessary for readers to obtain the EPCs assigned to a pallet, for example, a situation where the number of shares obtained by readers is not sufficient to reach the threshold protects the tags from unauthorized scanning (i.e., unauthorized readers that cannot obtain the sufficient number of shares cannot obtain the EPCs either). The approach can be implemented on EPC Gen2 tags without requiring any change to the current tag specification. A limitation is the amount of tag memory space required for storing the shares. However, the shrinking of shares can allow the application of the scheme to current EPC tags. A more important problem is that the location tracking threat is not addressed. Indeed, the shares used in the approach are static. This problem must be solved before deployment of the scheme.

2.5 Concluding Summary and Remarks

At the beginning of this chapter we presented an analysis of threats to the RFID system of the EPC architecture. We identified different groups of threats that we consider relevant for further research. We ranked the eavesdropping, rogue scanning, and tampering threats as critical; and cloning, tracking, and denial of service threats as major. We concluded that they must be handled by appropriate countermeasures. We then surveyed in the sequel practical and theoretical security defenses that can be useful to reduce the risk of the identified threats. We looked at the different defenses from two different research perspectives. On the one hand, we surveyed research on hardware-based defenses that aim at providing additional security primitives on tags such as one-way hash functions, encryption engines, and physically unclonable functions (PUFs). On the other hand, we surveyed research on software protocols that make use of these new on-tag primitives for designing and implementing reliable algorithms for dealing with security and privacy issues.

We have also seen that the implementation of well-known cryptographic primitives is possible and allows the design of software protocols to reduce the risk of threats ranked as critical or major. The cost and requirements of these proposals are the main difficulties. Indeed, they are too expensive for their deployment in supply chain scenarios based on the EPC technology. We have also surveyed the combination of cryptographic primitives together with the use of PUFs for the design of cost-effective solutions. These solutions present drawbacks, such as the sensitivity of PUFs to physical noise and the difficulty to model and analyze them. They are, however, promising solutions that successfully meet the implementation constraints and requirements for handling the set of threats reported in our work. For the second group, we conclude that the avoidance of on-tag cryptographic processes on current algorithmic solutions seems to lead the future directions of research in RFID security. In this sense, the use of secret-sharing schemes present clear advantages for the management of keys in the design of authentication protocols and to deal with privacy issues. The main drawback is the use of static shares, limiting the use of this approach for addressing the location tracking threat.

Chapter 3

Weak EPC Pseudorandom Generators

In Chapter 2 we have seen that the EPC Gen2 is an international standard that proposes the use of Radio Frequency Identification (RFID) in ubiquitous environments. It has been designed to balance cost and functionality. As a consequence, security on-board of EPC tags is often minimal. In fact, the security of EPC Gen2 tags is mainly based on the use of on-board Pseudorandom Number Generators (PRNGs), used to obfuscate the communication exchanges requested by RFID readers. It is also used to acknowledge the proper execution of password-protected operations. In this chapter we deepen our analysis on PRNGs as main security tool for low-cost RFID. Cryptographic suitable PRNG designs must satisfy unpredictability characteristics. Otherwise, an external adversary who eavesdrops the communication between readers and tags can compute the PRNG internal state when enough outputs of the generator are observed. If this happens, it might allow the adversary to bypass the security of the password-protected commands defined in the EPC Gen2 standard (e.g., the access and the kill commands). We report a weak PRNG designed specifically for EPC Gen2 tags in [CHTW08, CCY+11]. We show that it is feasible to eavesdrop a small amount of pseudorandom values by using standard EPC commands and using them to determine the PRNG configuration that allows to predict the complete output sequence. We conclude by drawing some ideas to solve the linearity problem found in the analysis section. Parts of this chapter have been previously published in [188, 191, 192].

3.1 Pseudorandom Number Generators for EPC Gen2

The design of PRNGs for EPC Gen2 tags is not an easy task due to the computational and memory restrictions that these tiny devices imply. Capabilities of this type of tags are so small that security features for the EPC Gen2 standard are expected to be implemented with a small amount of equivalent logic gates (GE), defined in the literature between 2,000 and 5,000 [RC08]. This is a extremely small value if we consider that a standard hash function (the most simple cryptographic transformation), like SHA1, needs at least 8,120 GE (equivalent logic gates) to be implemented [FR06].

Existing Proposals

Existing commercial Gen2 tags do implement a PRNG, as it is an EPC standard mandatory, but companies are often reluctant to present the design of their PRNGs. Some statistical artifacts on the PRNGs of commercial Gen2 tags have been reported in [194]. Manufacturers simply refer to testbeds that show the accomplishment of some expected requirements, most of them for compatibility purposes. They fail to offer convincing information about the PRNGs designs [PLHCETR09]. This is mostly security through obscurity, which is always ineffective in security engineering, as it has been shown with the disclosure of the PRNG used in the MIFARE Classic chip [GKM$^+$08] that has shown a vulnerable PRNG.

Few PRNG proposals have been presented in the scientific literature specifically designed for EPC Gen2 tags. Peris-Lopez *et al.* present in [PLHCETR09] a deterministic algorithm that relies on the use of 32-bit keys and pre-established initial states. The set of functions mainly consists of bit rotations, bitwise operations, and modular algebra, building a 32-bit PRNG. The authors also propose an alternative 16-bit version of their PRNG for EPC Gen2 compatibility. To reduce the output length from 32 to 16 bits, Peris et al. divide the 32-bit output in two halves and XOR them to obtain the 16-bit output sequence. No evidences of further achievements other than hardware complexity and statistical behavior are provided. Moreover, the inherent peculiarity of their construction methodology obscures potential comparison with classical designs in the security literature. On a different vein,

Lee and Hong present in [LH07] an optimized variant of the shrinking generator [CKM94] for low-cost RFID tags. The shrinking generator is a well studied cryptographic design that combines two clocked linear feedback shift registers (LFSRs) [MOV01]. The output sequence of the first LFSR is used to discard some bits from the output sequence of the second LFSR. Some techniques presented in [MS95] can be used to attack the scheme. Moreover, there are no evidences of how the proposal in [LH07] controls the irregularities of the generator output rate. This is an important drawback inherent to any shrinking generator scheme, since it can hint at the state of the main LFSR, and so breaking the security of the generator. Che *et al.* describe in [CHTW08, CCY+11] another variant of the shrinking generator design, but based on a physical source of randomness that aims at handling the linearity of an underlying LFSR. In the sequel, we deepen our knowledge on LFSR-based PRNGS, and show that the linearity of the Che *et al.* proposal can be attacked with high probability.

Remainder Outline: Section 3.2 describes the suitability of using linear feedback shift registers (LFSRs) for the generation of pseudorandom sequences. Section 3.3 describes the proposal presented by Che *et al.* in [CHTW08, CCY+11], and provides a security analysis of the Che *et al.* scheme. We give the details of a statistical analysis performed over the output data based on the National Institute of Standards and Technology (NIST) statistical test for pseudorandomness. Based on the weakness detected by the NIST test, we also detail an attack that, given a small number of output bits, can determine the whole sequence. Section 3.4 provides the details of the attack implementation. The section provides information about the tools used to implement the attack and the empirical results obtained in the attack of the Che *et al.* scheme. Section 3.5 concludes the chapter and draws some ideas to handle the linearity problem found in [CHTW08, CCY+11].

3.2 LFSR-based Pseudorandom Number Generators

Linear feedback shift registers (LFSRs) are an important tool for designing PRNG for EPC Gen2 tags. They lead to extremely efficient and simple hardware implementations. For

instance, a 16-cell LFSR can be implemented with only 192 GE. An LFSR is a digital circuit that contains a shift register and a feedback function. The shift register is composed of n binary cells that share the same clock signal. Each time a bit is needed, the content of the register is shifted one cell, obtaining the most significant bit of the register in the previous state. The feedback function computes a new bit using some bits of the register, obtaining the less significant bit to be filled in the new state of the register. The feedback function of an LFSR is basically an exclusive OR logical operation of some cells content, named *taps*.

Although LFSRs can be implemented efficiently, their main drawback is that their sequences are high predictable [Her86, Che86]. For example, let $s_{k+1}, s_{k+2}, \cdots, s_{k+2n}$ be a sequence of $2n$ consecutive bits generated from an LFSR. Let $\mathcal{B} = (b_n, b_{n-1}, \cdots, b_1)$ be the feedback function of the LFSR. Then, the feedback function can be easily computed by solving the following equation system:

$$\begin{bmatrix} s_{k+1} & s_{k+2} & \cdots & s_{k+n} \\ s_{k+2} & s_{k+3} & \cdots & s_{k+n+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{k+n-1} & s_{k+n} & \cdots & s_{k+2n-2} \\ s_{k+n} & s_{k+n+1} & \cdots & s_{k+2n-1} \end{bmatrix} \begin{bmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_2 \\ b_1 \end{bmatrix} = \begin{bmatrix} s_{k+n+1} \\ s_{k+n+2} \\ \vdots \\ s_{k+2n-1} \\ s_{k+2n} \end{bmatrix} \tag{3.1}$$

By solving Equation (3.1) we obtain the feedback polynomial coefficients. Despite the $2^n - 1$ period length generated by a n LFSR, the full sequence can be determined only with $2n$ consecutive bits due to the linearity of the system. This linearity must be handled before using LFSRs to build robust PRNGs. Several basic constructions can be used to hide linearity, while maintaining suitable statistical properties and long output periods. One of these techniques are filters. Filters use a non-linear feedback function as an input to the register. The filter should not be too simple to be weak but neither too complex, otherwise it would become the bottleneck of the generator. However, recent attacks to the MIFARE PRNG [GKM$^+$08] have demonstrated the vulnerability of this kind of generators when the non-linear function is not taken carefully.

Another approach to handle the linearity of an LFSR is to use a non-linear combination of multiple LFSRs to generate a unique output. Generally the output of one LFSR is used to select or combine the output of one or more LFSRs, in the same or different clock times. Known examples of this approach are the Geffe, A5 or the Shrinking generator [Sch96]. The output generated from these constructions is statistically weak, being vulnerable to correlation or side-channel attacks [Jou09]. Moreover, the irregular output data rate from some of these constructions (e.g., the shrinking generator) is not suitable for PRNG used in security environments. Finally, generators with memory are another alternative. Additional memory can be used to add some non-linear information in between the clock steps of the LFSR. Besides the memory, also binary adders and carry registers should be used to complete this approach.

The different techniques of deterministic modifications of LFSRs explained so far are useful for keystream generators where *sender* and *receiver* can share a secret k as a key for the PRNG one-time pad communications. However, the specific communication model of EPC Gen2 systems uses another paradigm where *sender* and *receiver* cannot share any secret k. Instead of this, the low-power tag-to-reader communication is used to transmit in plain text the nonces to be used as a keystream for the reader-to-tag communication. This scenario allows other strategies for the linearity avoidance of LFSRs.

A first straightforward strategy is to suppress the LFSR itself and use a true random data source as a random number generator. Although this approach is theoretically sound, implementations of true random number generators obtain their randomness from the device energy and such energy is very scarce in an EPC Gen2 tag. As a result, the generator throughput cannot reach the minimal requirements of the EPC communication standard. Having this problem in mind, Che *et al.* propose in [CHTW08, CCY+11] the combination of true random numbers (*trn*) extracted from physical effects on tag, and LFSRs to increase the throughput of the generator while decreasing the predictability of the output sequence. In the sequel, we deepen our analysis on the Che *et al.* proposal and show that it fails at handling the linearity of the LFSR output sequence.

3.3 The Che *et al.* Proposal

In [CHTW08, CCY+11], Che *et al.* present a new PRNG for application in RFID tags. Their system relies on an oscillator-based Truly RNG (TRNG), and exploits the thermal noise of two resistors to modulate the edge of a sampling clock and generate the true random bits (*trn*). Authors state the final system prevents potential attackers to perform any effective prediction about the generated sequence (even if the design is known) thanks to the white noise based cryptographic key generation.

After describing its TRNG oscillator-based core, the authors focus on design considerations specially regarding *power consumption* and *output data rates* trade-offs. Knowing the fact that the higher the frequency oscillation of the system, the higher the current (thus also power) consumption, the authors look for system level optimization in order to reduce the power consumption due to the low-power restrictions of RFIDs.

The optimization proposed by Che *et al.* relies on the combination of the TRNG and an LFSR (cf. Figure 3.1). Adding an LFSR to the TRNG lets the system reduce the clock frequency proportionally to the number of cells of the LFSR. Specifically, exploiting the initial state of a 16-bit LFSR combined with the addition of the generated truly random number (*trn*) for each cycle ring, allows the system to decrease the clock frequency with a $\frac{1}{16}$ factor. Authors claim that [CHTW08]: *"If we add 1-bit truly random number in the cycle ring as a random number seed, the output sequence of the LFSR will also be unpredictable and irreproducible as a TRNG."*. We show in the next section that this claim does not hold.

3.3.1 Analyzing and Exploiting the Che *et al.* Proposal

Since the main property of a PRNG is to ensure the forward unpredictability of its generated sequence, the correctness of a PRNG can be measured with statistical tests applied to the output sequence.

We take the National Institute of Standards and Technology (NIST) suit test for checking the randomness deviations of a binary random sequence [NIS08]. NIST testing algorithms

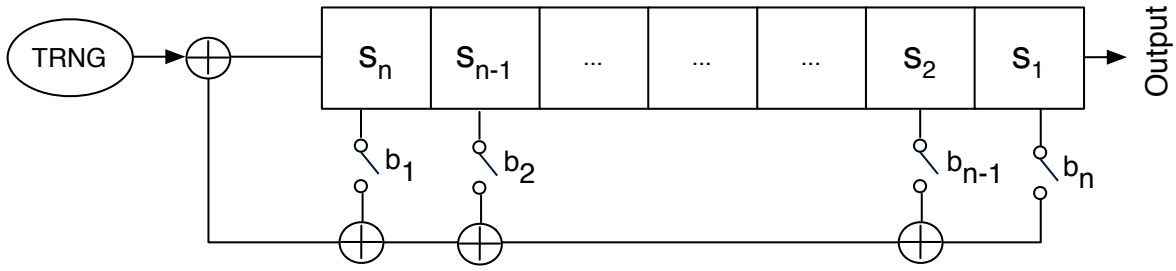


Figure 3.1: PRNG scheme based on the Che *et al.* specifications

use a hypothesis test considering the randomness of the sequence as the *null hypothesis* H_0, and the non-randomness as the alternative hypothesis, H_a. Tests are performed regarding a level of significance or critical value, denoted as α hereinafter.

NIST tests produce *P-values* summarizing the strength of the hypothesis. If *P-values* $\geq \alpha$, H_0 is accepted. It is not necessary that strictly all *P-values* hold this bound for the sequence to be considered as a good pseudorandom sequence. In fact, the NIST recommends that the proportion of test over the significance level, must fit in the interval

$$\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}} \tag{3.2}$$

where $\hat{p} = 1 - \alpha$, and *m* is the sample size. A common value used in cryptography [NIS08] to statistically confirm the randomness of the analyzed data would be $\alpha = 0.01$, that means one would expect 1 in 100 sequences to be rejected. *P-values* passing α give a confidence of 99.9% of the randomness of the evaluated sequence (if 100 sequences are evaluated, results should pass 0.9615 as defined in Equation 3.2). In order to evaluate the randomness quality of the sequence produced by the Che *et al.* scheme, we used matlab and generated 230 MB of output data from an implementation of their proposed PRNG. Such data is divided in ten different data sequences (T_i) that are independently analyzed using the NIST suit tests.

NIST test results for the Che *et al.* random generated data are presented in Table 3.1. Each column represents 23 MB of pseudorandom data generated with different *seed* and

Sequence	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
Frequency	0.99	0.98	0.96	1.00	1.00	0.97	0.97	0.99	0.96	0.97
BlockFrequency	1.00	1.00	0.97	0.98	1.00	0.98	1.00	0.99	0.98	0.99
Runs	0.98	1.00	1.00	0.99	0.99	0.99	0.96	0.98	0.98	1.00
LongestRun	**0.96**	**0.96**	0.98	0.97	**0.94**	0.96	**0.95**	0.98	0.99	**0.94**
Binary Matrix Rank	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
OverlappingTemplate	0.96	**0.94**	0.99	0.98	0.98	**0.93**	0.97	0.98	0.98	**0.95**
Universal	1.00	0.99	0.98	0.99	1.00	0.97	0.99	0.99	0.99	0.98
ApproximateEntropy	0.99	0.97	1.00	0.99	0.98	0.98	0.99	1.00	1.00	1.00
LinearComplexity	0.99	1.00	0.99	0.99	**0.95**	1.00	1.00	0.99	0.99	1.00
CumulativeSums	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$
NonPeriodicTemplate	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{147^\star}{148}$	$\frac{148}{148}$
RandomExcursions	$\frac{7^\star}{8}$	$\frac{7^\star}{8}$	$\frac{8}{8}$	$\frac{7^\star}{8}$	$\frac{8}{8}$	$\frac{7^\star}{8}$	$\frac{8}{8}$	$\frac{6^\star}{8}$	$\frac{8}{8}$	$\frac{8}{8}$
RandomExcursionsVariant	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$
Serial	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$

Table 3.1: Che *et al.* results for the NIST statistical test suite

true random source [Haa98]. Each row refers to a test included in NIST test suite. The first nine tests are represented with the numerical value of the uniformity of *P-values*. The last five tests are in fact a set of different tests thus in order to represent each of the values, an achievement ratio is represented following the same decision rule of the first tests (Equation 3.2). Tests refusing randomness hypothesis are denoted with bold letters in the table. For tests consisting on a set of tests, an asterisk is added when some of the tests are not successfully achieved. In fact, results show a statistical evidence of non randomness for the *Binary Matrix Rank Test* (cf. Table 3.1). Such test constructs binary matrices from the analyzed data and checks for linear dependence among the rows or columns of the constructed matrices. The fact that the *Binary Matrix Rank Test* fails for all the sequences, gives a clear evidence of a non-randomness due to linearity problems.

3.3.2 Exploiting the Linearity Weaknesses of the Scheme

As we have pointed out in Section 3.2, the main vulnerability of a PRNG based on a linear feedback shift register comes from its easy predictability due to its linearity properties.

Results presented in Table 3.1 show that the Binary Matrix Rank Test from the NIST statistical test suite fails for the Che *et al* scheme, providing information that the scheme does not succeed in avoiding the linearity of the underlying LFSR. A specific attack to break the Che *et al*. PRNG based on the inherent linearity of the LFSR was presented in [191] and is next briefly described.

Notice that in the Che *et al.* scheme the pseudorandom sequence is produced by an LFSR XORed in its first cell with a *truly* random bit (cf. Figure 3.1). That means we can find a $2n$ pseudorandom output sequence of the proposed scheme identically equal to the one of the $n-$bit LFSR (without of the XORed true bit) in case that 2 consecutive random bits are 0. Such event will occur with probability $1/4$ assuming bits are true random.

Attack description

Our scenario is composed by a Che *et al.* system that produces pseudorandom bits. Only a part of the pseudorandom output sequence, denoted by s_a is known to the attacker, besides the size n of the LFSR. On the other hand, the *seed* (initial state) and the feedback polynomial coefficients remain secret to the attacker. The attack will succeed if the attacker can provide the LFSR feedback polynomial (cf. Figure 3.2). To generalize the attack, we also assume that the attacker cannot determine the first bit of the sequence, that means he has no information if a given s_a sequence, with $|s_a| = 2n$ (the length of the sequence), has been affected by exactly two trn values (that means the attacker finds two exact LFSR rounds) or the sequence has been modified by three trn values.

With probability $\frac{1}{n}$, the sequence, s_a with $|s_a| = 2n$ has been affected by exactly two trn and, in this case, the probability to obtain the $2n$ values of the LFSR despite the XORed trn is $\frac{1}{4}$ (two consecutive zeros). That means that, with probability $\frac{1}{4n}$, we can obtain $2n$ values of the LFSR that composes the system and with this sequence we are able to compute the feedback polynomial and the whole pseudorandom sequence.

Now, assume that $|s_a| = 3n - 1$. If the sequence is divided into n subsequences of length $2n$, we can ensure that one of these subsequences has been affected by exactly two trn.

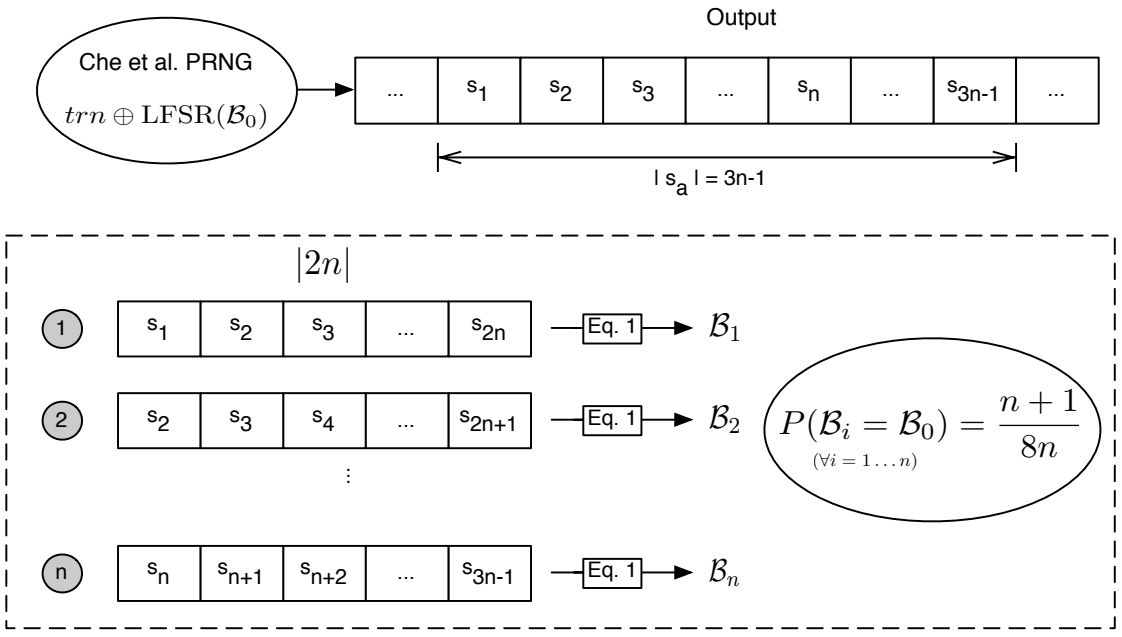


Figure 3.2: Attack scheme to the Che *et al.* PRNG

The remainder $n-1$ subsequences, are affected by three trn. However, notice that if the three trn are zeros, the n vectors of length $2n$ will give the same feedback polynomial. The probability of such event is $\frac{1}{8}$. Then, from this fact, we can derive Equation 3.3 which provides the probability of success of an attack that analyzes a sequence with $|s_a| = 3n-1$:

$$P_{\text{success}}(3n-1) = \frac{1}{4}\left(\frac{1}{n}\right) + \frac{1}{8}\left(\frac{n-1}{n}\right) = \frac{n+1}{8n} \tag{3.3}$$

Obviously, the probability of success increases with $|s_a|$ since increasing the $|s_a|$ implies that more trn bits affect the sequence and then the probability of finding three consecutive zeros also increases. Figure 3.3 shows the probability of success of an attack with s_a length for a particular system with an LFSR of length $n = 16$. Notice that only 160 bits $(10n)$ are enough to perform a successful attack with probability higher than 50%, and 464 bits $(29n)$ implies more than a 90% of success probability.

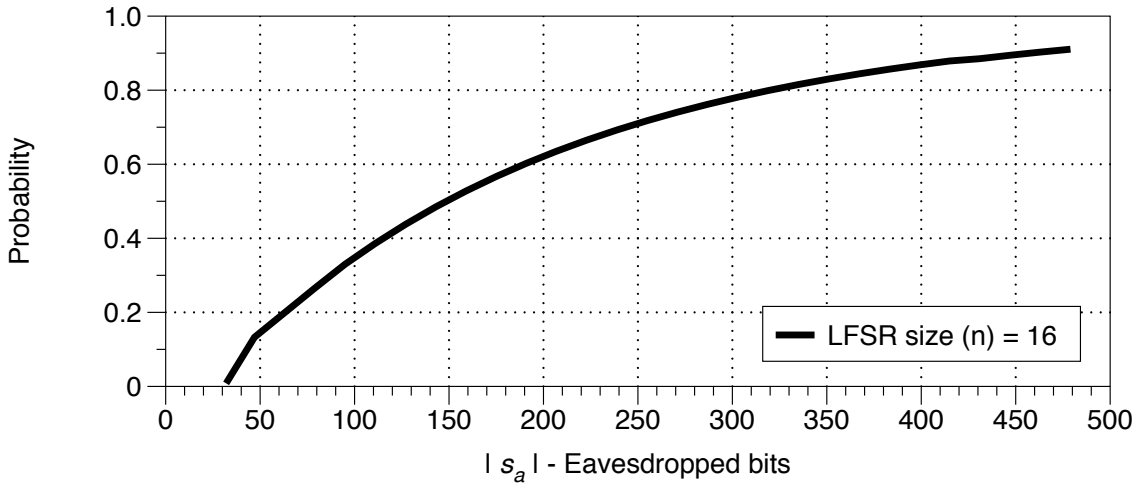


Figure 3.3: Reliability on the Che *et al.* attack regarding $|s_a|$

Obtained Results

To test the correctness of the theoretical evaluation, the described attack is implemented over the same ten pseudorandom sequences (T_i) used to execute the NIST tests (cf. Section 3.3.1).

The first analysis validates that the probability of finding the feedback polynomial matches the one described in Equation 3.3. In this case, the algorithm takes $|s_a| = 3n - 1$ bits from T_i starting at a random position and tries to attack the system by finding n equal feedback polynomials. The operation is repeated one thousand times for each test sequence T_i. Attack success rates are reported in Table 3.2. Notice that they are close to the theoretic value $\frac{(n+1)}{8n}$ with $n = 16 \approx 0,132$.

The second analysis provides the number of bits needed to achieve a successful attack. Ten different attacks are performed for every T_i data sequence taking the first bit of s_a at random. Results presented in Table 3.3 show the number of bits for a successful attack in the worst case, that is the attack that needs a major number of bits. Notice that, although taking the worst case, the number of bits is significantly lower than the whole period $2^{16}-1$.

Sequence	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
attack success (%)	0.132	0.137	0.131	0.126	0.139	0.137	0.129	0.137	0.138	0.128

Table 3.2: Attack success rate for $|s_a| = 3n - 1$

Sequence	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
$\|s_a\|$	238	254	254	190	510	158	254	286	238	222

Table 3.3: Value of $|s_a|$ for a successful attack in the worst case after 10 tests

3.4 Attack Implementation and Empirical Results

In this section, we present an empirical version of our proposed attack on a real EPC Gen2 setup. We describe the RFID devices used to implement the attack, the techniques used to eavesdrop the PRNG from the RFID communication, and the obtained results.

Background on the IAIK UHF Demo Tag

The *IAIK UHF demo tag* [SIC07] is a programmable device intended for developing new commands or functionalities to the EPC Gen2 standard. It allows, moreover, to verify the new functionality using compliant EPC Gen2 readers. We use this prototype to demonstrate and validate the concepts discussed in the previous sections.

The demo tag consists of four main components: an antenna, a radiofrequency (RF) front-end, a programmable microcontroller, and a firmware library. The antenna captures the energy emitted by the reader and powers up the RF front-end of the tag. The RF front-end demodulates the information encoded in the signal. The resulting data feeds the programmable microcontroller which, in turn, computes a response. To compute the response, the programmable microcontroller executes a software implementation of the EPC Gen2 protocol, implemented as a firmware library. The response is then modulated by the RF front-end and backscattered to the reader. We present in the sequel a condensed background on these four components. More details can be obtained in [SIC07, APF+07].

Antenna and RF Front-end

The antenna connected to the RF front-end consists of a 17cm dipole antenna. The RF front-end utilizes a two-stage charge-pump rectifier to perform amplitude-shift keying (ASK) demodulation. It demodulates the information stored in the signal transmitted on the reader-to-tag channel. It does, indeed, rectification, voltage multiplication, and envelope detection all at once [APF+07]. The power extracted by the rectifier from the RF field emitted by the reader from most compliant EPC Gen2 readers amounts to about 2.4mW. Since this is not enough to power the microcontroller, the demo tag adopts a semi-passive approach, meaning that although the analog parts are powered by the energy harvested from the reader, the digital parts (e.g., the programmable microcontroller) are powered by an external power supply or by an on-board battery. The backscattering of the information computed by the programmable microcontroller consists of the reflected power of the antenna. This power is indeed generated according to the transmitted data. The RF front-end of the demo tag combines both ASK and PSK (phase-shift keying) to modulate information. The backscattering components used by the demo tag to modulate the tag-to-reader signals consist of a resistor, a capacitor, and a fast-switching transistor placed close to the antenna. These components are controlled by the programmable microcontroller.

Programmable Microcontroller and Firmware Library

The programmable microcontroller connected to the RF front-end of the demo tag consists of an Atmel AVR ATmega128 (cf. `http://www.atmel.com/`). It contains all the logic and memory necessary for the demo tag. The ATmega128 is an 8-bit microcontroller based on the AVR architecture. The memory banks of the microcontroller, 128KB of flash memory and 4KB of data memory, can be addressed by three independent 16-bit registers. In addition, the ATmega128 has 32 registers of 8-bits. All 32 registers can act as the destinations of the ATmega128 arithmetic operations. The microcontroller operates exactly one instruction per clock cycle, at frequencies up to the order of 16MHz. An external crystal oscillator connected to the demo tag provides the 16MHz signal to the microcontroller. Three

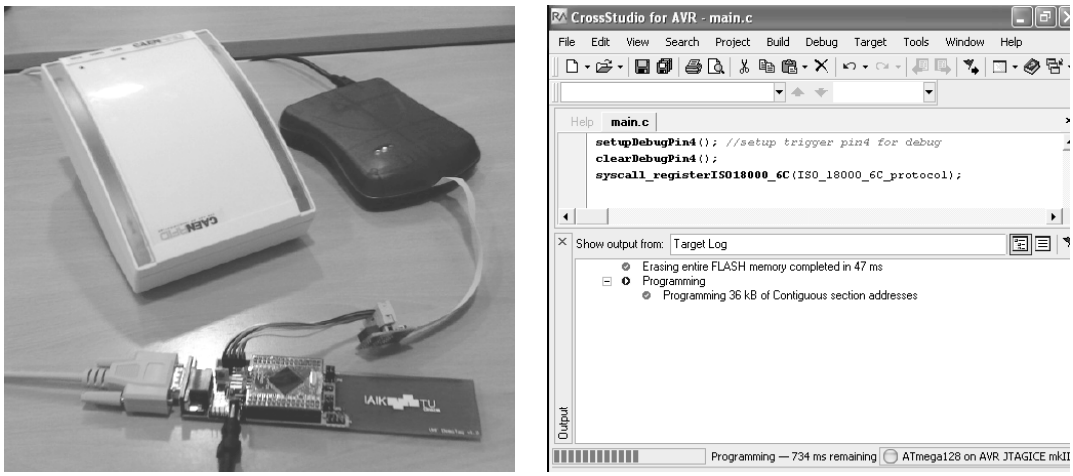
main signals connect the microncontroller to the RF front-end. A first signal, called DEMOD, provides the demodulated ultra high frequency (UHF) signal from the reader-to-tag channel. A second signal, called MOD, allows the ATmega128 to control the backscatter used to generate the tag-to-reader responses. Finally, a third signal, called RF ON, provides a boolean value to detect the presence of the RF field.

The original IAIK UHF demo tag already provides an appropriate implementation of the EPC Gen2 protocol for the ATmega128. The protocol is implemented as a firmware library stored in the flash memory of the microcontroller. This library contains all the functions necessary to process the readers' standard queries and to compute the appropriate responses. The microcontroller is connected, via an UART module, to a serial-interface connector. This serial interface allows to interact with the demo tag, to provide basic operations such as memory mapping, EPC Gen2 values' configuration, visualization of queries and responses exchanged with compliant readers, and execution of user defined operations. This latter allows to complement the original protocol implementation with new functionalities defined at a user level. By using the JTAG connector provided by the demo tag, it is possible to upload new functionalities to the flash memory of the microcontroller, as well as to perform program debugging. A combination of C code and assembly code can be used to complement or modify the original firmware library. A JTAG download cable allows the transfer of new functionalities or firmware updates. Some other modules connected to the demo tag allow more complex programming possibilities, such as FPGA-based UHF protocol implementations. We refer the reader to [SIC07, APF+07], and citations thereof, for more information.

3.4.1 Che *et al.* Implementation and Experimental Setup

In Section 3.3.2, we have seen how to attack the pseudorandom number generator proposed by Che *et al.* once a sufficient number of pseudorandom values are collected. We show in this section the results of a practical attack against the vulnerable scheme on a real Gen2 setup. The attack is based on the eavesdropping of the communication between a standard EPC Gen2 reader and the aforementioned demo tag. Indeed, we show how it is possible to

obtain an appropriate set of random queries generated by an on-board PRNG, based on the Che *et al.* scheme, to eventually predict the generation of pseudorandom sequences that will be generated later over the demo tag. Figure 3.4 shows our experimental setup. More details are available in [188].



(a) (b)

Figure 3.4: Experimental setup. In (a), we can see a CAEN A829EU UHF RFID Reader, the AVR JTAG MKII Programmer, and the IAIK Graz UHF Demo Tag. In (b), we can see the Crossworks IDE GUI for AVR, uploading the updated firmware over the demo tag

The Che *et al.* scheme has been implemented in ANSI C using the Crossworks IDE for AVR from Rowley Associates (cf. `http://www.rowley.co.uk/`). The original scheme provided in [CHTW08] has been adapted into a code-optimized EPC Gen2 version that can be executed over the microcontroller of the IAIK UHF demo tag. Arithmetic efficient functions such as *bit shifts*, logic operators (*AND*, *OR* and *XOR*) and *modulo 2*, are used to implement the LFSR in the demo tag [Sch96]. The *trn* addition is extracted from the less significant bits of the analogical to digital conversion in the demo tag's microcontroller. Since the generation of pseudorandom sequences is a mandatory operation specified in the EPC Gen2 protocol, an existing PRNG function is already included in the original firmware. By using the Crossworks IDE, we code and merge the PRNG based on the Che *et al.* scheme with the general firmware library to replace the existing PRNG. The JTAG programmer that we use to transfer and to debug the updated

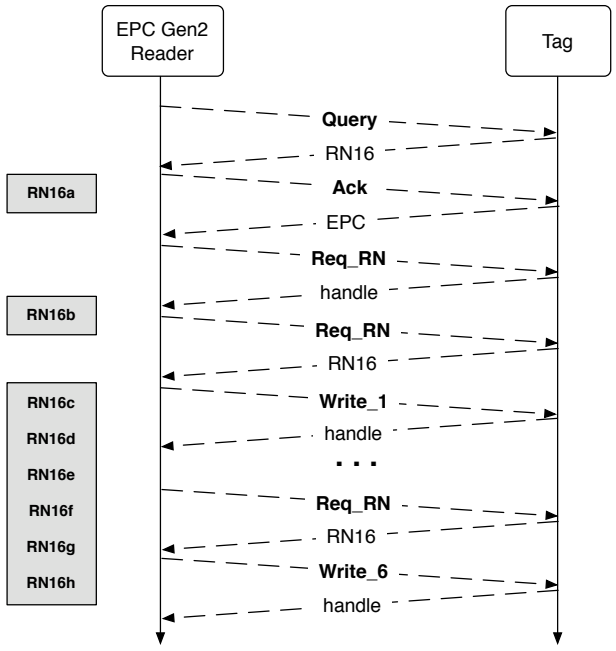
firmware merged with the new PRNG implementation is an AVR JTAG MKII programmer (cf. `http://www.atmel.com/`). The queries are generated from a standard RFID reader according to EPC Gen2. The RFID reader we use is a short-range reader CAEN A829EU (cf. `http://www.caen.it/rfid`). The reader is controlled by a desk computer over a USB serial port. For the generation of queries, we use a .NET application that controls the communication process with the reader. This application enables us to generate the set of queries required to proceed with the eavesdropping attack. We use matlab to decode the set of responses generated over the demo tag. This operation enables us to isolate the pseudorandom queries computed at the demo tag. When the number of sequences collected by the application reaches an appropriate threshold, it proceeds to execute the implementation of the attack we presented in Section 3.3.2. We provide in the sequel further details about the collection of pseudorandom sequences and the practical results.

3.4.2 Eavesdropping of Control Sequences and Practical Results

Due to the Gen2 RF power range characteristics, a realistic attack should only consider reader-to-tag queries because they are much easier to be eavesdropped [PLHCETR09]. Some reader-to-tag queries include pseudorandom sequences (hereinafter denoted as RN16s) that are computed from the on-board PRNG included on the EPC tags. Table 3.4 shows the mandatory operations for Gen2 reader-to-tag protocol and the minimum number of RN16s involved in each operation. Notice that the *write* command generates a minimum of eight RN16s for its proper execution. For a full EPC code writing, up to six RN16s must be generated to cover the reader-to-tag communication, besides the two previously generated pseudorandom sequences for the inventory query and the handle descriptor [EPC08b].

Operation	Inventory	Access			
Command	Identification	Read	Write	Lock	Kill
Number of RN16s	1	2	8	2	4

Table 3.4: Minimum number of RN16s involved in EPC Gen2 operations



(a) EPC Gen2 *write* protocol.

=> QUERY ...
=> ACK ...
=> Req_RN RN:14438
=> Req_RN RN:44282
=> Write (data) 27698
=> Req_RN RN:44282
=> Write (data) 47380
=> Req_RN RN:44282
=> Write (data) 44282
=> Req_RN RN:44282
=> Write (data) 60868
=> Req_RN RN:44282
=> Write (data) 32656
=> Req_RN RN:44282
=> Write (data) 34674

(b) Real capture of six *write* cycles.

Figure 3.5: *Write* process for EPC Gen2 and the PRNG utilization. In (a), we can see the six cycles of the EPC Gen2 *write* command. In (b), we can see a real sample of six *write* cycles captured from the reader-to-tag channel

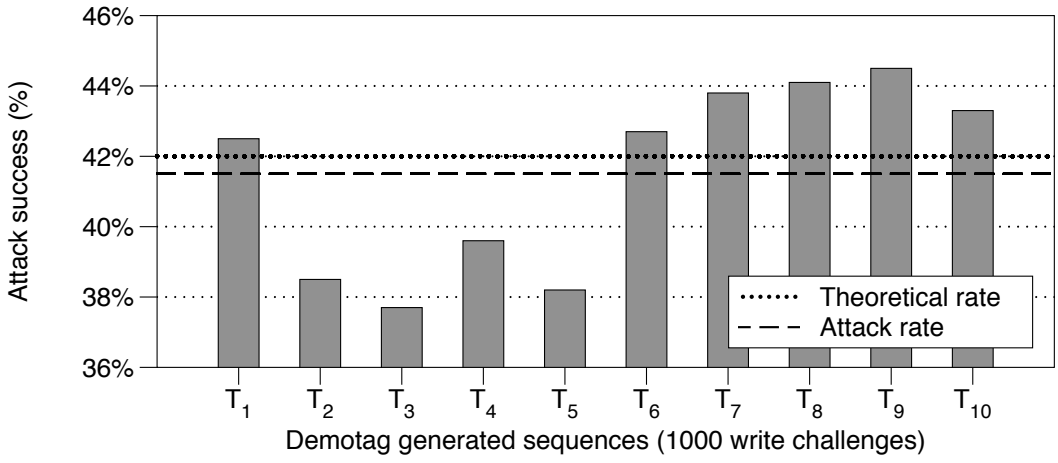


Figure 3.6: Che *et al.* PRNG attack success for real Gen2 environment

A *write* operation is an access command used to modify specific areas of a Gen2 tag memory. The reader first identifies the tag with *select* and *inventorying* commands (what shifts the tag from *ready* to *acknowledged* state). Once the tag is *acknowledged* (meaning that the tag has sent its EPC identification) the reader requests a new RN16 to the tag for establishing an *access* session. The new RN16 (denoted as handle) acts as a session key, and must be used to link all the *access* actions to a specific tag. Observe that all access commands can be executed both in the *open* or *secured* tag state [EPC08b]. If the accessed tag is in the secured state, it means a 32-bit password (exchanged as two 16-bit half-passwords XORed with two RN16s) is necessary to allow the reader to access the tag. In our experiments, we assume that the tag is in the open state, i.e., we do not consider the capture of PRNGs derived from the exchange of the two half-passwords. This way, an inventoried tag transitions directly to the access mode. For a *write* operation once the reader gets the handle, it initiates a round of writes of 16-bit data sequences (obscured with previously requested RN16s) to the tag. Thus, if a new EPC identification is written to the tag, six write cycles are performed, as we picture in Figure 3.5(a).

The eight generated RN16s represent 128 consecutive bits generated from the PRNG of an EPC Gen2 tag, as specified in the standard [EPC08b]. As we pointed out in Section 3.3.2, the Che *et al*. scheme can be predicted with a reasonable small amount of data. We can now demonstrate this property in our real Gen2 environment, by simply performing an appropriate series of *write* challenges to the adapted Che *et al.* PRNG implemented over the demo tag, and analyzing the resulting RN16s. More precisely, we show that by simply collecting 128 bits (generated from a series of eight RN16s associated to each *write* challenge) is enough to obtain the feedback polynomial of the LFSR with a confidence of about 42%. This value is consistent with the analytical results we anticipated in the previous section, and that are depicted in Figure 3.3. Figure 3.5(b) shows a simple example where six write cycles are captured. These captures allow us to collect 96 pseudorandom bits generated from the on-board Che *et al*. PRNG. The sequences are parsed from the matlab code that we feed with the serial interface output of the demo tag. Only reader-to-tag challenges are shown. The reader writes the EPC identification to 0 ($EPC_{96b} = 0_{i\ b...(i+16)\ b} \oplus RN16$), to obtain the RN16s directly from the ciphered data field of the *write* challenges.

The complete set of experiments that we summarize in Figure 3.6 consists of ten series of *write* commands. Each of these series generates a total of 1,000 *write* challenges from the A829EU reader to the demo tag. As a result, 8,000 RN16s, i.e., 128,000 pseudorandom bits, are captured in total. These pseudorandom bits are computed from the Che *et al.* PRNG implementation deployed over the demo tag. Once stored, the pseudorandom sequences are processed by the matlab code that contains the attack implementation. Let us recall that the attack applies the analysis of the linearity relation for each single *write* challenge. We show that the attack finds the appropriate feedback polynomials of the LFSR each 128 bits with a total ratio of success of 41.5%. This result is very close to the 42% that we predicted in Section 3.3.2 (Figure 3.3). Therefore, we are able to confirm the vulnerability of the Che *et al.* PRNG for Gen2 environments.

3.5 Concluding Summary and Remarks

Pseudorandom Number Generators (PRNGs) are the crucial components that guarantee the confidentiality of EPC Gen2 [EPC08b] RFID communications. In this chapter, we have described the problems of using linear feedback shift registers (LFSRs) as underlying mechanisms for the implementation of low-cost PRNGs. Without appropriate measures that increase their cost, the linearity of LFSR-based PRNGs lead to insecure implementations. We have analyzed a cost-effective PRNG proposal for EPC Gen2 devices presented by Che *et al.* [CHTW08, CCY+11]. The proposal combines thermal noise signal modulation and an underlying LFSR.

We have demonstrated that the proposal does not handle properly the inherent linearity of the resulting PRNG. We have described an attack to obtain the feedback polynomial function of the LFSR. This allows us to synchronize and to predict the resulting sequences generated by the Che *et al.* PRNG. We have presented the implementation of a practical attack in a real EPC Gen2 scenario. By means of a compatible Gen2 reader, and a programmable Gen2 tag [SIC07] implementing the Che *et al.* PRNG, we have shown that an

attacker can obtain the PRNG configuration with a confidence of 42% by only eavesdropping 128 bits of pseudorandom data. Although the attack implementation has been applied to a specific PRNG proposal, the procedure used to obtain the data is based on standard EPC commands and it can be applied to any EPC tag communication to eavesdrop the output of the PRNG.

A solution to handle the linearity problem of the Che *et al.* PRNG is adapting its architecture towards a Multiple-Polynomial LFSR construction. By adding multiple feedback polynomials to the LFSR, and by feeding the selection of polynomials with the same physical source of randomness proposed by Che *et al.* in [CHTW08, CCY+11], we can successfully avoid the inherent linearity of LFSR based PRNGs and satisfy the security requirements. A sample construction based on this idea is presented in the following chapter. Statistical analysis of the sequences generated by such a generator confirmed the validity of the proposed technique. An electronic circuit simulation confirmed, moreover, that the proposal has a simpler hardware implementation than previous schemes reported in the literature.

Chapter 4

Multiple-Polynomial LFSR based Pseudorandom Generator

We address in this chapter the linearity issues of the Che *et al.* PRNG reported in Chapter 3. We present a novel design, named J3Gen. J3Gen is based on a linear feedback shift register (LFSR) configured with multiple feedback polynomials. The polynomials are alternated during the generation of sequences via a physical source of randomness. A concrete hardware implementation of J3Gen is presented and evaluated with regard to different design parameters, defining the key-equivalence security and the achievement of the EPC Gen2 requirements. The results of a SPICE simulation confirm the power-consumption suitability of the proposal. Parts of this chapter have been previously published in [187, 193, 195].

Chapter Outline: Section 4.1 presents a high-level description of the J3Gen proposal. Section 4.2 presents a sample execution of J3Gen based on a 16-bit LFSR-based version. Section 4.3 determines an optimal selection of parameters to satisfy the EPC Gen2 requirements while guaranteeing a proportional degree of security. Section 4.4 evaluates the statistical properties of the proposed setup for EPC Gen 2, and evaluates its hardware complexity and power consumption. Section 4.5 discusses on the benefits and limitations of our proposal with regard to related work. Section 4.6 closes the chapter.

4.1 Proposal Design

The main challenge to obtain an efficient PRNG is how to guarantee the generation of sequences with (almost) true random properties, while also addressing efficiency and computational complexity. Indeed, the low power, chip area and output rate (among other constraints) of the EPC Gen2 technology makes the task of improving security harder. This is the case of true random number generator (TRNG) designs based on, e.g., thermal noise, high frequency sampling or fingerprinting, whose requirements of power consumption or computational complexity for full-length real-time generation of random sequences fall out of EPC Gen2 standards [EPC07]. We propose to address this problem by combining a physical source of true randomness and a deterministic linear feedback shift register (LFSR). That is, leveraging the physical source system requirements with the efficiency of LFSRs for hardware implementations.

Figure 4.1 depicts a block diagram of the J3Gen proposed design. It gets inspiration from a dynamic LFSR-based testing selection scheme presented by Hellebrand *et al.* in [HRT$^+$95, RAHN03]. Indeed, it substitutes the static feedback polynomial configuration of an LFSR by a multiple feedback primitive polynomials configuration architecture. The different feedback primitive polynomials are connected to the LFSR by a decoding matrix that selects each single feedback polynomial. After a given number of LFSR cycles, the *Polynomial Selector Module* shifts its position towards a new configuration. The number of shifts, i.e., the corresponding selection of each primitive polynomial at a certain LFSR cycle, is determined by a true random bit (hereinafter denoted as trn) that is obtained from a physical source of randomness. Next, the four main design modules are described. A detailed step by step sample execution of our construction is shown in Section 4.2.

LFSR Module: The J3Gen generator relies on a n-cell LFSR module. LFSRs produce pseudorandom sequences with good statistical values. They are very fast and efficient in hardware implementations, and quite simple in terms of computational requirements [MOV01]. This makes the use of LFSRs an ideal system for both energy and computational constrained environments. Moreover, LFSRs follow the same hardware scheme as

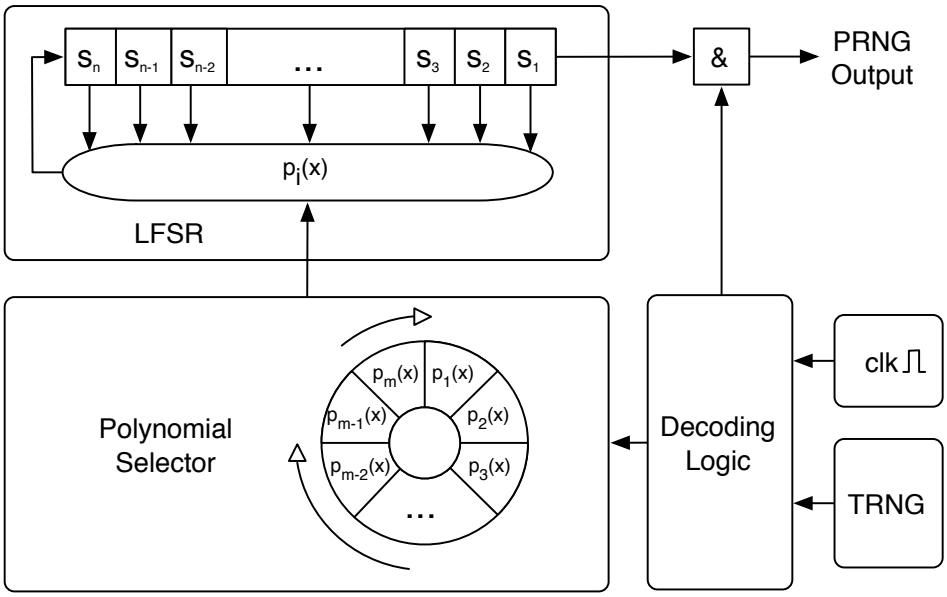


Figure 4.1: Block diagram of J3Gen

those cyclic redundancy check (CRC) functions already included in the EPC Gen2 standard [EPC07]. Therefore, current EPC Gen2 tags shall be able to execute LFSR-based functions in the same hardware.

Polynomial Selector Module: The Polynomial Selector is the responsible of the linearity avoidance of J3Gen. A set of m primitive feedback polynomials is selected, and each single feedback polynomial is used depending on the value of the truly random bit provided by the TRNG module. The feedback polynomials are implemented as a *wheel*, which rotates depending on the bit value given by the TRNG module. If the truly random bit is a logical 0, the *wheel* rotates one position, that is, it selects the next feedback polynomial. Instead, if the truly random bit is a logical 1, then the *wheel* rotates two positions, that is, the Polynomial Selector jumps one feedback polynomial and selects the next one.

Decoding Logic Module: The Decoding Logic is the responsible for managing the internal PRNG clock of J3Gen. It activates and deactivates the PRNG modules for its proper

performance. The internal PRNG modules have different activation and deactivation timings. Depending on the internal clock frequency, f_{clk}, some modules such as the LFSR or the TRNG need different activation cycles. For example, the trn sampling in the TRNG module is activated only once for each PRNG output.

The Decoding Logic also manages the trn obtained from the TRNG module, rotating the Polynomial Selector with regard to the trn value. This action is performed using ℓ cycles, with $1 \leq \ell \leq n-1$. This value is lower than the n cycles needed in the LFSR to avoid pseudorandom sequences generated from a single feedback polynomial. This way, the generated sequence does not have linearity flaws. Common attacks to retrieve the equivalent LFSR generator, like the Berlekamp-Massey algorithm [Mas69], are not able to perform.

Thermal-noise TRNG: Regarding the physical source of randomness (trn), there are different proposals to derive true random sequences of bits from the hardware of a radio-frequency identification (RFID) tag. The technique used in J3Gen is the oscillator-based high frequency sampler by Che *et al.* [CHTW08, CCY+11], that offers high simplicity and suitability for EPC Gen2 designs. To leverage the high power consumption of this technique for EPC Gen2 standards [CHTW08, CCY+11], the TRNG output is optimized with the LFSR Module, which divides the TRNG operating frequency by n as described above. The output of the TRNG is fed to the Decoding Logic which, in turn, manages the Polynomial Selector.

4.2 Sample PRNG Execution

Once described the logic components of the system, we depict now a sample execution round with our proposal. Table 4.1 shows a sample selection of parameters to construct a 16-bit version of J3Gen. Table 4.2 specifies as well a possible selection of feedback polynomials. The LFSR size n has been fixed to 16 and the total number of different feedback polynomials m has been set to 8. We take as the initial LFSR state the value $v_0 = 0x1$ (hexadecimal notation that represents a logical 1 in the less significant bit).

Table 4.1: Design parameters summary

Size of LFSR (bits)	$n = 16$
Number of feedback polynomials on tag	$m = 8$
trn sampling period	$f_r = 16$
Polynomial Selector update period	$\ell = 15$

Table 4.2: Feedback polynomials ($n = 16$)

Primitive polynomials
$p_1(x) : 1 + x + x^5 + x^6 + x^7 + x^{11} + x^{16}$
$p_2(x) : 1 + x^4 + x^5 + x^6 + x^7 + x^{11} + x^{16}$
$p_3(x) : 1 + x + x^3 + x^4 + x^5 + x^6 + x^7 + x^{11} + x^{16}$
$p_4(x) : 1 + x^3 + x^5 + x^6 + x^{10} + x^{11} + x^{16}$
$p_5(x) : 1 + x^5 + x^6 + x^{11} + x^{16}$
$p_6(x) : 1 + x^5 + x^6 + x^{10} + x^{11} + x^{13} + x^{16}$
$p_7(x) : 1 + x^4 + x^5 + x^6 + x^{10} + x^{11} + x^{16}$
$p_8(x) : 1 + x + x^3 + x^4 + x^5 + x^6 + x^{10} + x^{11} + x^{16}$

Table 4.3 details each LFSR state for 32 shift cycles (rows) providing 32 outputted PRNG bits (column *Tx*) consisting in two 16-bit sequences. Since the trn sampling frequency is $f_r = 16$ cycles, this 32 shift cycles need two true random values, that in the example have been set to $r_1 = 0$ and $r_2 = 1$.

The system starts with $p(x)_1$ and outputs $\ell = 15$ bits until the TRNG module transfers a bit with value $r_0 = 0$ to the Decoding Logic module. Then, a consecutive (but different) feedback polynomial is selected in the Polynomial Selector module, that is, $p_2(x)$. This generates the next $\ell = 15$ LFSR shifts with $p_2(x)$ until the next trn is obtained. The trn value for this PRNG update is $r_1 = 1$, hence, the Decoding Logic rotates the Polynomial Selector one position at shift 31, and another position at shift 32. Then, $p_2(x)$ is used 14 cycles, $p_3(x)$ is used one cycle, and $p_4(x)$ is used one cycle in this PRNG update and 14 cycles in the next PRNG update (not included in Table 4.3).

Table 4.3: LFSR iteration example

	S_{16}	S_{15}	S_{14}	S_{13}	S_{12}	S_{11}	S_{10}	S_9	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1	**Tx**
v_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Coeff.	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8	x^9	x^{10}	x^{11}	x^{12}	x^{13}	x^{14}	x^{15}	x^{16}	
$p_1(x)$	1	0	0	0	1	1	1	0	0	0	1	0	0	0	0	1	
1:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**
2:	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
3:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
4:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	**0**
5:	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	**0**
6:	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	**0**
7:	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	**0**
8:	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	**0**
9:	0	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	**0**
10:	1	0	1	0	0	1	1	1	1	1	0	0	0	0	0	0	**0**
11:	1	1	0	1	0	0	1	1	1	1	1	0	0	0	0	0	**0**
12:	1	1	1	0	1	0	0	1	1	1	1	1	0	0	0	0	**0**
13:	1	1	1	1	0	1	0	0	1	1	1	1	1	0	0	0	**0**
14:	1	1	1	1	1	0	1	0	0	1	1	1	1	1	0	0	**0**
15:	0	1	1	1	1	1	0	1	0	0	1	1	1	1	1	0	**0**
Coeff.	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8	x^9	x^{10}	x^{11}	x^{12}	x^{13}	x^{14}	x^{15}	x^{16}	
$p_2(x)$	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0	1	
16:	0	0	1	1	1	1	1	0	1	0	0	1	1	1	1	1	**0**
17:	1	0	0	1	1	1	1	1	0	1	0	0	1	1	1	1	**1**
18:	1	1	0	0	1	1	1	1	1	0	1	0	0	1	1	1	**1**
19:	1	1	1	0	0	1	1	1	1	1	0	1	0	0	1	1	**1**
20:	1	1	1	1	0	0	1	1	1	1	1	0	1	0	0	1	**1**
21:	0	1	1	1	1	0	0	1	1	1	1	1	0	1	0	0	**1**
22:	1	0	1	1	1	1	0	0	1	1	1	1	1	0	1	0	**0**
23:	0	1	0	1	1	1	1	0	0	1	1	1	1	1	0	1	**0**
24:	0	0	1	0	1	1	1	1	0	0	1	1	1	1	1	0	**1**
25:	0	0	0	1	0	1	1	1	1	0	0	1	1	1	1	1	**0**
26:	0	0	0	0	1	0	1	1	1	1	0	0	1	1	1	1	**1**
27:	1	0	0	0	0	1	0	1	1	1	1	0	0	1	1	1	**1**
28:	1	1	0	0	0	0	1	0	1	1	1	1	0	0	1	1	**1**
29:	1	1	1	0	0	0	0	1	0	1	1	1	1	0	0	1	**1**
30:	0	1	1	1	0	0	0	0	1	0	1	1	1	1	0	0	**1**
Coeff.	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8	x^9	x^{10}	x^{11}	x^{12}	x^{13}	x^{14}	x^{15}	x^{16}	
$p_3(x)$	1	0	1	1	1	1	1	0	0	0	1	0	0	0	0	1	
31:	1	0	1	1	1	0	0	0	0	1	0	1	1	1	1	0	**0**
Coeff.	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8	x^9	x^{10}	x^{11}	x^{12}	x^{13}	x^{14}	x^{15}	x^{16}	
$p_4(x)$	0	0	1	0	1	1	0	0	0	1	1	0	0	0	0	1	
32:	1	1	0	1	1	1	0	0	0	0	1	0	1	1	1	1	**0**

4.3 Selection of Parameters

Although J3Gen can be used as a security tool in multiple lightweight ubiquitous computing scenarios, we look for compatibility with the EPC Gen2 requirements. The EPC standard [EPC07] does not define any hardware requirement for the generation of the 16-bit pseudorandom sequences, nor any other hardware security features regardless of the CRC which shares the LFSR scheme with our proposal (cf. Section 4.1). Authors in the literature do not agree about the implementation area which can be devoted to security. Using logical gates equivalence (GE) to measure the implementation size (regardless of the manufacturing process), some authors defend that only 2,000 GE can be used for security [Jue06, WSRE03b], while others increase the range from 2,000 to 5,000 [RC08]. In this chapter, the worst case is assumed. Hence, J3Gen looks for a suitable trade-off between security and hardware implementation cost, that is, maximizing the security inside the proposed implementation area of 2,000 GE. This implementation area is approximately $10,400\,\mu\mathrm{m}^2$ using current $130\,\mathrm{nm}$ process for static CMOS designs.

4.3.1 Hardware

The size and design of each component of J3Gen implies a specific hardware implementation, being the LFSR size (n) and the number of implemented feedback polynomials on tag (m) the parameters that most significantly impact on the hardware complexity of our design. Furthermore, both parameters are also the key parameters from the security point of view. Hence, we shall look for the different combinations of n and m within the hardware implementation boundaries to find the best security implementation for this purpose.

Table 4.4 represents the different implementation counts (based on GE area), depending on the main PRNG parameters n and m, as described in Section 4.1. The n-bit register is included in the LFSR module, the LFSR feedback and decoder are included in the Polynomial Selector module, and the cycle clock, TRNG and PRNG output are included in the Decoding Logic module. These three elements add up to the most representative amount of GEs.

LFSR size (n)	16			24			32			64		
Feedback polynomials (m)	8	16	32	8	16	32	8	16	32	8	16	32
LFSR module	72.0	72.0	72.0	108.0	108.0	108.0	144.0	144.0	144.0	288.0	288.0	288.0
Polynomial Selector module	209.3	396.6	774.1	305.1	577.6	1,125.3	401.0	758.6	1,476.5	784.3	1,482.4	2,881.3
Decoding Logic module	48.3	48.3	48.3	53.3	53.3	53.3	53.3	53.3	53.3	61.3	61.3	61.3
TRNG	22.0	22.0	22.0	22.0	22.0	22.0	22.0	22.0	22.0	22.0	22.0	22.0
Additional Control	87.5	125.0	182.1	114.9	169.3	279.9	141.2	212.7	356.3	249.3	387.9	667.7
TOTAL	**439.1**	**663.9**	**1092.5**	**603.3**	**930.2**	**1,602.8**	**761.5**	**1,190.6**	2,052.1	**1,419.3**	2,241.6	3,920.3

Table 4.4: Logical GE Count for J3Gen

The remainder GEs mainly consist on the necessary extra circuitry for controlling the different states of the generator. Logic gates considered in this implementation count are basic two-input gates, except for the *decoder* (Polynomial Selector) where $(log(m)/log(2))$-input NAND gates are used depending on the number of implemented polynomials m. For the LFSR implementation purpose, we use the *D-flip-flop* (DFF) model specified at [Bak07] composed by 18 CMOS transistors. Hence, a DFF can be measured with approximately 4.5 GE.

As shown in Table 4.4, the Polynomial selector module implementation hardly depends on the total number of polynomials m that will be used as a pool of feedback polynomials. Furthermore, the exact selection of the m polynomials also affects the total number of GE needed for the implementation. To determine the GE number for this module, a first approach could be to analyze all combinations of the m possible polynomials and take the combination that needs the minimum GE. Although this strategy seems to be the best one regarding the implementation purposes for its efficiency (it will end up with the simplest implementation possible), it is not a good choice regarding security needs. As we will describe in the next section, the exact value of the m polynomials (not the value m itself) can be seen as a security key (since, together with the trn values, it determines the output of the pseudorandom sequence). Then, an attacker could determine the exact combination of the used polynomials simply by finding the combinations that produce the optimal implementation in GE. To avoid that, we compute the GE needed in this module by analyzing

all the possible polynomial combinations and then take the worst possible case. Since any chosen combination of the m polynomials is equally probable, an attacker cannot discard any of them regarding its implementation suitability. Based on this strategy, we show in Table 4.4 the resulting GE upper bound values.

We then provide the physical source of randomness assumed for our generator. For the gate equivalence of this component, we based our estimations on previous works presented in [RC08] and [PPR09]. The physical source of randomness that we assume consists of the thermal-noise oscillator presented by Che *et al.* in [CHTW08, CCY$^+$11], but specified and modeled in our work as proposed in [PC96] and [ZZW08].

From Table 4.4, it can be extracted that implementations using up to 32 cells for the LFSR are roughly EPC Gen2 suitable from the hardware perspective. Also, a combination of large LFSR with a small pool of polynomials (e.g., $n = 64$ and $m = 8$) offers a possible solution, regarding hardware constrains (bold values in the Total row). In the next subsection, we overview the security properties of our scheme for those parameters that fit the hardware constrains, discarding implementations surpassing the available implementation area, established in 2,000 GE.

4.3.2 Security

EPC Gen2 security relies on the PRNG utilization, and how the PRNG ciphers the *Access* or *Kill* operations [EPC07] to avoid eavesdroppers to obtain the cleartext of the transmitted sequences. That is, the security of an EPC Gen2 PRNG is based on the unpredictability of its output. In J3Gen, such unpredictability is based on the non-linearity module that depends on the combination of the selected m feedback polynomials, and the feedback polynomial update rate ℓ. In our scenario, assuming that n, the length of the LFSR, is a public value, the knowledge of the exact combination of m feedback polynomials would allow to an attacker to predict the output sequence. In that context, since such polynomials are kept secret, they may be considered as the secret key of our PRNG. Then, the security strength of J3Gen can be measured as a key length, understanding each key as every different possible m feedback polynomial combination.

LFSR size (n)	Primitive polynomials	Num. of pol. (m)	Possible combin.	Gate Eq. (GE)	Ratio (bit / GE)
16	2,048	8	2^{73}	439.1	0.1662
		16	2^{131}	663.9	0.1973
		32	2^{234}	1092.5	0.2141
24	276,480	8	2^{129}	603.3	0.2138
		16	2^{245}	930.2	0.2633
		32	2^{461}	1,602.8	0.2876
32	67,108,864	8	2^{192}	761.5	0.2521
		16	2^{372}	1,190.6	**0.3124**
64	1.44×10^{17}	8	2^{441}	1,419.3	0.3107

Table 4.5: Combinations using primitive polynomials

In order to achieve the best statistical properties, feedback polynomials of LFSRs shall be primitive [MOV01]. Table 4.5 shows, for each LFSR length n, the total number of existing primitive polynomials. Given such value, each possible number of m chosen feedback polynomials determines the total available combinations that can be taken to implement our scheme. Each combination represents a possible key, so the powers of the values of the fourth column (labeled as *Possible combin.*) in Table 4.5 determines the length of the key (in bits). At this point, it is worth mentioning that different authors point out that a security of 80 bits is adequate for low-cost RFID [PPR09, BKL+07], then all our combinations (except the first one) provide sufficient key length.

Figure 4.2 depicts the equivalent-key size regarding the necessary logic GE for the implementation of J3Gen. An implementation with $n = 32$ and $m = 32$ slightly exceeds the 2,000 available GE for security purposes. Using less feedback polynomials reduces the implementation area in exchange for smaller key sizes. This is the case of combining $n = 32$ and $m = 16$, which can be implemented with 1,191 logic GE giving a key of 372 bits. Figure 4.2 shows that, although GE and key bit length grows in parallel, such relation is not uniform, since with 98 GE we can increase 138 bits in key length when moving from

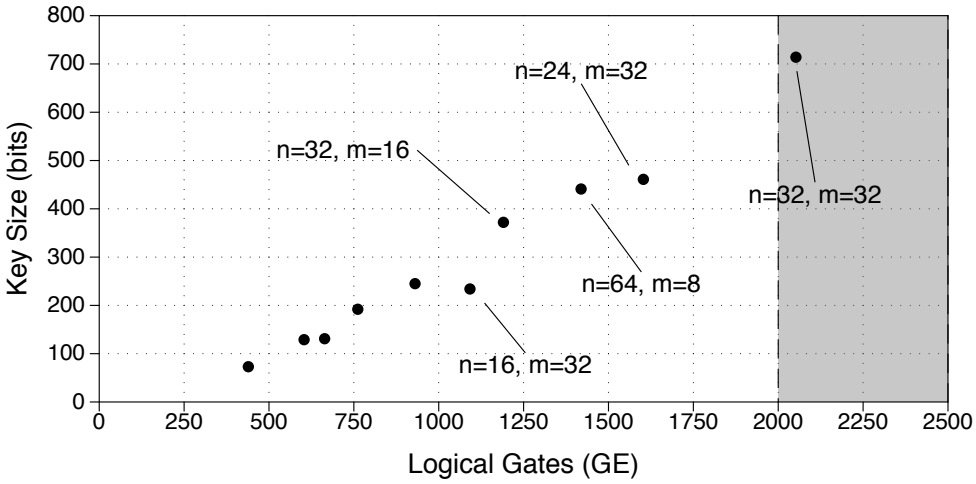


Figure 4.2: Different combinations present suitable trade-offs between security and implementation area

the implementation with $n = 16$ and $m = 32$ to the parameters $n = 32$ and $m = 16$, while 184 GE are required to increase only 20 key bits from $n = 64$ and $m = 8$ to $n = 24$ and $m = 32$. For that reason, we determine the best parameters to choose in terms of GE efficiency with respect to the key length. Taking that measure, we obtain that the best parameter configuration is $n = 32$ and $m = 16$, where the ratio between the key length and GE is maximized (last column of Table 4.5). That is, using an LFSR register with 32 cells ($n = 32$) and 16 feedback polynomials ($m = 16$) implemented on the tag.

Remarks

Regardless the parameters values, the core of the J3Gen generator is an LFSR with multiple polynomials (instead of a single one). The polynomial generator from a simple n-cell LFSR with period 2^n-1 can be determined with only $2n$ bits due to the linearity of this method, by using a system of n equations or the Berlekamp-Massey algorithm [Mas69]. The linearity of J3Gen is avoided with the following technique. The parameter ℓ described in Section 4.1 avoids the generation of more than ℓ consecutive bits from each LFSR polynomial. Depending on the level of desired security, ℓ can be bounded by $1 \leq \ell \leq n - 1$. Moreover, each n-bits pseudorandom sequence is generated by at least two different polynomials.

An attacker aiming to predict the J3Gen output, has to synchronize its output with the beginning of a feedback polynomial generated sequence, obtain the feedback polynomial from $2n$ using the method described above, and repeat the same operation for each m feedback polynomials. Here, the attacker has to face with the uncertainty added by the feedback update rate (ℓ). For example, using the selected parameters $n = 32$ and $m = 16$, if $\ell = 31$ it means there would be up to 4 possible solutions for each system of equations, that is, up to 4 possible feedback polynomials generating that sequence. If $\ell = 25$ then the possible solutions are up to 16,384, for $\ell = 21$ the possible solutions increase to 4,194,304, etc. The extreme case would be $\ell = 1$ where all 67 million primitive feedback polynomials would be equally probable (cf. Table 4.5). Thus, the smaller the polynomial update cycle ℓ, the harder the attack because more bits would be needed to disclose all m feedback polynomials. For instance, a $\ell = 31$ needs about 1,400 bits to obtain all primitive polynomials, $\ell = 25$ needs about 134,000 bits, and $\ell = 21$ needs about 33 million bits. Hence, depending on the desired level of security, the attack will need a longer output sequence of consecutive bits. Equation 4.1 bounds the probability of success of each attack to $2n$ bits, where $p_i(x)$ are the obtained polynomials and $\mathcal{P}_{\text{sel}}$ are the m implemented polynomials on the generator.

$$\frac{1}{2^{n-\ell+1}} \leq P(p_i(x) \in \mathcal{P}_{\text{sel}}) \leq 1 \tag{4.1}$$

If further security is desired, the pool of polynomials can include non primitive polynomials besides primitive polynomials (avoiding those leading to absorbing states), increasing the complexity of the system and decreasing the success odds of a brute force attack.

4.4 EPC Gen2 Suitability

Once the parameters have been fixed based on the hardware constraints and the security requirements discussed in previous sections, we now evaluate the proposed scheme for its implementation, and the restrictions imposed by the EPC standard. We analyze two important parameters of J3Gen: statistical requirements stated by the EPC Gen2 standard for pseudorandom sequence generation, and power consumption.

4.4.1 Statistical Performance

Detailed in the EPC Gen2 standard [EPC07], the requirements for the pseudorandom sequence generation can be summarized as follows:

1. Any single 16-bit sequence s drawn from the generator shall confirm Equation 4.2.

$$\mathrm{P}_{\min} = \frac{0.8}{2^{16}} < \mathrm{Prob}(S) < \mathrm{P}_{\max} = \frac{1.25}{2^{16}} \tag{4.2}$$

2. Among a tag population of up to ten thousand tags, the probability that any two tags simultaneously generate the same 16-bit sequence shall be less than 0.1%.

3. The chance of guessing the next 16-bit sequence generated by a tag shall be less than 0.025% even if all previous outputs are known to an adversary.

To confirm the suitability of the current design of J3Gen for handling the statistical and randomness requirements defined above, different pseudorandom sequences using the parameters defined in Section 4.3 shall be generated. The three EPC Gen2 statistical requirements tests are presented. To accomplish these tests, 30 million 16-bit pseudorandom sequences are generated using an implementation of the J3Gen design. This dataset size is chosen since it is the minimum necessary for a truly generated random sequence to accomplish the proposed requirement [193].

First, the probability of occurrence of any given value shall lie between probabilities defined in Equation 4.2. The results shown in Figure 4.3 confirm that, after analyzing 30 million 16-bit sequences, the probability of occurrence of any given value lies between the defined boundaries. Furthermore, J3Gen achieves similar statistical results with *Random.org* true random sequences [Haa98], based on its frequency properties.

The second property for building an EPC Gen2 compliant PRNG requires that two simultaneous identical sequences must not appear with more than 0.1% probability for a population up to 10,000 tags. To test this property, 10,000 instances of J3Gen, initialized at random, are used to simulate the test scenario. We conducted ten tests, running 1,000 iterations per

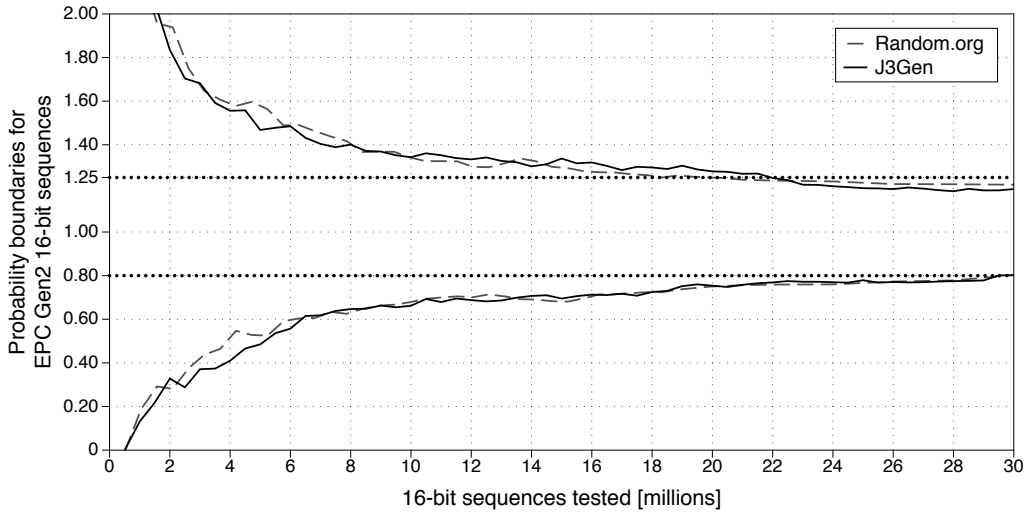


Figure 4.3: EPC Gen2 first randomness property test, achieving similar statistical results than *Random.org* true random sequences

Table 4.6: EPC Gen2 second and third randomness property tests

Test (% rate)	$1st$	$2nd$	$3rd$	$4th$	$5th$	$6th$	$7th$	$8th$	$9th$	$10th$
Same 16-bit sequence	0.0377	0.0383	0.0377	0.0370	0.0375	0.0375	0.0369	0.0375	0.0371	0.0379
Correlation	-0.0085	-0.0093	-0.0044	-0.0014	0.0003	0.0053	0.0073	0.0038	-0.0020	-0.0178

test. The second row of Table 4.6 (labeled as *Same 16-bit sequence*) presents the obtained results. We can observe that all the tests show a simultaneous identical sequence rate one order of magnitude smaller than requested by the standard.

Finally, to statistically confirm the fulfillment of the third property, we computed the degree of dependence of the ongoing bits regarding their predecessors, using the same pseudorandom sequences. Based on the results shown in Table 4.6, we can confirm that the generated sequences are not predictable within the requested correlation of 0.025%.

4.4.2 Power Consumption

The energy used for a (cryptographic) operation depends on the average power and the duration of the computation. For passively powered devices such as RFID tags, the average

power transmitted from the reader to the tag is relatively small (although, in general terms, the reader can supply the power during all the operation time [FW09]). Standard CMOS transistors is the current choice of most digital circuit designs built for low power consumption and robustness [RC08]. Feldhofer *et al.* have estimated the average power budget for cryptographic operations in 4 μW at five meters to the reader [FW09]. Therefore, it is important for the implementation of J3Gen to not surpass the available power budget.

Analytical methods for estimating the CMOS dynamic power dissipation can be adapted to the design of J3Gen [RC08]. Indeed, using custom values adapted to the J3Gen design, the average power consumption is estimated in $\mathrm{P_{avg}}$ = 178 nW (readers can refer to [193] for details).

After defining the design of the digital core of J3Gen based on GEs (cf. Section 4.3), we conduct an electronic circuit simulation of the proposed J3Gen construction, using very-large scale integration (VLSI). The simulation language SPICE [1] is used to simulate the circuit, and the *LTSpice IV* software is used to represent the circuit using logical gates. The resulting simulation also allows us to demonstrate the fundamental concepts of the current construction of J3Gen and confirm its validity as a stand-alone device.

Power dissipation is one of the most important factors in VLSI design and its technology choice. Therefore, accurate simulation of CMOS power dissipation using languages such as SPICE is highly desirable [Kan86]. To precisely evaluate the power consumption of our design it is necessary to provide libraries with parameter models of the specific technology which is simulated. These libraries include a variety of CMOS parameters modeling the transistor behavior and parasitic circuit elements. Using library models, which can be theoretically modeled or hardware measured, the precision of the calculations is improved to effectively simulate the circuit like a real fabricated device. Current UHF RFID products are fabricated on 180 and 130 nm CMOS processes [FW09]. For our simulation we use the *Predictive Technology Model* (PTM) libraries[2] provided by the Nanoscale Integration and Modeling Group from the Arizona State University, which provides CMOS models for 130 nm processes.

[1] Available at `http://bwrc.eecs.berkeley.edu/classes/icbook/spice/`.
[2] Available at `http://ptm.asu.edu`.

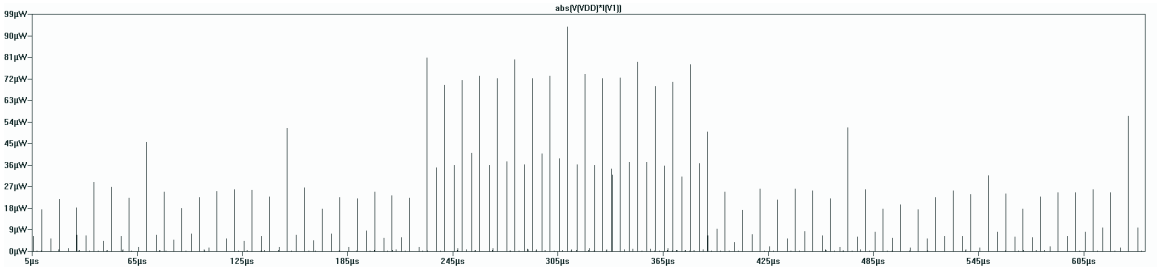


Figure 4.4: LTSpice power consumption simulation. Power dissipation is concentrated around the internal clock cycles.

The analysis targets the average power consumption of J3Gen, in order to evaluate its implementability in a real EPC Gen2 tag. Figure 4.4 depicts the PRNG power consumption during one 16-bit sequence, generated with LTSpice IV and the PTM libraries.

The simulated average power consumption for the 16-bit sequence generation is 156.3 nW, which is consistent with the aforementioned dynamic CMOS power estimation. The simulated power consumption is also under the average power consumption requirements for cryptographic operations in RFID tags proposed by Feldhofer *et al*. [FW09].

4.5 Discussion on Related Work Results

Although RFID is becoming an active research field, very few PRNG designs for lightweight RFID technologies have been disclosed in the related literature. For instance, PRNG implementations in [HBF07, LVHH11, AAKBD12], although efficient in their implementations, cannot be applied to UHF technologies due to the state-of-the-art technologies reported by their authors, or given the power consumption criteria of the EPC Gen2 specification. Manufacturers of existing EPC Gen2 commercial solutions are reluctant to provide their designs [PLHCETR09]. Finally, some of the designs that do appear in the literature, and that claim to be both secure and lightweight enough to fit the EPC Gen2 restrictions, fail to provide convincing proofs of such claims. Some proper examples are [LH07, PLHCETR09, CHTW08, CCY+11].

The design in [LH07] is an optimized variant of the shrinking generator [CKM94], a well studied cryptographic design that combines two clocked linear feedback shift registers (LFSRs) [MOV01]. The output sequence of the first LFSR is used to discard some bits from the output sequence of the second LFSR. However, it is worth pointing out that some techniques presented in [MS95] can be used to attack the scheme. Moreover, there are no evidences of how the proposal in [LH07] controls the irregularities of the generator output rate. This is an important drawback inherent to any shrinking generator scheme, since it can hint at the state of the main LFSR, and so breaking the security of the generator.

The design in [PLHCETR09] is based on a software engineering programming methodology, to automatically derive a set of routines that build a 32-bit PRNG. The hardware implementation of the resulting design is estimated by the authors to 1566 logic gates. An exhaustive statistical analysis confirms that the proposal successfully satisfies the EPC Gen2 statistical expectations. However, no evidences of further achievements other than hardware complexity and statistical behavior are provided. A complete security analysis is missing. Moreover, the inherent peculiarity of their construction methodology obscures potential comparison with other designs in the cryptographic literature.

The last example, presented in [CHTW08, CCY$^+$11], is also a variant of the shrinking generator discussed above, but based on a physical source of randomness that handles the linearity of an underlying LFSR. However, the work presented in [191, 192], and summarized in Chapter 3, proves that the generator scheme can be successfully attacked with very few observations. Indeed, and assuming a 16-bit version of the generator, it was proved that the feedback polynomial can be predicted with a probability higher than 50% by simply capturing 160 bits; and 90% by capturing 464 bits. Therefore, the scheme does not meet any security standard.

The PRNG design presented in this chapter can be applied to current lightweight security proposals in wireless sensor networks, like the one-time-pad encryption scheme by Dolev *et al.* [DGK+11], the proactive threshold cryptosystem for EPC Tags by Garcia-Alfaro *et al.* [181, 177], and the security protocols proposed by Delgado-Mohatar *et al.* [DMFSS11], Liu and Peng [LP06], and Tounsi *et al.* [203]. Since the proposed PRNG combines an LFSR with a non-linear technique (multiple polynomials) which was not used before in security scenarios, no references or previous security analysis can be provided.

4.6 Concluding Summary and Remarks

A pseudorandom number generator (PRNG) design for EPC Gen2 RFID technologies, named J3Gen, has been presented. The generator is based on a linear feedback shift register (LFSR) configured with a multiple-polynomial tap architecture fed by a physical source of randomness. It achieves a reduced computational complexity and low-power consumption as required by the EPC Gen2 standard. It is intended for security, addressing the one-time-pad cipher unpredictability principle. J3Gen is configurable for other purposes and scenarios besides EPC Gen2 RFID technologies through its main parameters n (LFSR size) and m (number of polynomials). The proposed architecture results in a security equivalent-key size of 372 bits, in opposition to the linearity of a single LFSR generator. We have validated the hardware complexity of the design and its suitability with regard to the EPC Gen2 standard. Furthermore, we have considered the randomness requirements through a statistical analysis and the power consumption through an evaluation based on CMOS parameters and SPICE language simulation. Besides EPC Gen2 compatibility presented in this chapter, stronger security can be obtained by adapting the main parameters of J3Gen.

Chapter 5

Proactive Threshold Cryptosystem for EPC Gen2 Tags

We present a second series of EPC Gen2 defense countermeasures, in addition to the PRNG presented in Chapter 4. The countermeasures aim at protecting the distribution of secrets between EPC readers and tags, e.g., passwords required to unlock the operation that protects a tag from unauthorized activation of the writing process. Our proposal can be used as a data-preserving mechanism to exchange the secrets from manufacturers to vendors. It relies on the use of a proactive anonymous threshold cryptosystem. The scheme allows on-board self-renewal of shares with secret preservation between asynchronous parties. The renewal process is based on the randomness provided by the on-board PRNG of a tag. Parts of this chapter have been previously published in [177, 181].

Chapter Outline: Section 5.1 surveys related work. Section 5.2 presents the formalization of our proposal. With regard to the list of threats reported in Chapter 2, three solutions are presented. The first two solutions address the eavesdropping and rogue scanning threats. The third solution mitigates as well tracking threats. Section 5.3 provides some simulation and experimental details. Section 5.4 concludes the chapter.

5.1 Secret Sharing Schemes for EPC Gen2

We have seen in previous chapters that the hardware and power constraints of EPC Gen2 tags makes very challenging the use of solutions based on traditional cryptography. The adoption of low-overhead procedures becomes the main approach to problems where traditional cryptography cannot be accommodated. In Chapter 2, we introduced the use of secret sharing schemes [KGH83] as a promising foundation for the management of keys for the design of authentication protocols and for dealing with privacy issues.

Secret sharing for RFID technologies was initially proposed by Langheinrich and Martin in [LM07b] as an evolution of the minimalist cryptography approach presented by Juels in [Jue04]. Instead of using lists of pseudonyms, the use of secret-sharing schemes is proposed to address authentication in scenarios such as supply chain applications of the retail industry. Indeed, the work presented by Langheinrich and Martin simplifies the lookup process performed on back-end databases for identifying tags, while guaranteeing authentication. Tag Identifiers (TIDs), seen in Langheinrich and Martin's work as the secrets that must be shared between readers and tags, are encoded as a set of shares, and stored in the internal memory of tags. To do so, a Perfect Secret Sharing (PSS) scheme is proposed, in which the size of the shares is equivalent to the size of the secret, based on the (t,n)-threshold secret sharing scheme introduced by Shamir in [Sha79]. The combination of shares at the reader side leads to the reconstruction of original TIDs. To prevent brute-force scanning from unauthorized readers — trying to obtain the complete set of shares — the authors propose a time-limited access that controls the amount of data sent from tags to readers. At the same time, a cache based process ensures that authorized readers can quickly identify tags. In [LM07a], Langheinrich and Martin adapt the (t,n)-threshold secret sharing scheme of Shamir to allow the exchange of secret shares across multiple tags. The idea is to encode the TIDs associated to an item tagged with multiple RFID devices by distributing it as multiple shares stored within the tags. Authentication is achieved by requiring readers to obtain and combine the set of shares. Still, the work in [LM07b, LM07a] cannot protect from information leaks due to the interaction between readers and tags, i.e., there is still the possibility of tracking tagged objects, since tags are always be emitting the same shares.

An alternative use of secret sharing schemes is presented by Juels *et al.* in [JPP08]. The authors propose the use of a dispersion of secrets strategy, rather than the aggregation strategy used by Langheinrich and Marti. In this new approach, a secret used to encrypt Gen2 TIDs is split into multiple shares and distributed among multiple tagged items. Construction and recombination of shares is based on a Ramp Secret Sharing (RSS) scheme, in which the size of each share is considerably smaller than the size of the secret, at the price of leaking out secret information for unqualified sets of shares. To identify the tags, a reader must collect a number of shares above a threshold. At the manufacturer facility, large quantities of items of the same product, initially tagged together with shares of the same secret, guarantee that authorized readers can always reconstruct the secret and, therefore, decrypt the TID of the tagged items. At the consumer side, the items get isolated. Without the space proximity of other items holding the remainder shares of the secret, an unauthorized reader cannot obtain the sufficient number of shares to reconstruct the key that allows identifying the TID. Privacy is, therefore, achieved though the dispersion of secrets and encrypted identifiers. Moreover, the proposed scheme helps to improve the authentication process of tags. Assuming that t shares are necessary for readers to obtain the EPC data assigned to a pallet, a situation where the number of shares obtained by readers is below t leads to conclude that unauthorized tags are present in the pallet. The main limitation of this approach is that a critical privacy threat to consumers, i.e., the tracking threat defined in Chapter 2, is not addressed. It is a requirement stated by most authors, such as Juels and Weis in [JW09]. Privacy-preserving solutions for RFID applications must guarantee both anonymity and untraceability.

In the sequel, we show that it is possible to improve the above limitations. We construct a novel threshold cryptosystem that provides, in addition to authentication and confidentiality, tracking protection. The approach allows share renewal with secret preservation. Renewal of shares does not require synchronization of the share holders. We show that the size of the shares can be reduced to less than 528 bits, as suggested by EPCglobal in [EPC08b]. This way, the approach is compact enough to fit into the inventory responses of low-cost EPC Gen2 tags. We prove that our construction guarantees strong security, and that the reconstruction of the secret does not require the identity of the shareholders.

5.2 Construction of our Cryptosystem Scheme

The construction of our proactive (*t,n*)-threshold cryptosystem relies on the computation of the Moore-Penrose pseudoinverse of a homogeneous system of n linear equations with t unknowns (where $t < n$) over a finite field $\mathbb{Z}_p$,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \quad \cdots \quad + a_{1t}x_t &= 0 \ (mod\ p), \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \quad \cdots \quad + a_{2t}x_t &= 0 \ (mod\ p), \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \quad \cdots \quad + a_{nt}x_t &= 0 \ (mod\ p), \end{aligned}$$

in which t and n are positive integers, and p is a prime number. The vector columns of the coefficient matrix A associated to the system of linear equations are linearly independent, i.e., matrix A has rank t and so the vector columns of A span an inner-product subspace in $\mathbb{Z}_p^{n\times t}$ of dimension t.

The Moore-Penrose pseudoinverse (also called the generalized inverse) of a non-square matrix $A \in \mathbb{Z}_p^{n\times t}$, hereinafter denoted as $A^\dagger$, is the closest representation that A can get to its inverse (since non-square matrices, i.e., $n \neq t$, do not have an inverse). Let us notice that if $rank(A) = t = n$, i.e., A is a full rank square matrix, the Moore-Penrose pseudoinverse of A is certainly equivalent to the inverse matrix A^{-1}, i.e.,

$$A^\dagger = A^{-1} \mid A \in \mathbb{Z}_p^{n\times t} \ \wedge \ \ rank(A) = t = n \tag{5.1}$$

Otherwise, the Moore-Penrose pseudoinverse of a rectangular matrix A exists if and only if the subspaces $Ker\ A$ (null space of matrix A) and $Im\ A$ (range space of matrix A) have trivial intersection with their orthogonals. In the case that $A \in \mathbb{Z}_p^{n\times t}$ has *rank(A)*$= t$, it can be proved that $A^\dagger$ exists and it can be computed as follows:

$$A^\dagger = (A^\perp A)^{-1}A^\perp \in \mathbb{Z}_p^{t\times n} \mid A \in \mathbb{Z}_p^{n\times t} \ \wedge \ rank(A) = t \neq n, \tag{5.2}$$

in which $A^\perp$ denotes the transpose of matrix A. It can also be proved, cf. [Mey00], that if $A \in \mathbb{Z}_p^{n\times t} \mid rank(A) = t$, $A^\dagger$ is the unique solution that satisfies all of the following four

equations defined by Penrose in [Pen55]:

$$\begin{aligned} (A\ A^{\dagger})^{\perp} &= A\ A^{\dagger}, \\ A^{\dagger}\ A\ A^{\dagger} &= A^{\dagger}, \\ (A^{\dagger}\ A)^{\perp} &= A^{\dagger}\ A, \text{ and} \\ A\ A^{\dagger}\ A &= A \end{aligned} \tag{5.3}$$

For our specific construction, we are interested in showing that the resulting matrix $A^{\dagger}$ keeps the orthogonal projection property required in [Pen55]. Indeed, we are interested in showing that the resulting matrix P_A computed as

$$P_A = A\ A^{\dagger} \in \mathbb{Z}_p^{n \times n} \mid A \in \mathbb{Z}_p^{n \times t}\ \wedge\ rank(A) = t \neq n \tag{5.4}$$

is indeed an *orthogonal projector* that satisfies the idempotent property (meaning that $P_A^k = P_A$ for all $k \geq 2$. Certainly, if $P_A = A\ A^{\dagger}$, then $P_A^2 = (A\ A^{\dagger})\ (A\ A^{\dagger})$, i.e., $P_A^2 = (A\ A^{\dagger}\ A)\ A^{\dagger}$). From Equation (5.3), we obtain that $P_A^2 = A\ A^{\dagger}$, i.e., $P_A^2 = P_A$, and so $P_A^k = P_A$ for all $k \geq 2$. Therefore, if $A \in \mathbb{Z}_p^{n \times t}$ and rank(A)= t, then $A\ A^{\dagger} \in \mathbb{Z}_p^{n \times n}$ is an orthogonal projector. Figure 5.1 shows how the orthogonal projector P_A can be used to project a vector v onto the column space of matrix A.

The Moore-Penrose pseudoinverse is a very useful technique used in many engineering fields such as error correction, identification, control design, and structural dynamics. For an over-determined system of linear equations without solution, the Moore-Penrose pseudoinverse finds the least squares solution (i.e., projection of the solution onto the range space of the coefficient matrix of the system). It is also helpful to find the infinite set of solutions in the range space of under-determined set of equations (i.e., fewer constraints than unknowns). The computation of the Moore-Penrose pseudoinverse of a homogenous system of t linear equations with n unknowns (e.g., the computation of the pseudoinverse of matrix $A^{\perp} \in \mathbb{Z}_p^{t \times n}$) is hence a valid alternative for the construction of our proactive threshold secret sharing.

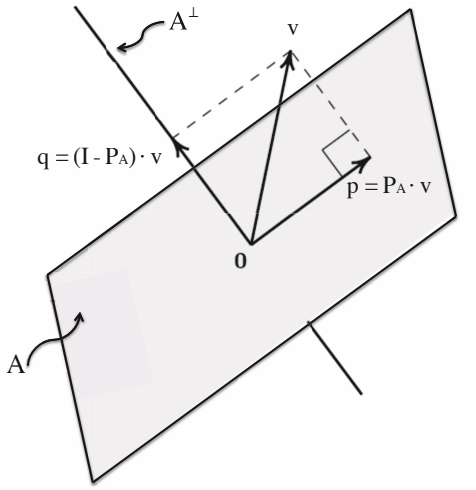


Figure 5.1: Orthogonal Projection of a Vector v onto the Subspace Spanned by the Column Vectors of Matrix A.

5.2.1 Basic (*t,n*)-Threshold Secret Sharing Scheme Based on the Invariance Property of Orthogonal Projectors

Orthogonal projectors have already been used for the construction of threshold secret sharing schemes. In [Bai06a, Bai06b], for example, the invariance property of orthogonal projectors is used for the redundant storage of computer images. Indeed, an asynchronous proactive (*t,n*)-threshold secret sharing scheme can be constructed based on the same observation — meaning that the invariance property of orthogonal projectors can be used to allow shareholders to renew their share without synchronization with other parties and without altering the secret. The key idea of the proposed approach is that the orthogonal projector P_A computed from Equation (5.4) and a random matrix $A \in \mathbb{Z}_p^{n \times t}$ with rank t is always equivalent to the projector P_B obtained from the same equation and any t independent random range images spanned from A.

Before going any further, let us start with a simple example that depicts the basic idea of our approach. It exemplifies the construction of a $(2, 3)$-threshold, non-proactive yet, cryptosystem; and the reconstruction process by three independent reconstruction processes.

Given two matrices $A \in \mathbb{Z}_{31}^{3\times 2}$, $X \in \mathbb{Z}_{31}^{2\times 3}$,

$$A = \begin{bmatrix} 7 & 13 \\ 6 & 29 \\ 13 & 28 \end{bmatrix} \quad X = \begin{bmatrix} 12 & 9 & 13 \\ 26 & 13 & 7 \end{bmatrix}$$

such that A is a random matrix composed of two linearly independent column vectors $a_1, a_2 \in \mathbb{Z}_{31}^{3\times 1}$, i.e., rank($A$) is equal to 2; and X is a random matrix composed of three linearly independent column vectors $x_1, x_2, x_3 \in \mathbb{Z}_{31}^{2\times 1}$, i.e., rank($X$) is equal to 3. Note that we simplify the notation, assuming $A = [a_1, a_2, \ldots, a_t]$, where each a_i is the i-th column vector of matrix A; and $X = [x_1, x_2, \ldots, x_n]$ where each x_i is the i-th column vector of matrix X. Let

$$A' \in \mathbb{Z}_{31}^{3\times 3} = \begin{bmatrix} 19 & 15 & 27 \\ 20 & 28 & 2 \\ 16 & 16 & 24 \end{bmatrix}$$

be the resulting matrix obtained by multiplying matrices A and X. We assume hereafter that the column vectors a'_1, a'_2, and a'_3 in matrix A' are indeed the shares of our cryptosystem; and that $P_A \in \mathbb{Z}_{31}^{3\times 3}$ is the secret of the cryptosystem, in which P_A is the orthogonal projector obtained by applying Equation (5.4) to matrix A.

Let us now assume that a distribution process δ disseminates the shares $a'_1, a'_2, a'_3 \in A'$ to three independent shareholders α, β, and γ. We define the following three column vectors:

$$V_\alpha = \begin{bmatrix} 19 \\ 20 \\ 16 \end{bmatrix}, \quad V_\beta = \begin{bmatrix} 15 \\ 28 \\ 16 \end{bmatrix}, \quad V_\gamma = \begin{bmatrix} 27 \\ 2 \\ 24 \end{bmatrix}$$

as the corresponding shares held respectively by α, β, and γ. We also assume that a reconstruction process ρ_1 requests to shareholders α and β their respective shares (notice that our example describes a $(2, 3)$-threshold cryptosystem and so only two shares suffice to reconstruct the secret). A second reconstruction process ρ_2 requests to shareholders α and γ their

respective shares. Finally, a third reconstruction process ρ_3 requests to shareholders γ and β their shares. Processes ρ_1, ρ_2, and ρ_3 build, independently, three reconstruction matrices B_1, B_2, and B_3 (by simply joining the share vectors they collected from each shareholder):

$$B_1 = \begin{bmatrix} 19 & 15 \\ 20 & 28 \\ 16 & 16 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 19 & 27 \\ 20 & 2 \\ 16 & 24 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 27 & 15 \\ 2 & 28 \\ 24 & 16 \end{bmatrix}$$

We can now observe that the orthogonal projector obtained by applying Equation (5.4) to either B_1, B_2, or B_3 is equivalent to the orthogonal projector obtained by applying Equation (5.4) to matrix A:

$$P_A = \begin{bmatrix} 27 & 13 & 11 \\ 13 & 23 & 21 \\ 11 & 21 & 14 \end{bmatrix}, \; P_{B_1} = P_{B_2} = P_{B_3} = \begin{bmatrix} 27 & 13 & 11 \\ 13 & 23 & 21 \\ 11 & 21 & 14 \end{bmatrix}$$

Therefore, the three processes ρ_1, ρ_2, ρ_3 may successfully reconstruct the secret (i.e., P_A) by performing the same operation described by Equation (5.4). The following theorem establishes the correctness of the approach for the general case.

Theorem 1 *Let $A \in \mathbb{Z}_p^{n \times t}$ be a random matrix of rank t. Let $A' \in \mathbb{Z}_p^{n \times n}$ be the result of multiplying matrix A with a set of n linearly independent column vectors $x_1, x_2, \ldots, x_n \in \mathbb{Z}_p^{t \times 1}$, i.e., $A' = Ax_i \; (mod \; p) \; \forall x_i \in [x_1, x_2, \ldots, x_n]$. Let B be any submatrix from A' with exactly t column vectors. Then, the orthogonal projectors P_A and P_B derived from Equation (5.4) are identical.*

Proof Note that $P_A = A \; A^{\dagger}$ and $P_B = B \; B^{\dagger}$ are the orthogonal projectors obtained by applying Equation (5.4) to both A and B. Since B is any submatrix derived from A' with exactly t column vectors, we can also denote B as the resulting matrix obtained by

multiplying $A \in \mathbb{Z}_p^{n \times t}$ times a given matrix $X \in \mathbb{Z}_p^{t \times t}$. Therefore, $P_B = B\ B^\dagger$ is equal to $P_B = (A\ X)(A\ X)^\dagger$ and so to $P_B = A\ X\ X^\dagger\ A^\dagger$. We know from Equation (5.1) that $X^\dagger = X^{-1}$ when X is a square matrix. Therefore, $P_B = A\ X\ X^{-1}\ A^\dagger$. Since matrix X gets canceled, we obtain that $P_B = A\ A^\dagger$ and so identical to P_A. □

Efficiency or the Proposal

The efficiency of a secret sharing scheme can be evaluated in terms of the information entropy of its shares and secret of the cryptosystem [IY06]. A secret sharing scheme is said to be perfect if it holds that the entropy of the shares is greater than or equal to the entropy of the secret. As a consequence, the size of each share of a perfect secret sharing scheme must be equal or greater than the size of the secret. This is an inconvenient to the hardware limitations of the EPC Gen2 model discussed in Chapter 2. Ramp Secret Sharing (RSS) may considerably improve this efficiency, by allowing a trade-off between security and size of the shares [BM85]. This is the case of the approach presented in this section. Notice that the size of each share $a'_i \in A'$ of our construction is considerably smaller than the size of the secret P_A. More precisely, every share a'_i is a column vector in $\mathbb{Z}_p^{n \times 1}$, while the size of the secret is a matrix in $\mathbb{Z}_p^{n \times n}$, i.e., the size of every share is n times smaller than the secret.

To analyze the robustness of a RSS scheme, in terms of its security, it is necessary to quantify the amount of information about the secret that an intermediate set of shares, smaller than the threshold t, may leak out. This leakage of secret information represents the size of the ramp, in which a small ramp provides stronger security to the scheme than a larger ramp. Yakamoto proposed in [Yam86] to quantify the exposure of secret information from each share by defining a second threshold t', where $0 < t' \leq t$. By definition, a qualified coalition of t shares may reconstruct the secret. An unqualified coalition of $t - t'$ shares cannot reconstruct the secret, but leaks out information about it. Less than t' shares may not reconstruct the secret and does not reveal any information about the secret. The amount of information leaked out from the secret by an unqualified coalition of $t - t'$ shares can be quantified in terms of information entropy. Yakamoto proved in [Yam86] that the

security of a ramp secret sharing scheme is strong enough when the following equivalence applies:

$$H(S|C) \;=\; \frac{t-t'}{t}\, H(S), \tag{5.5}$$

in which $H(S)$ is the information entropy of the secret, and C is an unqualified coalition of $t - t'$ shares. We prove in the sequel that the security of the threshold cryptosystem presented in this section is, according to [Yam86], strong enough.

Theorem 2 *Let $A \in \mathbb{Z}_p^{n\times t}$ be a random matrix of rank t. Let $P_A \in \mathbb{Z}_p^{n\times t}$ be the orthogonal projector obtained by applying Equation (5.4) to matrix A. Let $A' \in \mathbb{Z}_p^{n\times n}$ be the result of multiplying matrix A with a set of n linearly independent column vectors $x_1, x_2, \ldots, x_n \in \mathbb{Z}_p^{t\times 1}$. Then, the basic (t, t', n)-threshold secret sharing scheme constructed from the invariance property of the orthogonal projector P_A, in which matrix P_A is the secret of the cryptosystem, and the column vectors $a'_1, a'_2, \ldots, a'_n \in A'$ are the shares of the cryptosystem, is equivalent to Equation (5.5).*

Proof Since the information provided by matrix A derives P_A by simply applying Equation (5.4), we know that $H(P_A|A) = 0$. Using some information entropy algebra manipulation, we can use this result to decompose $H(P_A)$ as

$$\begin{aligned} H(P_A) &= H(P_A|A) + H(A) - H(A|P_A) \\ &= H(A) - H(A|P_A) \end{aligned} \tag{5.6}$$

Notice that matrix A is any full rank matrix chosen uniformly at random from the sample space in $\mathbb{Z}_p^{n\times t}$. It is proved in [MMO04] that there are exactly $\prod_{i=0}^{t-1}(p^n - p^i)$ random matrices of rank t in $\mathbb{Z}_p^{n\times t}$. Therefore, we can compute $H(A)$ as follows:

$$H(A) = \log_2\left(\prod_{i=0}^{t-1}(p^n - p^i)\right) \tag{5.7}$$

Knowing A and P_A easily leads to $H(A|P_A)$. From Equations (5.3) and (5.4), we have that P_A times A is equivalent to A, meaning that A is an eigenvector matrix of P_A. Hence, the

decomposition of P_A into t eigenvectors $[v_1, v_2, \ldots, v_t] = V \in \mathbb{Z}_p^{n \times t}$ provides information about A. More precisely, matrix A can be obtained from V by using a transformation matrix $W \in \mathbb{Z}_p^{t \times t}$. Since the sample space from which matrix W can be uniformly chosen is exactly of size $\prod_{i=0}^{t-1}(p^t - p^i)$, we have that $H(A|P_A)$ can be obtained as follows:

$$H(A|P_A) = \log_2 \left(\prod_{i=0}^{t-1} (p^t - p^i) \right) \tag{5.8}$$

Using Equations (5.7) and (5.8) we can now compute $H(P_A) = H(A) - H(A|P_A)$:

$$H(P_A) = \log_2 \left(\prod_{i=0}^{t-1} (p^n - p^i) \right) - \log_2 \left(\prod_{i=0}^{t-1} (p^t - p^i) \right) \tag{5.9}$$

Let us now quantify, in terms of entropy, the information about P_A provided by an unqualified coalition A' of t' shares, s.t., $A' = [a'_1, a'_2, \ldots, a'_{t'}]$, and where $0 < t' < t$. Since matrix A' can be seen as a random matrix of rank t' chosen uniformly from the sample space $\prod_{i=0}^{t'-1}(p^n - p^i)$, we have that $H(A')$ can be denoted as follows:

$$H(A') = \log_2 \left(\prod_{i=0}^{t'-1} (p^n - p^i) \right) \tag{5.10}$$

Matrix A' is also an eigenvector matrix of P_A. The decomposition of P_A into t eigenvectors $[v_1, v_2, \ldots, v_t] = V \in \mathbb{Z}_p^{n \times t}$ provides information about A'. Indeed, matrix A' can be obtained from V by using a transformation matrix $W' \in \mathbb{Z}_p^{t \times t'}$. Since the sample space from which matrix W' can be uniformly chosen is exactly of size $\prod_{i=0}^{t'-1}(p^t - p^i)$, we have that $H(A'|P_A)$ can be obtained as follows:

$$H(A'|P_A) = \log_2 \left(\prod_{i=0}^{t'-1} (p^t - p^i) \right) \tag{5.11}$$

We can quantify the amount of information about P_A provided by A', i.e., $H(P_A|A')$, using the results from Equations (5.9), (5.10), and (5.11):

$$\begin{aligned} H(P_A|A') &= H(P_A) - H(A') + H(A'|P_A) \\ &= \log_2\left(\prod_{i=0}^{t-1}(p^n - p^i)\right) - \log_2\left(\prod_{i=0}^{t-1}(p^t - p^i)\right) - \\ &\log_2\left(\prod_{i=0}^{t'-1}(p^n - p^i)\right) + \log_2\left(\prod_{i=0}^{t'-1}(p^t - p^i)\right) \end{aligned} \tag{5.12}$$

When p is a large number, we can simplify the logarithmic expressions in Equations (5.9) and (5.12) to derive $H(P_A)$ and $H(P_A|A')$ as the following approximations:

$$\begin{aligned} H(P_A) &\approx t(n-t)\ \log_2\ p \\ H(P_A|A') &\approx (t-t')(n-t)\ \log_2\ p \end{aligned}$$

We observe that the information entropy of P_A, knowing A', is approximatively $\frac{t-t'}{t}$ times the information entropy of P_A:

$$H(P_A|A') \approx \frac{t-t'}{t}\, H(P_A), \tag{5.13}$$

which, according to Equation (5.5) provided in [Yam86], guarantees that the security of the ramp threshold secret sharing scheme is strong enough. □

Let us conclude this section by determining a value of t, in terms of n, that guarantees that $t-1$ shares cannot reconstruct the secret. Given that the secret is the orthogonal projection P_A derived from the computation of Equation (5.4) and matrix A, and observing again that the projection of A onto the subspace spanned by its range space remains in the same place, i.e., $P_A \cdot A = A$, it is trivial that the projection of any share onto the same subspace does not change either. This effect can be used by a malicious adversary in order to discover P_A by solving n consecutive equations of $(t-1)$ shares. Since, by definition, a (t, n)-threshold secret sharing scheme must prevent any coalition of less than t shares from reconstructing

the secret, the parameter t of our construction shall be bounded in terms of n as follows:

$$\begin{aligned} (t-1)n &< \frac{n(1+n)}{2}, \\ t &< \frac{3+n}{2} \end{aligned} \tag{5.14}$$

From Theorems 1 and 2, we conclude that if $t < \frac{3+n}{2}$, the scheme presented in this section is a strong ramp threshold secret sharing scheme in which exactly t shares may reconstruct the secret, but $t-1$ or fewer shares cannot.

5.2.2 Pseudoproactive Threshold Secret Sharing Scheme Based on the Invariance Property of Orthogonal Projectors and Multiplicative Noise for the Renewal of Shares

We significantly improve in this section the results presented in Section 5.2.1 by showing that the introduction of multiplicative noise in the coefficients of matrix A' does not affect the reconstruction phase. By multiplicative noise we assume independent scalar multiplication of column vector shares $a'_i \in A'$ and scalar random numbers $r_1, \ldots, r_k$ for stretching these vectors. Indeed, we show that the introduction of multiplicative noise into the column vectors of any reconstruction matrix B_i obtained from t column vectors in A' does not affect the results.

The following example shows the key idea of this new version. Assuming again a $(2,3)$-threshold secret sharing scheme based on the orthogonal projectors of matrices $A \in \mathbb{Z}_{31}^{3\times 2}$, $X \in \mathbb{Z}_{31}^{2\times 3}$, and $A' = AX \in \mathbb{Z}_{31}^{3\times 3}$:

$$A = \begin{bmatrix} 7 & 13 \\ 6 & 29 \\ 13 & 28 \end{bmatrix}, \; X = \begin{bmatrix} 12 & 9 & 13 \\ 26 & 13 & 7 \end{bmatrix}, \; A' = \begin{bmatrix} 19 & 15 & 27 \\ 20 & 28 & 2 \\ 16 & 16 & 24 \end{bmatrix}$$

If we generate now three matrices B_1, B_2, and B_3 as combinations of vector columns from $A' = [a'_1, a'_2, a'_3]$ and multiplicative noise, such as $B_1 \in \mathbb{Z}_{31}^{3\times 2} = [5 \cdot a'_1, 17 \cdot a'_2]\ (mod\ 31)$, $B_2 \in \mathbb{Z}_{31}^{3\times 2} = [7 \cdot a'_1, 13 \cdot a'_3]\ (mod\ 31)$, and $B_3 \in \mathbb{Z}_{31}^{3\times 2} = [9 \cdot a'_3, 22 \cdot a'_2]\ (mod\ 31)$:

$$B_1 = \begin{bmatrix} 2 & 7 \\ 7 & 11 \\ 18 & 24 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 9 & 10 \\ 16 & 26 \\ 19 & 2 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 26 & 20 \\ 18 & 27 \\ 30 & 11 \end{bmatrix}$$

we can still observe that the orthogonal projectors obtained by applying Equation (5.4) to either B_1, B_2, or B_3 are certainly equivalent to the orthogonal projector obtained by applying Equation (5.4) to matrix A:

$$P_A = \begin{bmatrix} 27 & 13 & 11 \\ 13 & 23 & 21 \\ 11 & 21 & 14 \end{bmatrix}, \ P_{B_1} = P_{B_2} = P_{B_3} = \begin{bmatrix} 27 & 13 & 11 \\ 13 & 23 & 21 \\ 11 & 21 & 14 \end{bmatrix}$$

Theorem 1 also applies in the general case of this new approach. Notice that if $A \in \mathbb{Z}_p^{n\times t}$ is a random matrix of rank t, and $A' \in \mathbb{Z}_p^{n\times n}$ is the result of multiplying matrix A with n linearly independent column vectors $x_1, x_2, \ldots, x_n \in \mathbb{Z}_p^{t\times 1}$, i.e., $A' = Ax_i\ (mod\ p)\ \forall x_i \in [x_1, x_2, \ldots, x_n]$; then, any submatrix B derived from exactly t column vectors in A', but stretched by multiplicative noise, can still be factorized as $B = A\ X'$, where $X' \in \mathbb{Z}_p^{t\times t}$ is a square random matrix resulting from the set of t linearly independent column vectors in X, but stretched by a specific scaling random number r modulo p. We know from Equation (5.1) that $(X')^\dagger = (X')^{-1}$ when X' is square. Therefore, X' gets canceled during the reconstruction phase, i.e., $P_B = A\ X'\ (X')^{-1}\ A^\dagger$, and we obtain that $P_B = P_A = A\ A^\dagger$.

5.2.3 Proactive Threshold Secret Sharing Scheme Based on the Invariance Property of Orthogonal Projectors and Both Multiplicative and Additive Noise for the Renewal of Shares

As shown in the previous section, every share in the set of shares derived from matrix A' can be independently transformed by adding multiplicative noise. This allows generating numerically different shares while guaranteeing the invariance property of orthogonal projectors to reconstruct the initial secret (i.e., the orthogonal projector P_A derived from matrix A). However, even if the new shares are numerically different, any malicious adversary can successfully observe that the shares are always linearly dependent, since the transformation process is simply stretching the initial share by some scaling random factor r.

We solve this problem by combining both multiplicative and additive noise in the transformation process. The only requirement is to provide to the process in charge of reconstructing the secret a reference used in the transformation process. We assume that this reference is the last column vector in matrix A'. We also assume that the generation process in charge of the construction of A' guarantees that the last column vector is an unordered collection of distinct elements. Then, shareholders are given access to this reference to renew their shares with a linear combination of this reference column. Note that this reference column must be also known a priori by the reconstruction process, but not by any malicious adversary that has access to the renewed shares. Let us illustrate with an example the key idea of this version. Assuming a $(2, 3)$-threshold secret sharing scheme based on matrices $A \in \mathbb{Z}_{31}^{3\times2}$, $X \in \mathbb{Z}_{31}^{2\times3}$, and $A' \in \mathbb{Z}_{31}^{3\times3} = Ax_i \ (mod\ p)\ \forall x_i \in X$:

$$A = \begin{bmatrix} 7 & 13 \\ 6 & 29 \\ 13 & 28 \end{bmatrix},\ X = \begin{bmatrix} 12 & 9 & 13 \\ 26 & 13 & 7 \end{bmatrix},\ A' = \begin{bmatrix} 19 & 15 & \mathbf{27} \\ 20 & 28 & \mathbf{2} \\ 16 & 16 & \mathbf{24} \end{bmatrix}$$

Every shareholder is given column a'_3 and either column a'_1 or column a'_2. Let us assume two shareholders α and β in the system, each holding one of the following two share pairs

V_α and V_β:

$$V_\alpha = \begin{bmatrix} 19 & \mathbf{27} \\ 20 & \mathbf{2} \\ 16 & \mathbf{24} \end{bmatrix}, \quad V_\beta = \begin{bmatrix} 15 & \mathbf{27} \\ 28 & \mathbf{2} \\ 16 & \mathbf{24} \end{bmatrix}$$

Let us assume that a reconstruction process ρ_1 requests to each shareholder their share combination. Both α and β return to ρ_1 a linear transformation from the column vectors in their share pairs. Shareholder α generates a random value $r_\alpha = 15$, transforms $v_{\alpha 1}$ into $v_{\alpha 1} \cdot 15 \ (mod 31)$, and returns $b_\alpha \in \mathbb{Z}_{31}^{3\times 1} = v_{\alpha 1} + v_{\alpha 2}$. Similarly, β generates a random value $r_\beta = 14$, transforms $v_{\beta 1}$ into $v_{\beta 1} \cdot 14 \ (mod 31)$ and returns $b_\beta \in \mathbb{Z}_{31}^{3\times 1} = v_{\beta 1} + v_{\beta 2}$. Two other reconstruction processes ρ_2 and ρ_3 request to each share holder their shares. Shareholders α and β return to ρ_2 and ρ_3 two different linear combinations from the column vectors in their share pairs. Shareholder α returns $b'_\alpha \in \mathbb{Z}_{31}^{3\times 1} = 28 \cdot v_{\alpha 1} + v_{\alpha 2}$ to process ρ_2, and $b''_\alpha \in \mathbb{Z}_{31}^{3\times 1} = 5 \cdot v_{\alpha 1} + v_{\alpha 2}$ to process ρ_3. Shareholder β returns $b'_\beta \in \mathbb{Z}_{31}^{3\times 1} = 19 \cdot v_{\beta 1} + v_{\beta 2}$ to process ρ_2, and $b''_\beta \in \mathbb{Z}_{31}^{3\times 1} = 21 \cdot V_{\beta 1} + V_{\beta 2}$ to process ρ_3. Finally, the process ρ_1 assembles with b_α, b_β the reconstruction matrix $B_1 \in \mathbb{Z}_{31}^{3\times 2}$; the process ρ_2 builds with b'_α, b'_β the reconstruction matrix $B_2 \in \mathbb{Z}_{31}^{3\times 2}$; and the process ρ_3 produces with b''_α, b''_β the reconstruction matrix $B_3 \in \mathbb{Z}_{31}^{3\times 2}$:

$$B_1 = \begin{bmatrix} 2 & 20 \\ 23 & 22 \\ 20 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 9 & 18 \\ 1 & 10 \\ 17 & 2 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 30 & 24 \\ 28 & 15 \\ 20 & 27 \end{bmatrix}$$

We observe that the orthogonal projectors obtained by applying Equation (5.4) to matrices B_1, B_2, and B_3 are identical to the orthogonal projector obtained by applying Equation (5.4) to matrix A:

$$P_A = \begin{bmatrix} 27 & 13 & 11 \\ 13 & 23 & 21 \\ 11 & 21 & 14 \end{bmatrix}, \ P_{B_1} = P_{B_2} = P_{B_3} = \begin{bmatrix} 27 & 13 & 11 \\ 13 & 23 & 21 \\ 11 & 21 & 14 \end{bmatrix}$$

Notice that each matrix $B_i = [b_{i1}, b_{i2}]$, s.t. $i \in \{1 \ldots 3\}$, can be decomposed as follows:

$$\begin{aligned} B_i &= [r_\alpha \cdot a'_1 + a'_3 \ , \ r_\beta \cdot a'_2 + a'_3] \\ &= [r_\alpha \cdot Ax_1 + Ax_3 \ , \ r_\beta \cdot Ax_2 + Ax_3] \\ &= [A\,(r_\alpha \cdot x_1 + x_3) \ , \ A\,(r_\beta \cdot x_2 + x_3)] \\ &= [A\,x'_1 \ , \ A\,x'_2)] = A\,X'_i \end{aligned} \tag{5.15}$$

in which r_α and r_β are the random factors introduced by each shareholder on every interrogation as multiplicative noise; and $X'_i \in \mathbb{Z}_{31}^{2\times 2}$ is a random full rank square matrix derived from A', and so from $A\ X$, plus the multiplicative and additive noise introduced by the shareholders on every interrogation. Since matrix X'_i is a square matrix, the equivalence defined in Equation (5.1) applies, i.e., $X_i^\dagger = X_i^{-1}$. Therefore, the computation of any orthogonal projector P_{Bi} based on Equation (5.4) cancels matrix X'_i and so P_{Bi} is always identical to matrix P_A. This establishes the general case of the new approach based on the proof of Theorem 1.

Let us also observe that if processes ρ_1, ρ_2, and ρ_3 are executed by a qualified entity Ψ_1 with knowledge of reference a'_3, the returned set of column vectors b_α, b'_α, b''_α, and so forth, are clearly linked:

$$b_\alpha = \begin{bmatrix} 2 \\ 23 \\ 20 \end{bmatrix}, b'_\alpha = \begin{bmatrix} 9 \\ 1 \\ 17 \end{bmatrix} = r_1 b_\alpha + \begin{bmatrix} 27 \\ 2 \\ 24 \end{bmatrix}, \ldots$$

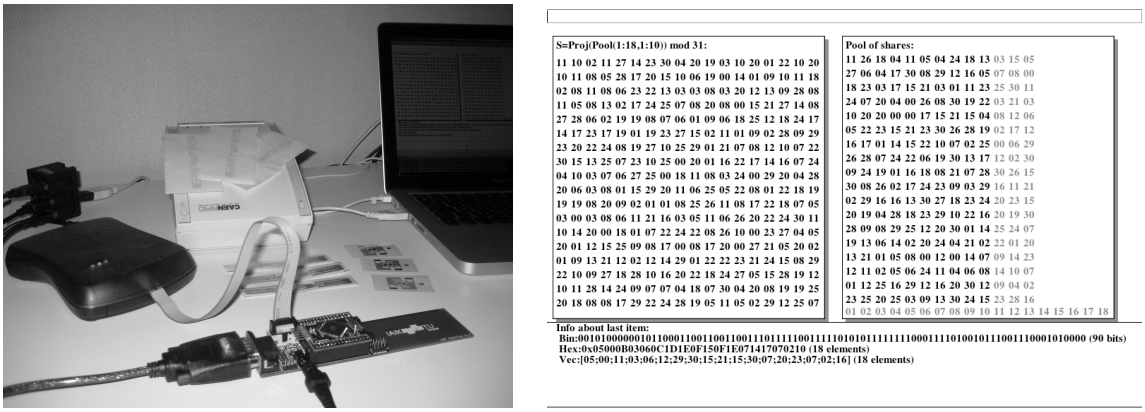
Conversely, if we assume that processes ρ_1, ρ_2, and ρ_3 were executed by a malicious adversary Ψ_2 who is trying to link the shares returned by either α or β, for tracking purposes, but not having access to the column vector reference a'_3, the returned set of column vectors b_α, b'_α, and b''_α, as well as column vectors b_β, b'_β, and b''_β, are viewed as unlinked.

5.3 Simulation and Experimental Results

In Section 5.2, we have seen the formalization of our proposal for the reconstruction of a predistributed secret once a sufficient number of shares are collected. We present in this section the results obtained with an experimental setup that simulates EPC Gen2 adapted shares generation and reconstruction of secrets. Our prototype system allows to experiment the exchange of shares with a regular EPC Gen2 reader and simulated Gen2 tags. The objective of this setup is to demonstrate the practical viability of our proposal.

Figure 5.2 pictures our experimental setup. It is based on the execution of standard EPC Gen2 inventory queries, but enabled with TIDs that are enhanced by our proposed threshold cryptosystem, between a regular EPC Gen2 reader and several Gen2 tag instances simulated by the IAIK UHF demo tag [SIC07]. As we have seen in Chapter 3, the IAIK UHF demo tag is a programmable device intended for developing new extensions to the EPC Gen2 standard. The demo tag consists of an antenna, an RF front-end, a programmable microcontroller, and a firmware library. The antenna captures the energy emitted by the reader and powers up the RF front-end of the tag. The RF front-end demodulates the information encoded in the signal. The encoded data is processed by the programmable microcontroller to compute a response. To compute the response, the programmable microcontroller executes a software implementation of the EPC Gen2 protocol, implemented in the firmware library. The response is then modulated by the RF front-end and backscattered to the reader. More details on our experimental setup are reported in [188].

The share renewal scheme has been implemented in ANSI C using the Crossworks IDE for AVR from Rowley Associates [IDE09]. The theoretical construction detailed in Section 5.2 has been adapted to be executed over the Atmel AVR ATmega128 [Cor09] microcontroller of the IAIK UHF demo tag. The ATmega128 is an 8-bit microcontroller based on the AVR architecture. It has 32 registers of 8-bits that can act as the destinations of standard arithmetic operations. In addition, the ATmega128 microcontroller contains 128KB of flash memory and 4KB of data memory that can be addressed by three independent registers of 16-bits. Since the response of inventory queries is a mandatory operation specified in the EPC Gen2 protocol, an existing response function implemented for the ATmega128



(a) (b)

Figure 5.2: Experimental Setup. In (a), we can see the CAEN A829EU Reader, the AVR JTAG MKII Programmer, the IAIK Graz UHF Demo Tag, and some regular EPC Gen2 tags. In (b), we can see the Java graphical front end that summarizes the process of collecting the secret shares and reconstruction of secrets.

microcontroller is already included in the original firmware stored on the IAIK UHF demo tag. By using the Crossworks IDE, we code and merge the new functionality with the general firmware library to adapt the existing inventory response process to the renewal scheme of shares. The AVR JTAG MKII programmer [Cor09] is used to transfer and to debug the updated firmware merged with the adapted inventory routine. On the reader side, the short-range reader CAEN A829EU [RFI09] emits the inventory queries. The reader is controlled by a back end computer over a USB serial port and a Java application. The Java application is in charge of generating the inventory queries and processing the reconstruction of secrets.

Collection of Shares and Reconstruction Rates

Four different populations of EPC Gen2 tags are simulated and tested. All four simulations are built according to the item-level inventory scenarios reported in [EPC08a, JPP08]. Our objective is to show how our construction can be used in order to maximize the item traceability rate at the upper levels of a supply chain, i.e., at the manufacturer, distributor and retailer sides, while minimizing the traceability rate at the lower levels of a supply chain,

i.e., at the consumer side. The study presented in [EPC08a] shows that items that are initially assembled and tagged together within large collections at the manufacturer side, i.e., top level of the supply chain, get progressively dispersed into very small subsets when they reach the bottom level of the supply chain, i.e., the consumer side. Two appropriate item examples analyzed in [EPC08a] are personal hygiene tools and pharmaceuticals products. According to [EPC08a], personal hygiene tools like, for instance, razor blades, are initially assembled and tagged together at the manufacturer side of the supply chain in large populations of more than 6,000 tagged items. They are later dispersed in the supply chain until being picked up by consumers in groups of less that five items. Similarly, for pharmaceutical items assembled initially in large quantities of more than 7,000 tagged items at the manufacturer side, we should only expect that no more than six items from the initial population can end in possession of a single consumer at the same time. In accordance with these observations, we simulate four different populations of EPC Gen2 tagged items. The tags of each population are initialized with four independent sets of secret sharing schemes constructed according to our proactive threshold secret sharing scheme in GF(2^5–1). More precisely, we initialize the tags of the first population with a $(13, 24)$ scheme that produces tag inventory responses of 120 bits; the tags of the second population with a $(10, 18)$ scheme that produces tag inventory responses of 90 bits; the tags of the third population with a $(7, 12)$ scheme that produces tag inventory responses of 60 bits; and the tags of the fourth population with a $(5, 8)$ scheme that produces tag inventory responses of 40 bits.

Figure 5.3(a)—(d) pictures the average and the 95% confidence intervals of the reconstruction rates obtained with the collection of less than 35 shares from each simulated population. We recall that these simulations take into account the evaluation reported in [EPC08a]. We, therefore, consider the upper bounds of five to thirteen items as the sizes of groups of items picked up by consumers (i.e., lower level of a supply chain). Above these bounds, it is straightforward that authorized readers at the store, warehouse or manufacturer facilities will always reach the necessary threshold to reconstruct the secret and access the appropriate TIDs. For each experimental test of each population, the inventory query emitted by the EPC Gen2 reader is responded by exactly m random tags, where $35 < m < 0$. We recall that the use of a (*t,n*) scheme means that of the n available shares, we need to collect, at

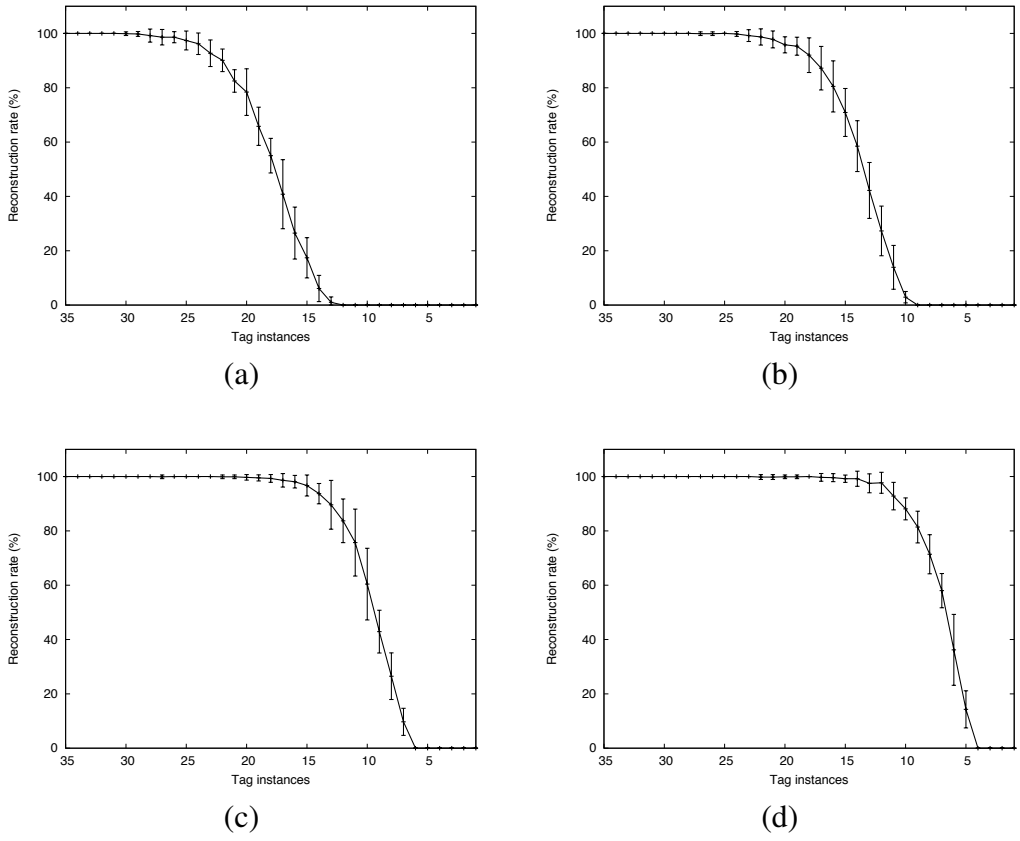


Figure 5.3: Simulation results. (a) First population results with a (13,24) proactive threshold secret sharing scheme of shares; (b) second population results with a (10,18) scheme; (c) third population results with a (7,12) scheme; (d) fourth population results with a (5,8) scheme.

least, t different shares to successfully reconstruct the distributed secret. Each population of tags is initialized by randomly allocating shares from each of threshold scheme. Each interrogation is executed 100 times with random series of simulated tags. Results show the average and 95% confidence intervals computed after each series of interrogations.

The results confirm that while the reconstruction rate minimizes the traceability of tagged items as soon as these items get dispersed in small quantities on the consumer side (amount of tagged items below quantities of less than twenty tags), it guarantees the identification of these items at the upper levels of the supply chain (amount of tagged items above quantities of more than thirty tags). From these results, we may also conclude that the compact size of shares of all four schemes are appropriate enough to fit on the inventory responses suggested on the EPC standard. Note that the resulting inventory responses that are containing the shares (i.e., 120 bits for the first population; 90 bits for the second population; 60 bits for the third population; and 40 bits for the fourth population) do successfully fit within the maximum response size of 528 bits suggested by EPCglobal in [EPC07].

5.4 Concluding Summary and Remarks

We presented a proactive secret sharing procedure to provide consumer privacy and distribution of secrets. Our solution addresses the eavesdropping, rogue scanning, and tracking threats. The main properties of our approach are: (1) low-cost share renewal with secret preservation and without a need of synchronization; (2) compact size of shares; (3) secret sharing construction that guarantees strong security; (4) reconstruction of a secret does not require the identity of the shareholders. We have also presented the implementation of a practical experiment with our proposed cryptosystem in a real EPC Gen2 scenario. By means of a compatible Gen2 reader, and a programmable Gen2 tag [SIC07] implementing our proactive share renewal process, we have shown that a standard EPC Gen2 reader can reconstruct an appropriate predistributed secret dispersed over a set of Gen2 tags. The set of tags communicate the renewed shares to the reader by using a standard inventory response operation, enhanced by our proposed proactive share renewal.

Chapter 6

Formal Verification of Defense Countermeasures

Defenses countermeasures reported in the RFID literature often lack of strong security foundations. Chapter 3 already reported this problem on a hardware-based security primitive for EPC tags. Security RFID protocols seem to be even much more error-prone. Note that a great number of protocols surveyed in Chapter 2 were reported insecure few time after their publication. These cases show the lack of formality during the verification phase of new security techniques for low-cost RFID technologies. In this chapter, we deepen on this problem and illustrate how a sample protocol for the EPC Gen2 technology shall be formally specified with regard to its security requirements. We define a key establishment protocol, and formally verify conformity to some security properties such as authenticity and secrecy. The verification is conducted by using automatic tools from the AVISPA framework [ABB+05]. Parts of this chapter have been previously published in [201, 203].

Chapter Outline: Sections 6.1 to 6.3 present the security assumptions, the security properties, and the protocol. Section 6.4 specifies the protocol and properties using the specification formalism of the AVISPA framework. Section 6.5 presents the obtained results. Section 6.6 surveys related work. Section 6.7 concludes the chapter.

6.1 Security Assumptions

Pseudorandomness Model

The security model of the EPC relies on the communication of one-time sequences used to encrypt sensitive data that must be sent over the insecure RFID channel. The purpose of generating one-time sequences for security is typically that both entities participating on the communications are able to repeat the sequence. However, this is not the case with the EPC technology, where only the tags have access to the sequence generation function. Therefore, the generated sequences cannot be reconstructed at the reader side, and must be sent as clear text over the insecure channel (i.e., tag-to-reader channel) [188]. Most RFID security protocols in the literature use the traditional model, and assume the use of a Pseudorandom Number Generator (PRNG) whose algorithm is known at both sides (i.e., known by readers and tags). This is also the model we assume in this chapter.

PRNG Schemes

A PRNG is seen in this chapter as a pseudorandom bit generator whose output is partitioned into blocks of a given length n. Each block defines a random-looking n-bit number said to be derived from the PRNG. The derived numbers are random-looking bits (statically independent and unbiased binary digits). The PRNG takes a single input called state ($seed$, if it is the initial state) and outputs a next state in addition to the output. All states are assumed to be hidden at all times. There are many nuances of PRNGs used in practice that are often more complicated. For example, some of them are associated to auxiliary inputs such as timestamps or counters which also can be controlled by the adversary. There have been numerous works on constructing PRNGs for symmetric encryption schemes. Common PRNGs consist of two components: (1) a generation function that taking an internal state, generates the next output and then updates the internal state accordingly; and (2) a seed generation function that generates the initial state (and/or key) of the system. While some designers propose a model that combines the internal state and the key of the PRNG

(cf. [BH05] and [BY03]), others aim at separating them. Our assumed PRNG meets the model cited in [DHY02] and considers state and *master* key separately. Indeed, the role played by the key and our concept of internal state are quite different. The key typically has a much longer lifetime and may be repeatedly used for different invocations of the PRNG. The internal state has an ephemeral nature, since it is usually updated during every iteration of the generation algorithm. Our construction concludes a solution to refresh the master key every N interactions as it is shown in Section 6.3.1.

PRNGs can be based on a wide range of cryptographic primitives. The PRNGs that are in relevant use today, are typically based on hash function or block cipher designs. Given the limited computing power of EPC Gen2 tags, we consider in our work PRNGs built from block cipher designs with low-resource hardware constraints. Existing implementations of block cipher based designs for passive RFID tags, such as the 65nm implementation of the Advanced Encryption Standard (AES) in [FWR05], could be adapted for our purpose with a hardware complexity of about 5,000 equivalent logic gates. Other plausible solutions could be adapted from HIGHT [HSH+06], Grain [HJM07], Trivium [DP08], LAMED [PLHCETR09], and J3Gen [193] (cf. Chapter 4), with even lower complexity. Such designs can be adapted to implement pseudorandom permutations (i.e., whose permutations are dependent of a key k) designed to approximate, as closely as possible, a random permutation function, in the sense that if the key k is not known and only input/output examples of executions based on k are captured, then, these should appear like input/output examples of random permutations.

Adversary Model

We assume an active adversary $\mathcal{A}$ who controls the communication channel shared between tags and readers. Therefore, $\mathcal{A}$ can eavesdrop, store, analyze, redirect, and reuse intercepted messages. $\mathcal{A}$ always knows the non-secret data and the functions that each part execute, as well as the inner working of the system (e.g., algorithms and environment associated with the protocol). Additionally, $\mathcal{A}$ can *impersonate* a reader or a tag, and inject new messages by such controlled entities. However, $\mathcal{A}$ cannot modify those messages already sent by a

non-controlled entity, nor can he prevent non-controlled entities from receiving a message already sent. Finally, $\mathcal{A}$ is motivated by any possible scenario leading to the disclosure of secret information used in the protocol. Therefore, we expect from $\mathcal{A}$ the application of the following scenarios:

- *Protocol exposure.* $\mathcal{A}$ can try to find any protocol flaw to decrypt the derived keys relying on its *a priori knowledge of the system*. Therefore, $\mathcal{A}$ can try to find any link between captured messages to correlate two or more protocol outputs. The aim is to obtain information about the derived keys.

- *Master key recovering.* Using the derived keys, $\mathcal{A}$ can try to detect, at least, a valid pair of internal state plus derived key. The aim is to conduct afterwards an exhaustive key search attack to derive the master key.

6.2 Security Properties

The protocol shall provide secrecy of the master and derived keys in addition to assuring that mutual authentication is done between honest participants preventing impersonation attacks. Strong notions of secrecy such as forward and backward secrecy must also be guaranteed even if adversary $\mathcal{A}$ corrupts the whole system by obtaining the session master key and the internal state of the key generation function by external means (e.g., by physically exposing the data of the tags). Therefore, our protocol shall guarantee the security properties defined below.

- **Mutual Authentication:** We define mutual authentication by the agreement of the reader and the tag on the value of a negotiated master key in each session. When this key is also proved to be secret (i.e., nobody except the intended parties knows the key), this strong agreement excludes potential man-in-the-middle and replay attacks in which the adversary could impersonate one of the two parties.

- **Secrecy of the master key:** At any time period, $\mathcal{A}$ cannot recover the master key from the derived keys used in a given session and within the valid period of generation (i.e., before reaching a given threshold N).

- **Forward secrecy:** After the exposure of a given master key, $\mathcal{A}$ cannot compute previous master keys used in the system once the master key is refreshed. In other words, let Km_i be the i^{th} master key negotiated between the tag and the reader, t_i be the last instant of the time interval during which Km_i is in use, and t_C be the instant of the total compromise event of the tag and the reader. Let $\mathcal{K}_t$ be the knowledge of $\mathcal{A}$ at an instant t, such that t comes after t_C. Then $\mathcal{A}$ cannot deduce Km_i from $\mathcal{K}_t$. Equation 6.1 summarized this property.

$$t_i < t_C < t \text{ then } \mathcal{K}_t \nvdash Km_i \tag{6.1}$$

- **Backward secrecy:** After the exposure of a given master key, $\mathcal{A}$ cannot compute future master keys used in the system after the master key is refreshed. In other words, let Km_i be the i^{th} master key negotiated between the tag and the reader, t_i be the first instant of the time interval during which Km_i is in use, and t_C be the instant of the total compromise event of the tag and the reader. Let $\mathcal{K}_t$ be the knowledge of $\mathcal{A}$ at an instant t, such that t comes after t_C. Then $\mathcal{A}$ cannot deduce Km_i from $\mathcal{K}_t$. Equation 6.2 summarized this property.

$$t_C < t_i < t \text{ then } \mathcal{K}_t \nvdash Km_i \tag{6.2}$$

6.3 Proposed Protocol Scheme

Our protocol, hereinafter referred as KEDGEN2, and inspired by previous efforts such as [167], assumes dynamic key establishment based on key transportation techniques [MOV01]. This rationale is used since parties in our system have not the same capabilities. Indeed, RFID readers are expected to have enough computational resources to calculate robust keys. Once computed, the master keys are communicated to the tags, assumed to be resource constrained components.

6.3.1 Key Generation Function

Let $\mathcal{BS} = (\mathcal{K}d, \mathcal{E}n, \mathcal{D}c)$ be the base symmetric encryption scheme of our protocol, specified by its key generation $\mathcal{K}d$, encryption $\mathcal{E}n$ and decryption $\mathcal{D}c$ algorithms. Let $\mathcal{G}en = (\mathcal{S}, \mathcal{G})$ be the PRNG based on a block cipher primitive whose block size is the length of the derived key of the base scheme. $\mathcal{G}en$ consists of two algorithms. The first algorithm $\mathcal{S}$ is a probabilistic algorithm which outputs an initial state St_1 and a master key Km_1. It takes no input values. The second algorithm is an iterative deterministic generation function $\mathcal{G}$, computing in each iteration from three inputs (a master key Km, a state St, a counter cnt) an output Kd and a new state St_i. The counter avoids cases where the same state and key are used. It is a replay defense. For $i \geqslant 1$, the generation algorithm $\mathcal{G}$ takes as input the key Km and the current state St_{i-1} (including the cnt) to generate Kd_i and St_i as: $Kd_i, St_i \leftarrow \mathcal{G}(Km, St_{i-1})$. We associate with our PRNG a re-keyed encryption scheme, which establishes a new key in every new session. The re-keying function can rely on a one way function that is responsible for changing the keys for each session.

An encryption process of the model we propose is pictured in Figure 6.1. The objective is to encrypt every secret message with a new derived key Kd using $\mathcal{E}n$. Thus, the derived keys are used once in each transaction while the master key Km has a longer life time. Notice that our aim is to minimize the advantage (i.e., the likelihood) of the adversary to compromise the security of $\mathcal{G}$ using the data he recovered in each transaction. A potential attack that can take advantage of the weaknesses of the encryption under block ciphers is the birthday attack [BDJR97]. To safely encrypt more data, a practical solution is to enlarge the limited threshold leading to birthday attacks. Thus, we can use the results in [AB00] by introducing a master key re-keying every $N = 2^{n/3}$ encryptions, where n is the block length. The solution increases the encryption threshold from $N = 2^{n/2}$ to $N \approx 2^{2n/3}$. This solution requires less resources than the data dependent re-keying. In addition, it follows the basic protocol design in refreshing the keys every new session. With the re-keying function, our encryption scheme is divided into several stages (in a given session). In stage i, all encryptions are performed using the base scheme with Km_i. An encryption counter is maintained, and when N encryptions are performed, the stage ends and a new stage starts with a new counter cnt and a new master key.

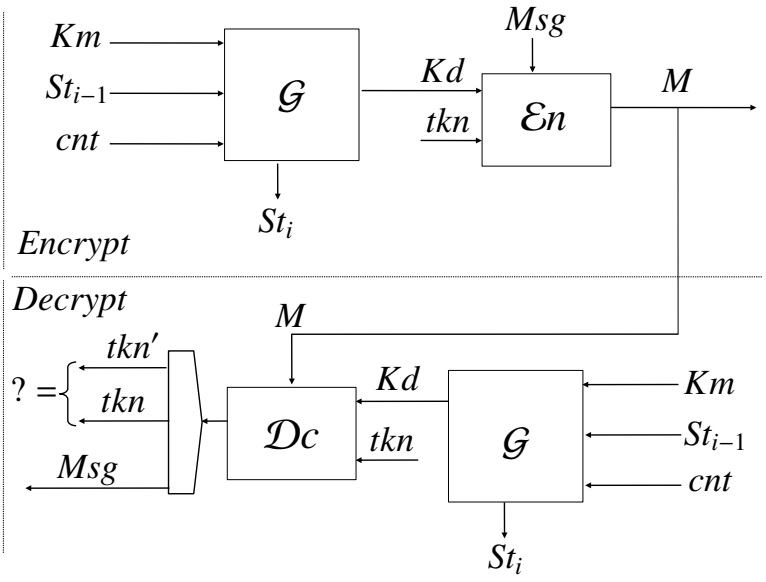


Figure 6.1: Proposed encryption scheme.

In addition to the key generation function, we also assume the following elements:

- *Tag.* A passive constrained device that communicates with readers via a radio interface. The tag is able to give access to its memory only with one reader at a time. It holds the function $\mathcal{G}$ of the generator $\mathcal{G}en$ and is able to derive keys according to its secret master key and state.

- *Reader.* An active entity communicating with the tags and a back-end server. It implements (1) a radio interface to communicate with the tags; and (2) a trusted interface to communicate with the server. It holds the functions $\mathcal{G}$ and $\mathcal{S}$ and is responsible for refreshing master keys when necessary.

- *Back-end server.* A trusted entity that stores in its database all tags and readers information. It is responsible for setting up the initial keys either in the tag or in the reader. It also operates to reset the system when problems arise.

- *Channels.* There are a reader-to-server channel and a reader-to-tag channel. The

readers and the server are related with a high-level security channel. They are assumed to be secured with common security protocols (e.g., SSL/TLS). Reader-to-tag channel is the vulnerable channel that captures our interest.

- *Sessions.* We separate each execution of the protocol by a process named session. For each communication session between a pair of reader and tag, a different master key is established. Session key ensures the independence across sessions to avoid long-term storage of shared keys and to limit the number of ciphertexts available for cryptanalysis.

6.3.2 Protocol Description

We describe now the steps of the protocol. For simplicity, the reader and the server refer to one entity named *sensor*, since (i) readers do not store locally secret information related to tags, and (ii) the linking channels to the server are assumed to be secure.

Each tag and sensor store a generation algorithm $\mathcal{G}$ (cf. Section 6.3.1) with a synchronized process. $\mathcal{G}$ is deterministic. Thus, given Km_i and $St_{i,j-1}$ in the i^{th} session and $(j-1)^{th}$ derivation, $\mathcal{G}$ always outputs the same derived key $Kd_{i,j}$ and State $St_{i,j}$. The function of initialization $\mathcal{S}$ is performed once, i.e., the first time the protocol is executed. It can be re-called if the system has to be reinitialized. The sensor stores in its database all the tag information. For each tag, it records the tag pseudonym (or identifier) TagID, its current state St_i, the master keys (Km_{i-1}, Km_i) to recover the last key in case of desynchronization, a generator counter cnt (cf., Section 6.3.1) and an encryption token tkn. The token tkn can be a counter or a timestamp. In our scheme, we are using a counter since tags are not usually connected to a server that can synchronize their clocks. We assume in all transactions that tkn guarantees that sent messages will be different from the ones sent in the previous transactions. It is meant to ensure the message integrity.

In case of loss in the transmission due to interference or noise, the messages are assumed to be resent with the same counters cnt and tkn. That is, if the reader or the tag do not receive the acknowledgment of the last message, the message can be retransmitted with a bit set to

indicate that it is a duplicate. Hence, the receiver accepts only one validated message.

The master key Km is sent to the destination whenever the key generator $\mathcal{G}$ needs to be refreshed. $\mathcal{G}$ has to be refreshed in the two following situations :

1. When a new session starts. In this case, the new master key becomes the session key.
2. When the key generator counter cnt reaches the threshold N. When this happens, $\mathcal{G}$ is rekeyed with a new master key to extend its lifetime. The new master key is provided to the destination. The new key is used by both parties to replace the actual session key.

In the sequel, we present the different steps and elements of the protocol in detail.

Sessions: A set-up phase is required for initializing the state and the master key Km. In this phase, authentic and secret initial keying material is distributed by a trusted third party over a secure channel.

- *First session.* The tag and the sensor agree on an initial secret composed of an initial master key Km, an initial internal state $seed$ and a shared token tkn. The cnt associated to function $\mathcal{G}$ is also initialized.
- i^{th} *session.* At the beginning of the i^{th} session (i.e., before refreshment), the sensor and the tag share the function $\mathcal{G}$ with the same properties as those used in the $i - 1^{th}$ session meaning that they use Km_{i-1} for generating derived keys. The period of generation is assumed to be still valid for unpredictable derived keys. After the establishment of the master key, the tag and the sensor share: (1) a secret master key Km_i, (2) an internal state St_i associated with a new counter cnt and (3) a token tkn_i which are newly refreshed.

Refreshing $\mathcal{G}$ during the same session: When the generation counter cnt reaches the N value, the sensor sends a query for refreshing $\mathcal{G}$ as follows: the sensor sends to the tag

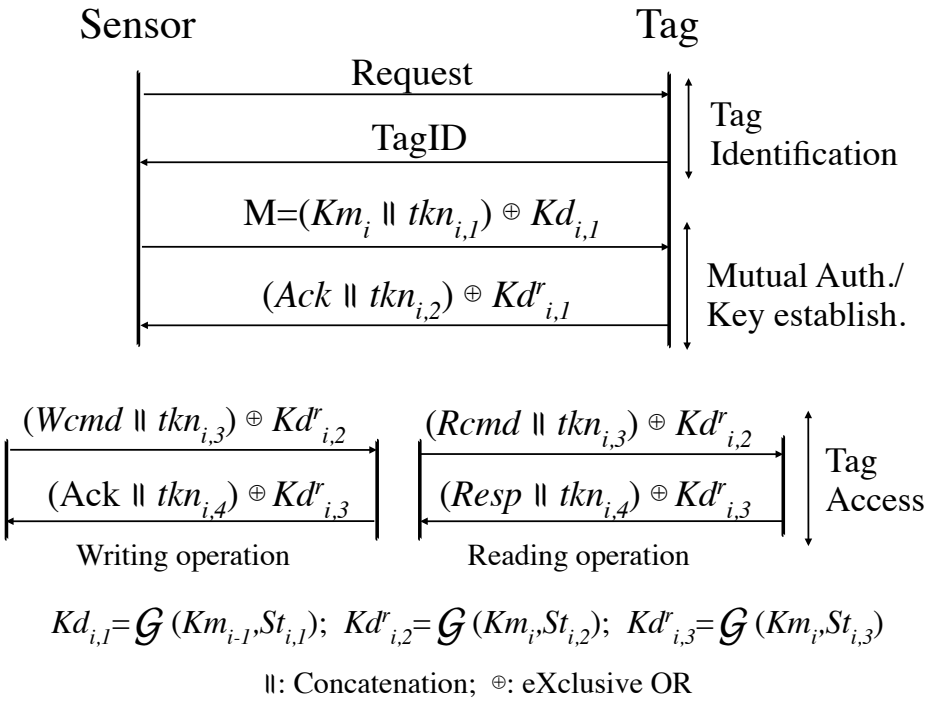


Figure 6.2: Main stages of the protocol.

a special command $Scmd$ XORed with a new derived key $Kd_{i,j}$. The tag acknowledges the refreshment with $Kd_{i,j+1}$. Then, the sensor sends a new master key. The tag verifies the derived key and the token that was used to encrypt the message. In case of equality, it accepts the new master key.

Stages: In each new session, three stages are required. One stage for identification and another for authentication and key establishment. The third stage is a consequence of the successful authentication which means the access to the internal memory of the tag. These stages are shown in Figure 6.2. We briefly summarize them as follows:

- *1. Tag identification stage.* The sensor starts by sending successive requests to the tag until it obtains the TagID. The sensor checks in its database the received TagID.

If there is a match, the sensor associates the TagID with the database identification and the related secret information (i.e., the master key and the state previously negotiated). Both sides must have the same secrets. Otherwise, the next authentication process will fail. The sensor calculates the response including the derived key (by using $\mathcal{G}$ with the valid previous master key Km_{i-1} and St_{i-1}) to prove that it recognizes the tag and XORes the result with the new established master key Km_i. The sensor stores both the current and previous master keys to handle desynchronization.

- *2. Mutual authentication and refreshment stage.* Upon receiving the message M, the tag checks the derived key used for encryption. Then, it calculates a new session key $Kd'_{i,1}$ (i for current session) and decrypts M by applying an XOR operation as follows: $Op = (Km_i, tkn_{i,1}) \oplus Kd_{i,1} \oplus Kd'_{i,1}$. If the decrypted suffix of Op is equal to the predefined token $tkn_{i,1}$, then the tag authenticates the sensor and accepts Km_i as a new master key. It returns an acknowledgment Ack associated with a new derived key $Kd^r_{i,1}$ (r for refreshed key) set from the refreshed values. Otherwise, the tag does not accept the sensor's key and aborts the communication. Upon receiving the value of $Kd^r_{i,1}$, the sensor verifies it and authenticates the tag, in case of validity.

- *3. Tag access stage.* After a successful authentication, the sensor is authorized to access the tag. Thus, it has the ability to execute privileged commands like reading or writing on it. The same process of authentication is used to perform an access operation. Instead of sending the master key, the sensor sends the data to be written on the tag or the tag sends the data required by the sensor encrypted with a fresh derived key:

  - Writing operation: the sensor starts by sending the write command $Wcmd$ concatenated with the token and XORed with a new session key $Kd^r_{i,2}$. The tag verifies the key and accepts the command if the value is valid. Then, it acknowledges the reception with session key $Kd^r_{i,3}$. Otherwise, the tag aborts the communication.

– Reading operation: the sensor starts by sending the read command $Rcmd$ XORed with the new generated key $Kd^r_{i,2}$. If the tag accepts the request, by checking $Kd^r_{i,2}$, then it sends the response $Resp$ XORed with $Kd^r_{i,3}$.

Concurrent Executions: In an EPC environment, a large number of tags can be interrogated at the same time. Thus, it is important for participants to separate concurrent protocol executions. This issue is usually handled by adding a session ID field to the exchanged messages. In our protocol, we assume that each protocol session is associated with an initial internal state, a secret Km, and a token tkn that differentiate all the tags in the system. The sensor can run concurrently many protocol sessions at a time since it maintains a set of tag information. In contrast, the tag cannot run concurrently many protocol sessions at a time, particularly when it needs to update its secrets simultaneously (e.g., the secrets have to be updated before starting a new session). We consider that the tag can respond to several identification requests by sending its TagID. For password-protected requests (e.g., reading and writing access operations), the tag does not respond to simultaneous queries, nor is it able to increment its internal state and token two times simultaneously. Finally, for synchronization reasons, the tag has to run each session for a small period of time, and then switches off automatically — even if the session has not ended.

6.4 Formal Specification and Verification of the Protocol

We use model checking techniques to specify and verify the security of the KEDGEN2 protocol using finite state machine theory. The goal is to dicover logical flaws and attacks against the protocol, w.r.t. the security assumptions provided in Section 6.1. Automated reasoning is highly desirable to avoid errors associated with hand-written proofs [Kre11, SBCC$^+$07]. We start by presenting the verification framework, as well as some preliminary notions about the specification language and the structure of the expected results.

The AVISPA Framework

There is a number of successful protocol verification tools that are supporting algebraic reasoning, e.g., the extended ProVerif [KT11], Maude-NRL Protocol Analyzer [EMM09], On the Fly Model Checker (OFMC) [BMV05] and Constraint-Logic based Attack Searcher (CL-AtSe) [CM09]. They use different models and verification techniques. For example, extended ProVerif is based on tree automata and Horn clauses techniques. MaudeNPA is based on rewriting techniques and backward search of *bad* states. OFMC is based on a state space exploration and CL-AtSe is based on a constraint solving technique. Each tool has some strengths and weaknesses [CLN09]. We use the CL-AtSe protocol analyzer, refereed as the most mature tool using the constraint solving technique [Kre11]. The tool is part of the AVISPA project [ABB$^+$05]. It has recently been extended by the AVANTSSAR [AAA$^+$12] project. We use the latest version of CL-AtSe [CM09] to verify our protocol. The AVISPA platform is a suite of applications commonly used for formal specification and automated validation and verification of cryptographic protocols. It is composed of several modules: (1) a translator called HLPSL2IF [CV05], used to transform from a high-level language specification to a low-level language specification; and (2) a suite of verification tools to analyze the low-level language specifications. From the suite of verification tools, we only use one, the CL-AtSe tool. The reason is that this tool can verify protocols that use algebraic properties of operators like XOR or exponentiation. It also allows running many consecutive, concurrent, sessions. Such features allow us the verification of our protocol.

The *CL-AtSe tool* (from *Constraint-Logic based Attack Searcher*) runs the protocol in all possible ways, and generates families of traces with positive or negative constraints on the adversary knowledge, variable values, and many other aspects. Each protocol run consists in (1) adding new constraints on the current adversary and environment state, (2) reducing these constraints down to a normalized form for which satisfiability is easily decidable, and (3) deciding whether some security property has been violated up to this point. CL-AtSe does not limit the protocol execution in any way except for bounding the maximal number of times the protocol execution can be iterated — in case such a number of iterations is specified by the user. Otherwise, the analysis process could never end. In our case, we

(a)
```
role A(param) played by ag
def= local L
init Init
transition
evt.0 ^ evt.1 = | > act.1
evt.2 = | > act.2
...
evt.n = | > act.n
end role
```

(b)
```
role S (param)
def= const C
composition
R1 (param-1) ^ . . . ^ Rn
(param-n)
end role
```

(c)
```
role environment (param)
def= const C
intruder_knowledge = IN
composition
S1 (param-1) ^ . . . ^ Sn
(param-n)
end role
```

(d)
```
goal
secrecy_of sec_kn
end goal
```

Figure 6.3: HLPSL main elements. (a) Basic role structure. (b) Session role structure. (c) Environment role structure. (d) Secrecy in the goal section.

specify three consecutive series of protocol execution iterations. This number is indeed representative of the different steps of our protocol, i.e., it is sufficient to check the security properties we aim at verifying.

The HLPSL Specification Language

Our protocol and the security assumptions defined in Section 6.1 are specified in the High Level Protocol Specification Language (HLPSL) [CCC+04]. HLPSL is a specification language for formalizing protocols and security goals based on Lamport's Temporal Logic of Actions (TLA) [Lam94]). The language, developed in the context of the AVISPA project [ABB+05], is a role-based language. Roles can be `basic` (e.g., agent roles) describing the action of a legitimate participant during the execution of the protocol; or `composed` (e.g., session and environment roles) describing scenarios of basic roles to model an entire protocol run. The HLPSL language is also used to specify the knowledge and powers of the adversary. Next, we elaborate in detail these aforementioned elements.

Basic roles. Figure 6.3(a) shows how a basic role is generally structured. Each basic role declares its name (`A`), its initial information or parameters (`param`) and the agent playing the role (`ag`). The basic role can declare a set of local variables (`L`). The `init` section

assigns the initial values to the local variables, if required. The `transition` section describes changes of the agent state. It consists of a trigger (e.g., `evt.2`) and an action (e.g., `act.2`) to be performed when the triggered event occurs. The `=|>` operator separates the two phases.

Composed roles. Composed roles combine basic roles, either in parallel or in sequence. HLPSL defines two composed roles: the session role and the environment role. Actions, in composed roles, are not defined in a `transition` section like for the basic roles. Rather, a `composition` section is defined to instantiate other roles `Ri` or `Si`, with sets of parameters `param-i`, that run in parallel (cf. Figures 6.3(b) and 6.3(c)). The session role, referred by `S` in Figure 6.3(b), instantiates in its composition section the basic roles and the different channels relating them while the environment role instantiates in its composition section all the sessions to be run (`Si`). The environment role is called the main role, as it declares the global constants (`C`) and defines the adversary knowledge denoted by `IN` (from INtruder).

Security properties. HLPSL provides an independent section to declare the security properties required, named `goal`. The goal declaration can be done either by using predefined macros of the predefined security properties (secrecy, weak authentication, strong authentication) or by using Linear Temporal Logic formulas [Lam94]. We are interested in the predefined secrecy and strong authentication properties. We use the predefined secrecy property to check whether the secrecy of the key is maintained in a given session and to check (with a slight change of the specification) whether the forward and backward secrecy properties defined in Section 6.2 are guaranteed in the next sessions. We also use the authentication property to validate the goals defined in Section 6.2.

- Secrecy is modeled by means of the goal predicate *secret(Km,sec_km,Sensor,Tag)*, meaning that for the value of term Km is a secret shared only between agents *Sensor* and *Tag*. The secrecy property is violated every time the adversary learns a value that is considered as secret and that he is not allowed to know (i.e., Km).
- Authentication is modeled by means of the goal predicates $witness(A, B, id, T1)$, $request(B, A, id, T1)$ and $wrequest(B, A, id, T1)$. These predicates are used to

check if an instance of a role is right in believing that its peer is present in the current session. This is done by agreeing on a certain value (e.g., $T1$) which is typically fresh. The predicates always appear in pairs and have in common the third parameter. This third parameter id is the identifier of the authentication goal and it is used in the goal section of the HLPSL code. There exists two definitions of authentication: weak and strong authentication.

1. $witness(A, B, id, T1)$ for the strong or weak authentication properties of A by B on $T1$, declares that agent A is witness for information $T1$. This goal will be identified by the constant id in the goal section;
2. $request(B, A, id, T1)$ for a strong authentication property of A by B on $T1$, declares that agent B requests a check of the value $T1$. This goal will be identified by the constant id in the goal section;
3. $wrequest(B, A, id, T1)$ similar to request, but for the weak authentication property. It is used to specify an authentication goal with no replay protection.

Strong authentication is an extension of the weak authentication which precludes replay attacks. We can thus conclude that, if strong authentication is achieved, then $T1$ has not been previously received by B in a given session.

Each property is added to the honest role and to the goal section. It is identified by the `protocol_id` type. Figure 6.4 shows a declaration of a strong authentication property of the *sensor* by the *tag* on the value of $Kd_1 = keygen(KM', succ(KM', InState'))$ declaring that agent *sensor* is witness for the value of Kd_1 and that agent *tag* requests a check of the value Kd_1. This goal is identified by the constant $sensor\_tag\_kd1$ in the `goal` section.

Format of the Output Results

After the verification process, the output describes the results, and under what conditions they have been obtained (e.g., Figure 6.7 shows the verification results of the KEDGEN2

```
role sensor(...)
…
/\ witness (A,Tag',sensor_tag_kd1,
   keygen(KM',succ(KM',InState')))
…
end role

role tag(...)
…
/\request
(Tag,Sensor,sensor_tag_kd1,keygen
(KM.InState))

end role
```

```
role environment(...)
…
tag_sensor_kd1 :protocol_id
…
end role

goal

  authentication_on
sensor_tag_kd1

end goal
```

Figure 6.4: Strong authentication property definition.

protocol). The output format is nearly common to all tools of the AVISPA framework. In the SUMMARY section, it is indicated if the protocol is safe, unsafe, or if the analysis is not conclusive. In a second section titled DETAILS, the output shows conditions under what the protocol is declared safe/unsafe/inconclusive. If a security property of the input specification is violated, then the tools output a warning, some details about the analysis (e.g., whether the considered model is typed or untyped), the property that was violated (e.g., authentication), statistics on the number of explored states, and, finally, an ATTACK TRACE that gives a detailed account of the attack scenario. If no attack was found, then similar information is provided without announcing any violation and attack trace.

HLPSL Specification of our Protocol

The specification of both the protocol and the security goals is described into four HLPSL sections: the sensor, the tag, the environment roles and the goal. Figure 6.5 shows the specification with mutual authentication and secrecy of the master key goals. The generation function $\mathcal{G}$ is specified by two functions $keygen$ and $succ$. The first function generates the derived keys and the second one generates the new state (i.e., $InState$). Figure 6.6 shows the specification of the protocol to handle the forward secrecy property as well. Note that

```
role sensor(A: agent,
  DataBase: (agent.text.message.text) set,
  Snd,Rcv : channel (dy)) played_by A def=

  local
    Tag:agent, InState:message,
    Tkn,KM,NewKM: text, State:nat

  init
    State := 0

  transition

    0.State=0 /\ Rcv(start) =|>
      Snd(A.reqID) /\ State':= 1

    1.State=1 /\ Rcv(Tag')
    /\ in(Tag'.Tkn'.InState'.KM',DataBase)
    =|>
    State':=0 /\ NewKM':= new()
    /\ DataBase':=cons(Tag'.Tkn'.
     succ(KM',InState').KM',
     delete(Tag'.Tkn'.
     InState'.KM',DataBase))
    /\ Snd(xor(NewKM'.Tkn',
     keygen(KM',succ(KM',InState'))))
    /\ State':=2
    /\ witness (A,Tag',sensor_tag_kd1,
    keygen(KM',succ(KM',InState')))
    /\ secret(KM',sec_km1,{A,Tag'})

    2.State=2
    /\ Rcv(keygen(KM',InState'))
    /\ in(Tag'.Tkn'.InState'.KM',DataBase)
    =|>
    request(A,Tag,tag_sensor_kd1r,
    keygen(KM'.InState))

end role
```

```
role tag(Tag,Sensor: agent,
  InState : message,
  KM : text,
  Tkn:text,
  Snd,Rcv: channel(dy)) played_by Tag def=

  local
     State : nat

  init
     State := 0

  transition

     0.State=0 /\ Rcv(Sensor.reqID) =|>
     State':=1 /\ Snd(Tag)
     /\ InState':= succ(KM,InState)

     1.State=1 /\ Rcv(xor((KM'.Tkn),
     keygen(KM.InState))) =|>
     Snd(Tag.keygen(KM'.succ(KM'.InState)))
     /\InState':=succ(KM'.succ(KM'.InState))
     /\ State':=0
     /\request(Tag,Sensor,sensor_tag_kd1,
       keygen(KM.InState))
     /\witness(Tag,Sensor,tag_sensor_kd1r,
       keygen(KM'.succ(KM'.InState)))

end role
```

```
role environment() def=

  const
    sensor,tag1,tag2: agent,
    token1,token2: text,
    instate1,instate2: message,
    km1,km2:text,
    reqID: text,
    succ,keygen: function,
    r2t,t2r: channel (dy),
     a:agent,
    sec_km1,sensor_tag_kd1,
    tag_sensor_kd1r : protocol_id


  intruder_knowledge={reqID,succ,
   keygen,sensor,tag1,tag2,keygen}

  composition
    reader(sensor,
       {
          tag1.token1.instate1.km1,
          tag2.token2.instate2.km2
       },
       r2t,t2r)
     /\ tag(tag1,sensor,instate1,
        km1,token1,t2r,r2t)
     /\ tag(tag2,sensor,instate2,
        km2,token2,t2r,r2t)

end role
```

```
goal
    secrecy_of sec_km1
    authentication_on sensor_tag_kd1
    authentication_on tag_sensor_kd1r
end goal


environment()
```

Figure 6.5: Original HLPSL specification of our proposed protocol

```
role sensor(A: agent,
  DataBase: (agent.text.message.text) set,
  Snd,Rcv : channel (dy)) played_by A def=

   local
     Iter:nat,Tag:agent,
     InState:message,Tkn,KM,NewKM: text,
     State:nat

   init
    State := 0

   transition

    0.State = 0 /\ Rcv(start) =|>
    Snd(A.reqID) /\ State':= 1

    1.State = 1 /\ Rcv(Tag')
    /\ in(Tag'.Tkn'.InState'.KM',DataBase)
    =|>
    State':=0 /\ NewKM':= new()
    /\ Snd(Tag'.NewKM'.Tkn'.
    succ(NewKM',succ(KM',InState')).
    xor(NewKM'.Tkn',
    keygen(KM',succ(KM',InState'))))
    /\ DataBase' := cons(Tag'.Tkn'.
    succ(NewKM',succ(KM',
    succ(KM',InState'))).NewKM',
    delete(Tag'.Tkn'.InState'.KM',
    DataBase))

    /\ secret(KM',sec_km1,{A,Tag'})


end role
```

```
role tag(Tag: agent,
  InState : message, % InState = instate0
  KM : text,Tkn:text,
  Snd,Rcv: channel(dy)) played_by Tag def=

   local
     Reader: agent,
     State : nat

   init
     State := 0

   transition

     0.State=0 /\ Rcv(Reader'.reqID) =|>
     State':=1 /\ Snd(Tag)
     /\ InState' := succ(KM,InState)

     1.State=1 /\ Rcv(xor((KM'.Tkn),
     keygen(KM,InState))) =|>
     State':=0
     /\ Snd(Tag.keygen(KM',
     succ(KM',InState)))
     /\ InState' :=
     succ(KM',succ(KM,InState))


end role
```

```
   role environment() def=

        const
          sensor,tag1,tag2: agent,
          token1,token2: text,
          instate1,instate2: message,
          km1,km2:text,
          reqID: text,
          succ,keygen: function,
          r2t,t2r: channel (dy),
          sec_km1, sec_resp,
          sensor_tag_kd0,
          tag_sensor_kd1 : protocol_id


   intruder_knowledge={reqID,succ,keygen
   ,sensor,tag1,tag2,keygen}

        composition
        reader(sensor,
          {
              tag1.token1.instate1.km1
              ,tag2.token2.instate2.km2
          },
          r2t,t2r)
        /\ tag(tag1,instate1,km1,
        token1,t2r,r2t)
        /\ tag(tag2,instate2,
           km2,token2,t2r,r2t)
   end role
```

```
goal
    secrecy_of sec_km1
end goal


environment()
```

Figure 6.6: Modified HLPSL specification of our protocol, to handle forward secrecy

for the cases of forward and backward secrecy, we slightly change the specification (compared to Figure 6.5) as the AVISPA tool only supports a single execution trace. Thus, we modeled the execution of consecutive iterations in order to show whether leaking a secret during session *i* helps the adversary to obtain secrets, e.g., from session *i-1* for forward secrecy or session *i+1* for backward secrecy. In the sequel, we detail the evaluation results.

6.5 Evaluation of the Results

For each security property defined in Section 6.2 and specified in Section 6.4, we show in this section the results obtained after the evaluation of our protocol specifications under the CL-AtSe tool.

Mutual authentication: Figure 6.7(a) shows the results of the evaluation of the mutual authentication property. To obtain these results, we specify an iteration of the protocol with legitimate roles and give to the adversary the knowledge of the generation functions, roles and standard commands used in the KEDGEN2 protocol communication (cf., Figure 6.5). In the HLPSL language, the authentication property is specified using the $witness/request$ predicates. These predicates are used to check if an instance of a role is right in believing that its peer is present in the current session. We use the HLPSL strong authentication definition to require that a given value is accepted by the sensor in exactly the same session in which it was proposed by the tag. We add these predicates to the tag and sensor transactions to evaluate the authentication of each of the two roles and prevent man-in-the-middle and replay attacks. The tool finds no attack violation of the strong authentication property. This strong property guarantees the resilience to man-in-the-middle and replay attacks in which the adversary could impersonate one of the two parties.

Secrecy of the Master Key: Figure 6.7(a) shows, as well, the results of the secrecy property evaluation. We recall that the secrecy of the master key when shared securely between the tag and the sensor is mathematically maintained since the security threshold

```
INPUT V7-1-ProtocolAuthentifSecrecy.if
SUMMARY NO_ATTACK_FOUND
DETAILS TYPED_MODEL
BACKEND CL-ATSE VERSION
2.5-8_(February_23th_2011)
STATISTICS TIME 44 ms
TESTED 105 transitions
REACHED 34 states
READING 0.04 seconds
ANALYSE 0.00 seconds
```

(a) Authentication and secrecy evaluation

```
INPUT V7-6-forward-orig-chiff.if
SUMMARY ATTACK_FOUND
GOAL:secrecy_of_sec_km1(km1,set_53)
DETAILS TYPED_MODEL
BACKEND CL-ATSE VERSION 2.5-8
_(February_23th_2011)
STATISTICS TIME 28 ms
TESTED 10 transitions
REACHED 6 states
READING 0.01 seconds
ANALYSE 0.02 seconds
```

(b) Forward secrecy evaluation (original specification)

```
INPUT V8-forward-chiff.if
SUMMARY NO_ATTACK_FOUND
GOAL:secrecy_of_sec_km1(km1,set_53)
DETAILS TYPED_MODEL
BACKEND CL-ATSE VERSION 2.5-8
_(February_23th_2011)
STATISTICS TIME 24 ms
TESTED 27 transitions
REACHED 17 states
READING 0.01 seconds
ANALYSE 0.01 seconds
```

(c) Forward secrecy evaluation (modified specification)

```
INPUT V7-6-backward-chiff.if
SUMMARY ATTACK_FOUND
GOAL: secrecy_of_sec_km1(n3(NewKM),set_55)
DETAILS TYPED_MODEL
BACKEND CL-ATSE VERSION
2.5-8_(February_23th_2011)
STATISTICS TIME 928 ms
TESTED 16 transitions
REACHED 12 states
READING 0.05 seconds
ANALYSE 0.88 seconds
```

(d) Backward secrecy

Figure 6.7: Evaluation results.

N of distinguishability is not reached. In other words, the adversary is not able to detect correlations between the outputs of $\mathcal{G}$, named the derived keys. The model checker is used in our evaluation to confirm that the adversary is not able to desynchronize the two participants and replay some messages to reconstruct the master key (and with that, the secret messages encrypted using such a key). To verify the secrecy of the master key, we specify with HLPSL a single instance of the protocol with legitimate roles and give the adversary the knowledge of inner working of the system (cf., Figure 6.5). Secrecy is modeled using the goal predicate $secret(Km, sec\_km, Sensor, Tag)$ standing for the value of term Km is a secret shared only between agents Sensor and Tag. The secrecy property would be violated if the adversary could learn the value Km. Results show that this does not happen.

Forward Secrecy: Figures 6.7(b) and 6.7(c) show the forward secrecy evaluation results. To prove forward secrecy, we consider a setting in which the tag and the sensor try to establish a new master key $NewKm$ using the previous master key Km. Once $NewKm$ has been established, we reveal to the adversary the internal states $NewKm$, $InState$, and Tkn of both the tag and the sensor. Our goal is to prove that this knowledge is not sufficient to enable the adversary to compute the previous Km. We prove first that the original specification of our protocol (cf. Figure 6.5) does not provide forward secrecy. This is shown with the results in Figure 6.7(b). The analysis of the attack trace shows that after establishing and sending the new master key to the tag (i.e., $M = (NewKm||Tkn) \oplus Kd_1$ where $Kd_1 = \mathcal{G}(Km, InState_1)$), the adversary obtains Km in the next generation of $InState$ ($InState_2 = \mathcal{G}(NewKm, InState_1)$) relying on the knowledge of $NewKm$ and Kd_1. The countermeasure is to hide the generation of $InState_2$ by values which are not deduced by the adversary. This way, the adversary cannot obtain the key Km. In fact, by changing $\mathcal{G}(NewKm, InState_1)$ to $\mathcal{G}(NewKm, \mathcal{G}(Km,\ InState_1))$, we use a double generation of the initial state depending on values that cannot be computed by the adversary (i.e., Km). This modification is shown in Figure 6.6. The evaluation results in Figure 6.7(c) show that the modified version satisfies the forward secrecy property even under the hypothesis of a complete compromise in the following sessions.

Backward Secrecy: Figure 6.7(d) shows the results of the backward secrecy property evaluation. We consider two executions of the sensor. One execution in which the tag and the sensor establish a master key Km and where the last secrets of both the tag and the sensor (i.e., $Km, State, Tkn$) are revealed to the adversary. The goal is to verify if this knowledge is sufficient to enable the adversary to compute the new master key $NewKm$ related to this execution. The results show that the protocol is insecure. CL-AtSe finds an attack on the secrecy of the new master key. Indeed, if the adversary follows all the messages sent in the network, it is possible to reconstruct the following master key $NewKm$ because the new derived key used to encrypt the message of refreshment (i.e., 3^{rd} pass in the Figure 6.2) can be computed. The new derived keys are based on the previous secrets that the adversary has already gained, and once obtaining these secrets, the adversary takes all the power of the target tag itself. He can trace it at least during the authentication immediately following the attack. This attack can be avoided by changing the adversary capacities. If the adversary does not eavesdrop on the tag continuously after the time of corruption, i.e., missing the master key establishment transaction, then it will not be possible to predict the next refreshed derived keys. This notion is known to *restricted* backward security through key insulation [SM08, LK06]. This assumption has been assumed in previous efforts, like in [DKS11]. The assumption is realistic since in typical RFID system environments, tags and readers operate only at a short communication range and for a short periods of time.

6.6 Related Work

Work in [VLBDM07, KOK$^+$08, HOM$^+$11, BCdH10, BM11, MM11] are proper examples of RFID protocols verifying forward security and other communication faults at various levels of formality. Some of them define properties such as authentication and secrecy using the computational model, typically in terms of games. In [VLBDM07], Van Le *et al.* define two security protocols to assure authentication and forward secrecy using the universal composability framework. After detecting a synchronization problem related to [VLBDM07], a new series of protocols was proposed by Burmester and Munilla. The

last version in [BM11] ameliorates the protocol. It was verified for the restricted backward secrecy property using the same framework. Hanatani *et al.* propose in [HOM+11], the use of a game-based approach to prove the robustness of an RFID protocol against a man-in-the-middle adversary. They do not propose a new protocol to be applied on constrained tags but a new method to prove the security of the OSK protocol [OSK03] that they combine with a mechanism to synchronize the internal state of the tag and reader. The resulted protocol can be applied on RFID tags supporting hash functions. The security of the protocol is proved using the computational model of CryptoVerif verification tool combined with some handwritten proofs to overcome the limitations of the tool regarding desynchronization and forward privacy verification.

Work in [BCdH10], [KOK+08] and [MM11] use the symbolic model to formally verify the security properties. These are the closest efforts to ours. The advantage of using the symbolic model, as our work does, is its ease to automatically prove the security of cryptographic protocols and to clarify complex protocols with provided definitions of formal languages. Brusò *et al.* propose in [BCdH10] the use of the applied pi calculus language with the ProVerif automated verification tool and apply their proposed techniques to the OSK protocol [OSK03] in order to formally prove the untraceability and forward privacy properties. The proposed technique, which consists in the concept of frame independence between sessions, meets our security goals. However, the proposed verification technique is applied only on one class of protocols that Brusò *et al.* refer as *single step* identification. This technique is applied in protocols with two distinct hash functions. This is only possible on tags computationally strong enough to use such functions. These two criteria make the solution different from our proposal, i.e., our proposal uses more steps for both identification and authentication in the context of Gen2 tags (i.e., without hashing capabilities). In [KOK+08], Kim *et al.* use an automated verification tool called FDR (Failure Divergence Refinement). The work can be compared to ours as it also uses a model checking tool to verify the secrecy and authentication of an RFID protocol. However, the use of a hash based scheme added to a pseudorandom number generator to implement the protocol presents a different solution model. As opposed to our proposal, Kim *et al.* do not consider

	[VLBDM07]	[BM11]	[HOM+11]	[BCdH10]	[KOK+08]	[MM11]	KEDGEN2
Formal model	Computational	Computational	Computational	Symbolic	Symbolic	Symbolic	Symbolic
Framework	Universal Composability	Universal Composability	CryptoVerif	ProVerif	FDR	AVISPA (OFMC)	AVISPA (CL-AtSe)
Forward secrecy	√	√	√	√	–[1]	–	√
Backward secrecy	–	√[2]	–	–	–	–	√[3]
Authentication	√	√	√	√	√	√	√
Cryptographic primitives	Pseudorandom generator. e.g., shrinking generator	PRNG[4]	Three distinct hash functions	Two distinct hash functions	Hash function + PRNG	Hash function	PRNG
Application on highly constrained tags	Possible	Possible	Not possible	Not possible	Not possible	Not possible	Possible

√: Checked by the authors – Not checked by the authors
[1] A different definition of forward secrecy is checked named forward untraceability
[2] Checked under the same adversary used for verifying the forward secrecy
[3] Checked under a weaker adversary used for verifying the forward secrecy
[4] PRNG: Pseudorandom Number Generator

Table 6.1: Comparison between recent related work using various models of formality

in their work strong secrecy notions such as forward and backward secrecy that handle linkability between the sessions. In [MM11], Mahdi and Mohammad propose a new protocol that assures mutual authentication and privacy which they define as anonymity and forward untraceability. The verified property of forward untraceability is different from forward secrecy. Mahdi and Mohammad define the protocol as attacked when the adversary detects twice the same hash result, which means detecting the same tag. Whereas in our case, an attack is shown when an adversary obtains the secret keys of last sessions of communicated keys for a given tag.

To conclude, Table 6.1 shows the different aspects that differentiate our protocol to those aforementioned efforts. We prove the security properties of our protocol using the AVISPA framework, and the CL-AtSe (Constraint-Logic based Attack Searcher) automated verification tool. This allows us to conduct the verification process based on an attack construction methodology, i.e., by attempting to find vulnerabilities using the algebraic properties of our protocol. Differently from all those aforementioned work, the proposal presented in this

chapter is expected to be applied on highly constrained RFID tags based on based on pseudorandomness. We assume that the use of hash functions is beyond the capabilities of our system. KEDGEN2 achieves mutual authentication, and forward and backward secrecy in different conditions. Only the work of Burmester *et al.* in [BM11, VLBDM07] approaches a similar verification problem also for the EPC Gen2 technology. Their work builds upon the universal composability framework, which is based on the computational model.

6.7 Concluding Summary and Remarks

We presented in this chapter the specification and verification of a key establishment and derivation protocol for EPC Gen2 systems. The goal was to illustrate the appropriate way of ensuring the achievement of security requirements when specifying a security protocol for the EPC technology. Our proposed protocol is proven to achieve secure data exchange between tags and readers, based on a key generation model adapted to Gen2 RFID tags. The generated keys are used in the proposed protocol as one time encryption keys. To guarantee the security of the protocol, the generation function has to respond to a number of properties, including the resilience against key recovery and the indistinguishability of the derived keys. We described the steps of our protocol and verified the expected properties under the presence of an active adversary. The current version of the protocol guarantees the properties of mutual authentication and forward secrecy. Backward secrecy is also verified under weaker adversary assumptions (consistent with typical RFID environments).

Perspectives for future work include a more deep evaluation of security primitives for EPC, such as specific on-board hardware add-ons. As we have seen in previous chapters, lightweight primitives are often reported in the literature as insecure (see, for instance, flawed designs in [CHTW08, KYWG10, CCY+11] reported, e.g., in [191, 192] and [Bia11, EBM12]. Major weaknesses are related to the linearity achievement of knowledge with regard to eavesdropped messages (often authentication is linearly linked to the amount of exchanged messages) and desynchronization. An interesting question would be to address this issue using proof construction methodologies. Conversely to the work presented in this

chapter, proof construction methodologies are suitable for proving correctness and completeness rather than finding vulnerabilities in a security protocol. In addition, verification frameworks able to quantify weaknesses of security protocols with regard to dictionary and guessing attacks might also help to enhance the validity of new security primitives. Some existing work in the literature on protocols verification, such as [Del03, GM09, GM11], as well as extensions of the AVISPA framework in the AVANTSSAR [AAA$^+$12] project, seem to head in this direction, and might be interesting to explore.

Chapter 7

Perspectives for Future Work

In this habilitation thesis I have summarized a selection of results I obtained in the last few years while working on Wireless Security. In every chapter, a list of conclusions per contribution has already been presented. In this final chapter, some directions for future work are outlined. These research directions aim at establishing research on security, dependability, and privacy management in Ambient Intelligence (AmI) environments following the methodology and know-how reported in this dissertation.

Towards Ambient Intelligence

Ambient Intelligence (AmI) environments refer to ubiquitous computing systems in which a wide range of heterogeneous devices, with a vast variety of sensing and reactive capabilities, are expected to assist human activities. Sample AmI environments include smart home health services for elderly and people with disabilities, management of next generation power grid services, and surveillance infrastructures for military and civil defense forces. AmI is a fascinating research field which faces numerous challenges. The provision of security, dependability and privacy in AmI environments is a crucial one. AmI

leads to complex security and safety scenarios that not only require the management of traditional communication systems, but also the cooperation among a myriad of autonomous devices that conform the underlying control layers. Moreover, AmI applications allow the collection of vast amounts of data that may contain personal and sensitive characteristics that must be protected. AmI systems require from new solutions to report and anticipate threats and to provide efficient mechanisms to cope with such threats. My proposal aims at establishing a scientific framework to investigate and handle these challenges.

In a nutshell, Ambient Intelligence (AmI) integrates a wide variety of technologies, such as sensors, Radio-Frequency Identification (RFID) labels, human implants, software agents, biometrics, and affective computing. The combination of all these technologies is expected to lead to fully autonomous systems, capable of acting on behalf of human beings. In the AmI paradigm, the concepts of architecture, system or application, as we know them today, evolve to what is known as AmI ecosystems, where complex self-organizing entities coexist and cooperate with each others for achieving operational effectiveness, notably environmental, economical and technical. In general, AmI aims at contributing to personal communications, intelligent highways, wildlife tracking, military battlefield networks, aircraft satellite communication, industrial control systems and outer-space networks. Some more specific examples also include:

- **Smart Home health services for elderly and people with disabilities:** Within the so-called Smart Home scenario, in which AmI devices are deployed to enable a more intelligent surrounding environment at home, expectations lead to monitoring scenarios that must foster the autonomy of, e.g., elderly or cognitively impaired people [NLR11],[204]. In general, AmI technologies deployed at home aim at allowing daily activities of patients to be supervised without the necessity of imposing them to leave their homes or to be permanently supervised by a real presence of nurses or close relatives. Some systems based on the use of wireless sensor nodes and RFID (Radio-Frequency IDentification) systems drastically reduce the deployment costs of traditional surveillance cameras [PSJ+05], as well as reducing the response time for decision-making situations and granting users access to the necessary knowledge when interacting with such an increased level of intelligence [CRS+04].

- **Management of next generation power grid services:** AmI is the key component for dealing and complementing the next generation power grid, known as smart grid [VJS10]. The smart grid scenario is an "*upgraded electricity network to which two-way digital communication between supplier and consumer, intelligent metering and monitoring systems have been added*" [Com11]. The European Union is expecting from its member states an 80% adoption of AmI enhanced energy meters (the so-called *smart meters*) by 2020. In this context, the concept of AmI ecosystems will definitively emerge as intelligent electronic autonomous networks, vertically integrated within power suppliers; and lead to the adoption of sensing infrastructures, information management, storage, transmission and consumption devices present in urban spaces, buildings and homes.

- **Surveillance infrastructures for military and civil defense forces:** The adoption of AmI-based monitoring and surveillance technologies is starting to be deployed by the military and state defense departments, to assist the protection of critical or strategical infrastructures. Examples include intruder detection and barrier coverage in geographic delimiters, state borders or SCADA systems [KLA05]. In such scenarios, mobile autonomous sensors are getting control and look over crossing paths within specified regions so as to prevent threats or illegal activity. The construction of smart, virtual, fences for replacing traditional barriers is a reality along the U.S.-Mexico border [Hu08], in order to monitor large and unprotected areas. Similar solutions are likely to be deployed on the occasion of civil events of great magnitude, such as the World Cup and the Olympic Games, in which world-class personalities and athletes require rigorous protection [TT08, VALMGC$^+$10].

Research Goals and Expected Contributions

The aim is to establish a consolidated scientific framework that benefits from the expertise acquired during my last ten years of research work, in order to handle crucial challenges introduced by the AmI paradigm. AmI environments create significant research problems,

especially with respect to the management of information. Indeed, AmI requires from sophisticated structures and models to represent the huge amounts of events and interactions that will be generated, stored, exchanged and consumed. In this regard, evaluation and validation of AmI-based systems is increasingly difficult. The provision of security, dependability and privacy management in AmI is even harder [DPR07]. With this in mind, I elaborate next some sample research topics in which my scientific background can help successfully to contribute.

Provision of Security and Dependability in AmI. Security and dependability solutions for AmI must handle the autonomous and openness nature of this computing paradigm. This implies that the entities of the AmI ecosystem must be aware of the hostile actions that the environment can perpetrate against them, in order to preempt such actions, or mitigate them. In this regard, *our main goal will be the definition and implementation of a complete framework that formally handles the concept of security and self-protection in AmI*. It shall be interesting to explore trade-offs and the interaction of different protection models allowing the representation of the properties that the different entities are obliged to guarantee, and study the necessary mechanism to enforce the associated degrees of security and protection that the whole system must provide. *The concept of autonomous reaction shall be handled as well.* Autonomous reaction must be seen as the series of system updates required to handle threats and faults without the necessity of human intervention [BSF08]. This should happen not only whenever attacks and malfunctions target the system, but also when system vulnerabilities are detected, prior their exploitation.

Privacy Management of AmI Information Flows. Beyond traditional security and dependability properties, AmI is considered to be a very invasive technology in terms of privacy. Tools and resources related to ubiquitous computing models and computer-assisted services raise a huge rate of data flows with personal and sensitive characteristics that must be protected [FVPW07]. The goal of *privacy management* is to provide efficient and effective solutions for releasing such data, while providing scientific guarantees that the identities and other sensitive information of the individuals, who are the subjects of the data,

are protected. While *data security* ensures that a given entity has the authority to receive a given piece of information, privacy addresses disclosures based on inferences that can be drawn from released data. *Our main action in the related AmI area is to propose new paradigms for privacy management by means of exploration and development of methods and models for advancing the field of privacy preservation*. This shall include, among others, advancing in terms of privacy metrics and prioritization of requirements. Right now, very few AmI-based metrics required to guarantee a flexible degree of privacy preservation have been studied in the literature. *We propose to start the work by fostering new knowledge on contextual metrics, measurements of utility and risk functions, and metrics upon unstructured data such as e-health reports, energy audits, and monitoring alarms*. With respect to requirements prioritization, the action that we propose is in terms of aggregation operators and multi-criteria techniques. The former relies on retrieving data based only on aggregate statistics, e.g., based on Bayesian theory, and providing subsets of entities represented in a given knowledge database. The latter relies on decision-making techniques to handle disclosures [Tri00]. Given the cooperative nature of the problem domain, our work will be driven by handling phenomena such as rank reversals (i.e., adding or deleting decision alternatives may cause a reversal in the ranking of the old ones) and coherence measures for preference divergences (i.e., to avoid conflicts or redundant decision making processes). The use of probabilistic programming [Pré03] is envisioned to define models, compute metrics and solve conflicts.

Conciliation of Security and Privacy in AmI. A final goal would be to conduct research towards establishing the foundations of *Privacy by Design* in terms of AmI technologies. This is directly related with the enforcement of the *privacy by design* principle [Cav09], which relies on the philosophy of proactively embedding privacy management aspects into the technologies themselves, while guaranteeing the expected degree of security and dependability. Broadly speaking, it tackles a series of general principles that must be considered in the phases of the development process, from the analysis to the final implementation of the technology itself, as well as finding the necessary tradeoff to make possible to enhance both security and respect for privacy. This way, the design of privacy management technologies shall not only target particular privacy aspects at some specific stages of the

development process. They should be integrated as if it was a unique, single procedure. In light of these issues, *we plan to study the assimilation of the principle into the distributed monitoring mechanism expected from the sensing devices of those AmI scenarios identified in this chapter, and contribute in terms of automatic verification of privacy and data protection requirements*.

Concluding Summary and Remarks

The provision of security and dependability in AmI (Ambient Intelligence) ecosystems, as well as security and privacy conciliation in AmI, is a crucial open problem. Perspectives outlined in this chapter are centered on modeling and analysis techniques. The research agenda aims at establishing a science of security and privacy techniques for AmI environments, driven by foundational approaches and methodologies that have proven valid to handle traditional IT problems. The experience and knowledge in related areas reported in this habilitation thesis, from standards and organizational security to technical implementations and practical skills, acquired while working with respected researchers in the field, guarantees the fulfillment of the proposed agenda. My experience supervising and guiding graduate and doctoral students, as well as organizing and leading research projects and scientific animation activities, shall also guarantee guidance for fostering further investigations programs.

Bibliography

[AAA+12] A. Armando, W. Arsac, T. Avanesov, M. Barletta, A. Calvi, A. Cappai, R. Carbone, Y. Chevalier, L. Compagna, J. Cuellar, G. Erzse, S. Frau, M. Minea, S. Mödersheim, D. Oheimb, G. Pellegrino, S. Ponta, M. Rocchetto, M. Rusinowitch, Mohammad Torabi D., M. Turuani, and L. Vigano. The AVANTSSAR platform for the automated validation of trust and security of service-oriented architectures. In *18th international conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2012)*, pages 267–282. Springer, 2012.

[AAKBD12] J.A. Aguilar-Angulo, E. Kussener, H. Barthelemy, and B. Duval. A new oscillator-based random number generator. In *Faible Tension Faible Consommation (FTFC), 2012 IEEE*, pages 1–4, 2012.

[AB00] M. Abdalla and M. Bellare. Increasing the lifetime of a key: a comparative analysis of the security of re-keying techniques. *Advances in Cryptology (ASIACRYPT'00)*, 1976:546–559, 2000.

[ABB+05] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. Drielsma, P. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, O. Von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Vigano, and L. Vigneron. The AVISPA tool for the automated validation of internet security protocols and applications. In *17th International Conference on Computer Aided Verification (CAV'05)*, pages 135–165. Springer, 2005.

[AEKBB+03] A. Abou El Kalam, R. Baida, P. Balbiani, S. Benferhat, F. Cuppens,

Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin. Organization based access control. In *IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003)*, pages 120–131. IEEE, 2003.

[AO05a] G. Avoine and P. Oechslin. RFID traceability: A multilayer problem. In *Financial Cryptography (FC'05)*, volume 3570 of *Lecture Notes in Computer Science*, pages 125–140, Roseau, The Commonwealth Of Dominica, February–March 2005. IFCA, Springer-Verlag.

[AO05b] G. Avoine and P. Oechslin. A scalable and provably secure hash based RFID protocol. In *International Workshop on Pervasive Computing and Communication Security (PerSec 2005)*, pages 110–114, Kauai Island, Hawaii, USA, March 2005. IEEE, IEEE Computer Society Press.

[APF+07] M. Aigner, T. Plos, M. Feldhofer, C. Tutsch, A. Ruhanen, Y. Na, S. Coluccini, and M. Tavilampi. BRIDGE — Building Radio frequency IDentification for the Global Environment. Report on first part of the security WP: Tag security (D4.2.1). Technical report, Stiftung Secure Information and Communication Technologies, 2007.

[Bai06a] L. Bai. A reliable (k, n) image secret sharing scheme. In *Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium on*, pages 31–36. IEEE, 2006.

[Bai06b] L. Bai. A strong ramp secret sharing scheme using matrix projection. In *2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, pages 652–656. IEEE Computer Society, 2006.

[Bak07] R.J. Baker. *CMOS: Circuit design, layout, and simulation*. Wiley-IEEE Press, 2007.

[Bar90] P.H. Bardell. Analysis of cellular automata used as pseudorandom patterngenerators. In *International Test Conference*, pages 762–768, 1990.

[BCdH10] M. Brusò, K. Chatzikokolakis, and J. den Hartog. Formal Verification of Privacy for RFID Systems. In *CSF*, pages 75–88. IEEE, 2010.

[BCI07] J. Bringer, H. Chabanne, and T. Icart. Improved Privacy of the Tree-Based Hash protocols using Physically Unclonable Function. Cryptology ePrint Archive, Report 2007/294, 2007.

[BDJR97] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation. In *38th Annual Symposium on Foundations of Computer Science (FOCS'97)*, pages 394–403, 1997.

[BGK+06] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. An elliptic curve processor suitable for RFID-tags. Cryptology ePrint Archive, Report 2006/227, 2006.

[BH05] B. Barak and S. Halevi. A model and architecture for pseudo-random generation with applications to/dev/random. In *12th ACM conference on Computer and communications security (CCD 2005)*, pages 203–212. ACM, 2005.

[Bia11] G. Bianchi. Revisiting an RFID Identification-Free Batch Authentication Approach. *IEEE Communications Letters*, 15(6):632–634, 2011.

[BKL+07] A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 4727 of *LNCS*, pages 450–466. Springer Berlin / Heidelberg, 2007.

[BM85] G.-R. Blakley and C. Meadows. Security of ramp schemes. In *Advances in Cryptology*, pages 242–268. Springer, 1985.

[BM11] M. Burmester and J. Munilla. Lightweight RFID authentication with forward and backward security. *ACM Transactions on Information and System Security*, 14(1):11, 2011.

[BMP97] S. R. Blackburn, S. Murphy, and K. G. Paterson. Comments on 'theory and applications of cellular automata in cryptography'. *IEEE Trans. Softw. Eng.*, 23(9):637–638, 1997.

[BMV05] D. Basin, S. Mödersheim, and L. Vigano. OFMC: A symbolic model checker for security protocols. *International Journal of Information Security*, 4:181–208, 2005.

[BR07] L. Bolotnyy and G. Robins. Physically unclonable function-based security and privacy in RFID systems. In *International Conference on Pervasive Computing and Communications (PerCom 2007)*, pages 211–220, New York, USA, March 2007. IEEE, IEEE Computer Society Press.

[BSF08] R. Badonnel, R. State, and O. Festor. Self-configurable fault monitoring in ad-hoc networks. *Ad Hoc Networks*, 6(3):458–473, 2008.

[BY03] M. Bellare and B. Yee. Forward-security in private-key cryptography. *Topics in Cryptology (CT-RSA'03)*, 2612:1–18, 2003.

[Cav09] A. Cavoukian. Privacy by design, 2009.

[CCC+04] Y. Chevalier, L. Compagna, J. Cuellar, P.H. Drielsma, J. Mantovani, S. Moedersheim, and L. Vigneron. A high level protocol specification language for industrial security-sensitive protocols. In *Proceedings of Workshop on Specification and Automated Processing of Security Requirements*, volume 180 of *(SAPS'04)*, 2004.

[CCY+11] W. Chen, W. Che, N. Yan, X. Tan, and H. Min. Ultra-Low Power Truly Random Number Generator for RFID Tag. *Wireless Personal Communications*, 59(1):85–94, 2011.

[Che86] C.-L. Chen. Linear dependencies in linear feedback shift registers. *Computers, IEEE Transactions on*, 100(12):1086–1088, 1986.

[CHTW08] W. Che, H.Deng, X. Tan, and J. Wang. *Networked RFID Systems and Lightweight Cryptography, Chapter 16*, chapter A Random Number Generator for Application in RFID Tags, pages 279–287. Springer, November 2008.

[CKM94] D. Coppersmith, H. Krawczyk, and Y. Mansour. The shrinking generator. In *Advances in Cryptology - Crypto'93*, pages 22–39. Springer, 1994.

[CLL05] E. Choi, S. Lee, and D. Lee. Efficient RFID authentication protocol for ubiquitous computing environment. In *International Workshop on Security in Ubiquitous Computing Systems – secubiq 2005*, volume 3823 of *Lecture Notes in Computer Science*, pages 945–954, Nagasaki, Japan, December 2005. Springer.

[CLN09] C. Cremers, P. Lafourcade, and P. Nadeau. Comparing State Spaces in Automatic Security Protocol Analysis. In *Formal to Practical Security*, pages 70–94, 2009.

[CM09] A. Charu and T. Mathieu. Validating Integrity for the Ephemerizer's Protocol with CL-Atse. In *Formal to Practical Security*, 2009.

[Com11] European Commission. Smart Grids: from Innovation to Deployment — Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions, COM(2011) 202, 2011.

[Cor09] Atmel Corporation. The AVR ATmega128 processors, 2009.

[CR08] P. Cole and D. Ranasinghe, editors. *Networked RFID Systems and Lightweight Cryptography — Raising Barriers to Product Counterfeiting*. Springer Berlin Heidelberg, first edition, 2008.

[CRS+04] S. Consolvo, P. Roessler, B. Shelton, A. LaMarca, B. Schilit, and S. Bly. Technology for care networks of elders. *IEEE Transactions on Pervasive Computing*, 3(2):22–29, 2004.

[CV05] Y. Chevalier and L. Vigneron. Rule-based Programs Describing Internet Security Protocols. *Electronic Notes in Theoretical Computer Science*, 124:113–132, 2005.

[CvLB06] C. Chatmon, T. van Le, and M. Burmester. Secure anonymous RFID authentication protocols. Technical Report TR-060112, Florida State University, Department of Computer Science, Tallahassee, Florida, USA, 2006.

[Del03] S. Delaune. Intruder Deduction Problem in Presence of Guessing Attacks. In Michaël Rusinowitch, editor, *Proceedings of the Workshop on Security Protocols Verification (SPV'03)*, pages 26–30, Marseilles, France, September 2003.

[DGK+11] S. Dolev, N. Gilboa, M. Kopeetsky, G. Persiano, and P.G. Spirakis. Information security for sensors by overwhelming random sequences and permutations. *Ad Hoc Networks*, 2011. (In press).

[DHY02] A. Desai, A. Hevia, and Y. Yin. A practice-oriented treatment of pseudorandom number generators. In *Advances in Cryptology (EUROCRYPT'02)*, pages 368–383. Springer, 2002.

[DKS11] S. Dolev, M. Kopeetsky, and A. Shamir. Rfid authentication efficient proactive information security within computational security. *Theory of Computing Systems*, 48(1):132–149, 2011.

[DMFSS11] O. Delgado-Mohatar, A. Fúster-Sabater, and J. Sierra. A light-weight authentication scheme for wireless sensor networks. *Ad Hoc Networks,*, 9(5):727–735, 2011.

[DP08] C. De Canniere and B. Preneel. Trivium specifications. Technical report, Ecrypt 2008, 2008. [Online, last access Apr. 2013] Available at `http://www.ecrypt.eu.org/stream/triviumpf.html`.

[DPR07] Y. Deswarte, D. Powell, and Y. Roudier. Sécurité, protection de la vie privée et disponibilité, 2007.

[DR02] J. Daemen and V. Rijmen. *The Design of Rijndael: AES–the Advanced Encryption Standard*. Springer, 2002.

[EBM12] K. Elkhiyaoui, E.-O. Blass, and R. Molva. ROTIV: RFID ownership transfer with issuer verification. In *RFID. Security and Privacy*, pages 163–182. Springer, 2012.

[EMM09] S. Escobar, C. Meadows, and J. Meseguer. Maude-NPA: Cryptographic Protocol Analysis Modulo Equational Properties. In *Foundations of Security Analysis and Design (FOSAD 2007)*, pages 1–50. Springer, 2009.

[EPC07] EPCglobal. The EPCglobal architecture framework, 2007.

[EPC08a] EPCglobal. EPC Item Level Tagging Joint Requirements Group, 2008.

[EPC08b] EPCglobal. EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860-960 MHz. Technical report, Version 1.2.0, 2008.

[ETS03] ETSI. Methods and protocols for security; part 1: Threat analysis. etsi ts 102 165-1 v4.1.1, 2003.

[FDW04] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004)*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370, Boston, Massachusetts, USA, August 2004. IACR, Springer.

[FR06] M. Feldhofer and C. Rechberger. A case against currently used hash functions in RFID protocols. Workshop on RFID Security (RFIDSec 06), July 2006.

[FRJ05] K. Fishkin, S. Roy, and B. Jiang. Some methods for privacy in RFID communication. In Claude Castelluccia, Hannes Hartenstein, Christof Paar, and Dirk Westhoff, editors, *European Workshop on Security in Ad-hoc and Sensor Networks – ESAS 2004*, volume 3313 of *Lecture Notes in Computer Science*, pages 42–53, Heidelberg, Germany, August 2005. Springer-Verlag.

[FVPW07] M. Friedewald, E. Vildjiounaite, Y. Punie, and D. Wright. Privacy, identity and security in ambient intelligence: A scenario analysis. *Telematics and Informatics*, 24(1):15–29, 2007.

[FW09] M. Feldhofer and J. Wolkerstorfer. Hardware Implementation of Symmetric Algorithms for RFID Security. In P. Kitsos and Y. Zhang, editors, *RFID Security*, pages 373–415. Springer US, 2009.

[FWR05] Martin Feldhofer, Johannes Wolkerstorfer, and Vincent Rijmen. AES Implementation on a Grain of Sand. *Information Security*, 152(1):13–20, October 2005.

[Gas03] B. Gassend. Physical Random Functions. Master's thesis, Massachusetts Institute of Technology, 2003.

[GCvDD02] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Silicon physical random functions. In *9th ACM conference on Computer and communications security*, pages 148–160, New York, NY, USA, 2002. ACM.

[GKM+08] F. Garcia, G. Koning, R. Muijrers, P. van Rossum, R. Verdult, R. Wichers, and B. Jacobs. Dismantling MIFARE Classic. In *Computer Security - ESORICS*, pages 97–114, 2008.

[GM09] B. Groza and M. Minea. A calculus to detect guessing attacks. In *Information Security*, pages 59–67. Springer, 2009.

[GM11] B. Groza and M. Minea. Formal modelling and automatic detection of resource exhaustion attacks. In *6th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2011)*, pages 326–333. ACM, 2011.

[Haa98] M. Haahr. True random number service, 1998. [Online, last access Oct. 2012] Available at `http://random.org`.

[Han07] G. Hancke. Noisy carrier modulation for HF RFID. In *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.

[HBF07] D.E. Holcomb, W.P. Burleson, and K. Fu. Initial SRAM State as a Fingerprint and Source of True Random Numbers for RFID Tags. In *Third International Conference on RFID Security (RFIDSec 2007)*, Malaga, Spain, 2007.

[Her86] T. Herlestam. On functions of linear shift register sequences. In *Advances in Cryptology (EUROCRYPT'85)*, pages 119–129. Springer, 1986.

[HJM07] M. Hell, T. Johansson, and W. Meier. Grain: a stream cipher for constrained environments. *International Journal of Wireless and Mobile Computing,*, 2(1):86–93, 2007.

[HM04] D. Henrici and P. Müller. Hash-Based Enhancement of Location Privacy for Radio-Frequency Identification Devices Using Varying Identifiers. In *International Workshop on Pervasive Computing and Communication Security (PerSec 2004)*, pages 149–153, Orlando, Florida, USA, March 2004. IEEE, IEEE Computer Society.

[HOM+11] Y. Hanatani, M. Ohkubo, S. Matsuo, K. Sakiyama, and K. Ohta. A Study on Computational Formal Verification for Practical Cryptographic Protocol: The Case of Synchronous RFID Authentication. In *Financial Cryptography Workshops*, pages 70–87, 2011.

[HRT+95] S. Hellebrand, J. Rajskia, S. Tarnick, S. Venkataraman, and B. Courtois. Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers. *IEEE Transactions on Computers,*, 44(2):223–233, Feb. 1995.

[HSH+06] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In *Cryptographic Hardware and Embedded Systems (CHES 2006)*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer, November 2006.

[HTKC06] D. Han, T. Takagi, H. Kim, and K. Chung. New Security Problem in RFID Systems Tag Killing. In *Computational Science and its Applications (ICCSA 2006)*, volume 3982 of *Lecture Notes in Computer Science*, pages 375–384. Springer, 2006.

[Hu08] S. S. Hu. Virtual Fence along borders, The Washington Post, February 28, 2008.

[IDE09] Rowley Crossworks IDE. Crossworks v1.4 and v2.0 for AVR, 2009.

[Isr06] P. Israsena. Securing ubiquitous and low-cost RFID using tiny encryption algorithm. In *International Symposium on Wireless Pervasive Computing*, Phuket, Thailand, January 2006. IEEE, IEEE Press.

[IY06] M. Iwamoto and H. Yamamoto. Strongly secure ramp secret sharing schemes for general access structures. *Information processing letters*, 97(2):52–57, 2006.

[Jou09] A. Joux. *Algorithmic cryptanalysis*. CRC Press, 2009.

[JP03] A. Juels and R. Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. In *Financial Cryptography (FC'03)*, volume 2742 of *Lecture Notes in Computer Science*, pages 103–121, Le Gosier, Guadeloupe, French West Indies, January 2003. IFCA, Springer.

[JPP08] A. Juels, R. Pappu, and B. Parno. Unidirectional Key Distribution Across Time and Space with Applications to RFID Security. In *USENIX Security Symposium*, San Jose, CA, July-August 2008. USENIX.

[Jue04] A. Juels. Minimalist cryptography for low-cost RFID tags. In *International Conference on Security in Communication Networks (SCN 2004)*, volume 3352 of *Lecture Notes in Computer Science*, pages 149–164, Amalfi, Italia, September 2004. Springer.

[Jue06] A. Juels. RFID security and privacy: A research survey. *IEEE Journal on Selected Areas in Communication*, 24(2):381–394, February 2006.

[JW05] A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology (CRYPTO'05)*, volume 3126 of *Lecture Notes in Computer Science*, pages 293–308, Santa Barbara, California, USA, August 2005. IACR, Springer.

[JW09] A. Juels and S.-A. Weis. Defining strong privacy for RFID. *ACM Transactions on Information and System Security (TISSEC)*, 13(1):7, 2009.

[Kan86] S.M. Kang. Accurate simulation of power dissipation in VLSI circuits. *IEEE Journal of Solid-State Circuits,*, 21(5):889–897, 1986.

[KGH83] E. Karnin, J. Greene, and M. Hellman. On secret sharing systems. *Information Theory, IEEE Transactions on*, 29(1):35–41, 1983.

[KHK+03] S. Kinoshita, F. Hoshino, T. Komuro, A. Fujimura, and M. Ohkubo. *Non identifiable anonymous-ID scheme for RFID privacy protection (Japanese)*. Auto-ID Labs, http://www.autoidlabs.com/whitepaper/KEI-AUTOID-WH004.pdf, 2003. White Paper Series.

[KLA05] S. Kumar, T. Lai, and A. Arora. Barrier coverage with wireless sensors. In *11th annual international conference on Mobile computing and networking*, pages 284–298. ACM, 2005.

[KOK+08] H.-S. Kim, J.-H. Oh, J.-B. Kim, Y.-O. Jeong, and J.-Y. Choi. Formal Verification of Cryptographic Protocol for Secure RFID System. In *4rth International Conference on Networked Computing and Advanced Information Management (NCM'08)*, volume 2, pages 470–477. IEEE, 2008.

[Kre11] S. Kremer. *Modelling and analyzing security protocols in cryptographic process calculi*. PhD thesis, École normale supérieure de Cachan-ENS Cachan, 2011.

[KT11] R. Küsters and T. Truderung. Reducing Protocol Analysis with XOR to the XOR-Free Case in the Horn Theory Based Approach. *J. Autom. Reasoning*, 46(3-4):325–352, 2011.

[KYWG10] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan. Lightweight mutual authentication and ownership transfer for RFID systems. In *29th IEEE Conference on Computer Communications (INFOCOM 2010)*, pages 1–5. IEEE, 2010.

[LAK06] S. Lee, T. Asano, and K. Kim. RFID mutual authentication scheme based on synchronized secret information. In *Symposium on Cryptography and Information Security*, Hiroshima, Japan, January 2006.

[Lam94] L. Lamport. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems–TOPLAS'94*, 16:872–923, 1994.

[LH07] H. R. Lee and D. W. Hong. The tag authentication scheme using self-shrinking generator on RFID system. *International Journal of Applied Science, Engineering and Technology,*, 3:33–38, 2007.

[LHLL05] S. Lee, Y. Ju Hwang, D.-H. L., and J. Lim. Efficient authentication for low-cost RFID systems. In *International Conference on Computational Science and its Applications (ICCSA 2005)*, volume 3480 of *Lecture Notes in Computer Science*, pages 619–627, Singapore, May 2005. Springer.

[LK06] C.-H. Lim and T. Kwon. Strong and robust RFID authentication enabling perfect ownership transfer. In *8th international conference on Information and Communications Security (ICICS'06)*, pages 1–20, 2006.

[LLG+05] D. Lim, JW Lee, B. Gassend, GE Suh, M. van Dijk, and S. Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(10):1200–1205, 2005.

[LM07a] M. Langheinrich and R. Marti. Practical minimalist cryptography for RFID privacy. *Systems Journal, IEEE*, 1(2):115–128, 2007.

[LM07b] M. Langheinrich and R. Marti. RFID privacy using spatially distributed shared secrets. In *Ubiquitous Computing Systems*, pages 1–16. Springer, 2007.

[LP06] Z. Liu and D. Peng. True random number generator in RFID systems against traceability. In *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, volume 1, pages 620–624, 2006.

[LVHH11] J.-W. Lee, D. H. T. Vo, Q.-H. Huynh, and S. H. Hong. A Fully Integrated HF-Band Passive RFID Tag IC Using 0.18-μm CMOS Technology for Low-Cost Security Applications. *Industrial Electronics, IEEE Transactions on*, 58(6):2531–2540, 2011.

[MAD03] S. Mangard, M. Aigner, and S. Dominikus. A highly regular and scalable AES hardware architecture. *IEEE Transactions on Computers*, 52(4):483–491, April 2003.

[Mas69] J. Massey. Shift-register synthesis and BCH decoding. *Information Theory, IEEE Transactions on*, 15(1):122–127, 1969.

[Mey00] C. Meyer. *Matrix analysis and applied linear algebra book and solutions manual*, volume 2. Siam, 2000.

[MM05] Minime and Mahajivana. RFID Zapper. 22nd Chaos Communication Congress (22C3), December 2005.

[MM11] A. Mahdi and T. Mohammad. A Privacy-friendly RFID Protocol using Reusable Anonymous Tickets. In *10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'11)*, pages 206–213. IEEE, 2011.

[MMO04] T. Migler, K.E. Morrison, and M. Ogle. Weight and rank of matrices over finite fields. *arXiv preprint math/0403314*, 2004.

[MOV01] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone. *Handbook of applied cryptography*. CRC Press, Fifth Printing, 2001.

[MS95] W. Meier and O. Staffelbach. The self-shrinking generator. In *Advances in Cryptology - EUROCRYPT'94*, pages 205–214. Springer, 1995.

[MS11] J. Melià-Seguí. *Lightweight PRNG for Low-Cost Passive RFID Security Improvement*. PhD thesis, Universitat Oberta de Catalunya, 2011.

[MSQ07] F. Mace, F.-X. Standaert, and J.-J. Quisquater. ASIC Implementations of the Block Cipher SEA for Constrained Applications. In *Conference on RFID Security*, pages 103–114, Malaga, Spain, July 2007.

[MSW05] D. Molnar, A. Soppera, and D. Wagner. A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags. In *Selected Areas in Cryptography (SAC 2005)*, volume 3897 of *Lecture Notes in Computer Science*, pages 276–290, Kingston, Canada, August 2005. Springer.

[MT72] C.H. Meyer and W.L. Tuchman. Pseudorandom codes can be cracked. *Electronic Design*, 23, 1972.

[MW04] D. Molnar and D. Wagner. Privacy and security in library RFID: Issues, practices, and architectures. In *ACM Conference on Computer and Communications Security (CCS 2004)*, pages 210–219, Washington, DC, USA, October 2004. ACM, ACM Press.

[NF13] E. Nataf and O. Festor. Accurate online estimation of battery lifetime for wireless sensors network. In *2nd International Conference on Sensor Networks (SENSORNETS 2013)*, pages 59–64, 2013.

[NIS08] NIST. Random number generation. National Institute of Standards and Technology., 2008.

[NLR11] P. Najera, J. Lopez, and R. Roman. Real-time location and inpatient care systems based on passive RFID. *J. Network and Computer Applications*, 34(3):980–989, 2011.

[Ore07] Y. Oren. Remote power analysis of RFID tags. Cryptology ePrint Archive, Report 2007/330, 2007.

[OSK03] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic Approach to Privacy-Friendly Tags. In *RFID Privacy Workshop*, 2003.

[Pap01] R. Pappu. *Physical One-Way Functions*. PhD thesis, Massachusetts Institute of Technology, 2001.

[PC96] C.S. Petrie and J.A. Connelly. Modeling and simulation of oscillator-based random number generators. In *Circuits and Systems, IEEE International Symposium on*, volume 4, pages 324–327, May 1996.

[Pen55] R. Penrose. A generalized inverse for matrices. In *Proc. Cambridge Philos. Soc*, volume 51(3), pages 406–413. Cambridge Univ Press, 1955.

[PLHCETR09] P. Peris-Lopez, J. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. LAMED a PRNG for EPC Class-1 Generation-2 RFID specification. *Computer Standards & Interfaces*, 31(1):88–97, Dec. 2009.

[PPR09] C. Paar, A. Poschmann, and M.J.B. Robshaw. New Designs in Lightweight Symmetric Encryption. In P. Kitsos and Y. Zhang, editors, *RFID Security*, pages 349–371. Springer US, 2009.

[Pré03] A. Prékopa. Probabilistic programming. *Handbooks in operations research and management science*, 10:267–351, 2003.

[PSJ+05] M. Philipose, J.-R. Smith, B. Jiang, A. Mamishev, S. Roy, and K. Sundara-Rajan. Battery-free wireless identification and sensing. *IEEE Pervasive Computing*, 4(1):37–45, 2005.

[RAHN03] P. Rosinger, B.M. Al-Hashimi, and N. Nicolici. Dual multiple-polynomial LFSR for low-power mixed-mode BIST. *IEEE Proceedings on Computers and Digital Techniques,*, 150(4):209–217, Jul. 2003.

[RC08] D. C. Ranasinghe and P. H. Cole. *Networked RFID Systems and Lightweight Cryptography, Chapter 8*, chapter An Evaluation Framework, pages 157–167. Springer, November 2008.

[RCT06] M. Rieback, B. Crispo, and A. Tanenbaum. Is your cat infected with a computer virus? In *Pervasive Computing and Communications*, Pisa, Italy, March 2006. IEEE, IEEE Computer Society Press.

[RDT09] G. Roussos, S. Duri, and C. Thompson. RFID meets the internet. *IEEE Internet Computing*, 13(1):11–13, 2009.

[REC04a] D. Ranasinghe, D. Engels, and P. Cole. Low-cost RFID systems: Confronting security and privacy. In *Auto-ID Labs Research Workshop*, Zurich, Switzerland, September 2004.

[REC04b] D. Ranasinghe, D. Engels, and P. Cole. Security and privacy solutions for low cost RFID Systems. In *2004 Intelligent Sensors, Sensor Networks & Information Processing Conference*, pages 337–342, Melbourne, Australia, 2004.

[RFI09] CAEN RFID. The A829EU RFID reader, 2009.

[Sar01] S. E. Sarma. Toward the 5 cent Tag. White Paper, November 2001. Auto-ID Center.

[SBCC+07] M. Salah-Bouassida, N. Chridi, I. Chrisment, O. Festor, and L. Vigneron. Automated verification of a key management architecture for hierarchical group protocols. *Annales des Télécommunications*, 62(11-12):1365–1387, 2007.

[SBCM07] J. Sounderpandian, R. V. Boppana, S. Chalasani, and A. M. Madni. Models for cost-benefit analysis of RFID implementations in retail stores. *Systems Journal, IEEE*, 1(2):105–114, Dec. 2007.

[Sch96] B. Schneier. *Applied Cryptography*. John Wiley and Sons, 1996.

[SDFMBD07] A. Solanas, J. Domingo-Ferrer, A. Martínez-Ballesté, and V. Daza. A distributed architecture for scalable private RFID tag identification. *Computer Networks*, 51(9), January 2007.

[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[SIC07] SIC. UHF RFID Demo Tag. Technical report, Stiftung Secure Information and Communication Technologies, 2007.

[SM08] B. Song and C.-J. Mitchell. RFID authentication protocol for low-cost tags. In *1st ACM conference on Wireless Network Security (WiSec 2008)*, pages 140–147, 2008.

[SSC+02] S. Sen, C. Shaw, D.-R. Chowdhuri, N. Ganguly, and P.-P. Chaudhuri. Cellular automata based cryptosystem (CAC). In *4th International Conference on Information and Communications Security (ICICS'02)*, pages 303–314, London, UK, 2002. Springer-Verlag.

[ST05] B. Skoric and P. Tuyls. Secret key generation from classical physics. *Philips Research Book Series*, September 2005.

[TB06] P. Tuyls and L. Batina. RFID-tags for anti-counterfeiting. In *Topics in*

Cryptology -CT-RSA 2006, The Cryptographers' Track at the RSA Conference, Lecture Notes in Computer Science, San Jose, California, USA, February 2006. Springer.

[Tri00] E. Triantaphyllou. *Multi-criteria decision making methods*. Springer, 2000.

[Tsu06] G. Tsudik. YA-TRAP: Yet another trivial RFID authentication protocol. In *4th IEEE Conference on Pervasive Computing and Communications Workshops (PerCom 2006 Workshops)*, pages 640–643. IEEE, 2006.

[TT08] K. Toohey and T. Taylor. Mega events, fear, and risk: terrorism at the olympic games. *Journal of Sport Management*, 22(4):451–469, 2008.

[TUI+01] K. Takaragi, M. Usami, R. Imura, R. Itsuki, and T. Satoh. An ultra small individual recognition security chip. *IEEE Micro*, 21(6):43–49, 2001.

[VALMGC+10] J. Vales-Alonso, P. López-Matencio, F.-J. Gonzalez-Castaño, H. Navarro-Hellín, P.-J. Baños-Guirao, F.-J. Pérez-Martínez, R. Martínez-Álvarez, D. González-Jiménez, F. Gil-Castiñeira, and R. Duro-Fernández. Ambient intelligence systems for personalized sport training. *Sensors*, 10(3):2359–2385, 2010.

[VJS10] M. Valocchi, J. Juliano, and A. Schurr. Switching Perspectives – Creating new business models for a changing world of energy, March 2010. Executive report, IBM Institute for Business Value Publication.

[VLBDM07] T. Van Le, M. Burmester, and B. De Medeiros. Universally composable and forward-secure RFID authentication and authenticated key exchange. In *2nd ACM symposium on Information, Computer and Communications Security*, pages 242–252. ACM, 2007.

[VSK03] I. Verbauwhede, P. Schaumont, and H. Kuo. Design and performance testing of a 2.29-GB/s Rijndael processor. *IEEE Journal of Solid-State Circuits*, 38(3):569–572, March 2003.

[Wan06] R. Want. RFID explained: A primer on radio frequency identification technologies. *Synthesis Lectures on Mobile and Pervasive Computing*, 1(1):1–94, 2006.

[WKHW02] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE Transactions on Industrial Electronics*, 49(6):1265–1282, December 2002.

[WN95] D. J. Wheeler and R. M. Needham. TEA, a Tiny Encryption Algorithm. In *Fast Software Encryption: Second International Workshop, Leuven, Belgium, December*, volume 1008 of *Lecture Notes in Computer Science*, pages 363–366. Springer, 1995.

[Wol86] S. Wolfram. Cryptography with cellular automata. In *Advances in cryptology (CRYPTO 85)*, volume 218 of *Lecture Notes in Computer Science*, pages 429–432, New York, NY, USA, 1986. Springer-Verlag New York, Inc.

[Wol05] J. Wolkerstorfer. Is Elliptic-Curve Cryptography Suitable to Secure RFID Tags? Handout of the Ecrypt Workshop on RFID and Lightweight Crypto, July 2005.

[WSRE03a] S. Weis, S. Sarma, R. Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *International Conference on Security in Pervasive Computing (SPC 2003)*, volume 2802 of *Lecture Notes in Computer Science*, pages 454–469, Boppard, Germany, March 2003. Springer-Verlag.

[WSRE03b] S. Weis, S. Sarma, R. Rivest, and D. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In *International Conference on Security in Pervasive Computing (SPC 2003)*, volume 2802 of *Lecture Notes in Computer Science*, pages 454–469, Boppard, Germany, March 2003. Springer.

[Yam86] H. Yamamoto. On secret sharing systems using (k,l,n)-threshold scheme. *IECE Trans., Electronics and Communications in Japan (Part I: Communications)*, 68-A(9):945–952, 1986.

[ZK05] X. Zhang and B. King. Integrity improvements to an RFID privacy protection protocol for anti-counterfeiting. In *Information Security Conference (ISC 2005)*, volume 3650 of *Lecture Notes in Computer Science*, pages 474–481, Singapore, September 2005. Springer.

[ZZW08] S.H. Zhou, W. Zhang, and N.J. Wu. An ultra-low power CMOS random number generator. *Solid-State Electronics Journal,*, 52(2):233–238, 2008.

Publications List

[167] Patrick Battistello, Joaquin Garcia-Alfaro, and Cyril Delétré. Transaction-based authentication and key agreement protocol for inter-domain voip. *Journal of Network and Computer Applications*, 35(5):1579–1597, September 2012.

[168] Gimer Cervera, Michel Barbeau, Joaquin Garcia-Alfaro, and Evangelos Kranakis. A multipath routing strategy to prevent flooding disruption attacks in link state routing protocols for manets. *Journal of Network and Computer Applications*, 36(2):744–755, 2013.

[169] Gimer Cervera, Michel Barbeau, Joaquin Garcia-Alfaro, and Evangelos Kranakis. Security issues in link state routing protocols for manets. In *Advances in Network Analysis and its Applications*, pages 117–148. Springer, 2013.

[170] Frédéric Cuppens, Fabien Autrel, Yacine Bouzida, Joaquin Garcia, Sylvain Gombault, and Thierry Sans. Anti-correlation as a criterion to select appropriate countermeasures in an intrusion detection framework. In *Annals of telecommunications*, volume 61, pages 197–217. Springer, 2006.

[171] Frédéric Cuppens, Nora Cuppens-Boulahia, Joaquin Garcia-Alfaro, Tarik Moataz, and Xavier Rimasson. Handling stateful firewall anomalies. In *27th IFIP TC-11 International Information Security Conference (IFIP SEC 2012)*, pages 174–186. Springer, 2012.

[172] Joaquin Garcia-Alfaro, Fabien Autrel, Joan Borrell, Sergio Castillo, Frédéric Cuppens, and Guillermo Navarro. Decentralized publish-subscribe system to prevent

coordinated attacks via alert correlation. In *6th International Conference on Information and Communications Security Information and Communications Security*, volume 3269 of *Lecture Notes in Computer Science*, pages 223–235. Springer, 2004.

[173] Joaquin Garcia-Alfaro, Michel Barbeau, and Evangelos Kranakis. Evaluation of Anonymized ONS Queries. In *International Workshop on Autonomous and Spontaneous Security (SETOP 2008)*, pages 47–60. Editions Publibook Universite, 2008.

[174] Joaquin Garcia-Alfaro, Michel Barbeau, and Evangelos Kranakis. Secure localization of nodes in wireless sensor networks with limited number of truth tellers. In *Seventh Annual Communication Networks and Services Research Conference (CNSR'09)*, pages 86–93. IEEE, 2009.

[175] Joaquin Garcia-Alfaro, Michel Barbeau, and Evangelos Kranakis. Handling security threats to the rfid system of epc networks. In *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET; Chapter 3*, pages 45–64. Auerbach Publications, Taylor & Francis Group, 2010.

[176] Joaquin Garcia-Alfaro, Michel Barbeau, and Evangelos Kranakis. Security threat mitigation trends in low-cost RFID systems. In *2nd International Workshop on Autonomous and Spontaneous Security (SETOP 2009)*, volume 5939 of *Lecture Notes in Computer Science*, pages 193–207. Springer, 2010.

[177] Joaquin Garcia-Alfaro, Michel Barbeau, and Evangelos Kranakis. Proactive threshold cryptosystem for epc tags. *Ad Hoc & Sensor Wireless Networks*, 12(3-4):187–208, May 2011.

[178] Joaquin Garcia-Alfaro, Michel Barbeau, and Evangelos Kranakis. Secure geolocalization of wireless sensor nodes in the presence of misbehaving anchor nodes. *Annals of Telecommunications*, 66(9-10):535–552, 2011.

[179] Joaquin Garcia-Alfaro, Michel Barbeau, and Evangelos Kranakis. Security threats on EPC based RFID systems. In *5th International Conference on Information Technology: New Generations (ITNG 2008)*, pages 1242–1244. IEEE, April 2008.

[180] Joaquin Garcia-Alfaro, Michel Barbeau, and Evangelos Kranakis. Les composants RFID, sont-ils vulnerables?, July/September, 2009. Techniques de l'ingenieur, N. 4/5.

[181] Joaquin Garcia-Alfaro, Michel Barbeau, and Evangelos Kranakis. A Proactive Threshold Secret Sharing Scheme Handling Gen2 Privacy Threats. Technical report, Carleton University, March 2009.

[182] Joaquin Garcia-Alfaro, Frédéric Cuppens, and Nora Cuppens-Boulahia. Analysis of policy anomalies on distributed network security setups. In *11th European Symposium on Research in Computer Security, ESORICS 2006*, volume 4189 of *Lecture Notes in Computer Science*, pages 496–511. Springer, 2006.

[183] Joaquin Garcia-Alfaro, Frédéric Cuppens, Nora Cuppens-Boulahia, Salvador Martinez, and Jordi Cabot. Management of stateful firewall misconfiguration. *Computers & Security*, 2013.

[184] Joaquin Garcia-Alfaro, Frédéric Cuppens, Nora Cuppens-Boulahia, Salvador Martinez, and Jordi Cabot. Model-driven extraction and analysis of network security policies. In *16th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MoDELS 2013*. Springer, 2013.

[185] Joaquin Garcia-Alfaro, Frédéric Cuppens, Nora Cuppens-Boulahia, and Stere Preda. MIRAGE: a management tool for the analysis and deployment of network security policies. In *Data Privacy Management and Autonomous Spontaneous Security*, volume 6514 of *Lecture Notes in Computer Science*, pages 203–215. Springer, 2011.

[186] Joaquin Garcia-Alfaro, Nora Cuppens-Boulahia, and Frédéric Cuppens. Complete analysis of configuration rules to guarantee reliable network security policies. *International Journal of Information Security*, 7(2):103–122, 2008.

[187] Joaquin Garcia-Alfaro, Jordi Herrera-Joancomarti, and Joan Melia-Segui. A Multiple-Polynomial LFSR based Pseudorandom Number Generator Design for EPC Gen2 Systems. *Mitacs Workshop on Network Security & Cryptography, Mitacs*

Focus Period, Toronto, Canada, 2010. [Online] Available at: `http://goo.gl/5wcVHY`.

[188] Joaquin Garcia-Alfaro, Jordi Herrera-Joancomarti, and Joan Melia-Segui. Practical Eavesdropping of Control Data from EPC Gen2 Queries with a Programmable RFID Toolkit. *Hakin9*, 2011. [Online] Available at: `http://goo.gl/rCd9P9`.

[189] Nabil Hachem, Hervé Debar, and Joaquin Garcia-Alfaro. HADEGA: A novel MPLS-based mitigation solution to handle network attacks. In *31st IEEE International Performance Computing and Communications Conference (IPCCC)*, pages 171–180. IEEE, 2012.

[190] Nabil Hachem, Joaquin Garcia-Alfaro, and Hervé Debar. An Adaptive Mitigation Framework for Handling Suspicious Network Flows via MPLS Policies. In *18th Nordic Conference on Secure IT Systems (NordSec 2013), Ilulissat, Greenland*. Springer, October 2013.

[191] Joan Melia-Segui, Joaquin Garcia-Alfaro, and Jordi Herrera-Joancomarti. Analysis and improvement of a pseudorandom number generator for EPC Gen2 tags. *Financial cryptography and data security*, 6054:34–46, 2010.

[192] Joan Melia-Segui, Joaquin Garcia-Alfaro, and Jordi Herrera-Joancomarti. A practical implementation attack on weak pseudorandom number generator designs for EPC Gen2 tags. *Wireless Personal Communications*, 59(1):27–42, 2011.

[193] Joan Melia-Segui, Joaquin Garcia-Alfaro, and Jordi Herrera-Joancomarti. Multiple-polynomial LFSR based pseudorandom number generator for EPC Gen2 RFID tags. In *37th Annual Conference on IEEE Industrial Electronics Society (IECON 2011)*, pages 3820–3825. IEEE, 2011.

[194] Joan Melia-Segui, Joaquin Garcia-Alfaro, and Jordi Herrera-Joancomarti. On the similarity of commercial EPC Gen2 pseudorandom number generators. *Transactions on Emerging Telecommunications Technologies,*, 2012. DOI: 10.1002/ett.2600 [Online].

[195] Joan Melia-Segui, Joaquin Garcia-Alfaro, and Jordi Herrera-Joancomarti. J3Gen: A PRNG for Low-Cost Passive RFID. *Sensors*, 13(3):3816–3830, 2013.

[196] Eugenia Papagiannakopoulou, Maria Koukovini, Georgios Lioudakis, Joaquin Garcia-Alfaro, Dimitra Kaklamani, Iakovos Venieris, Frédéric Cuppens, and Nora Cuppens-Boulahia. A privacy-aware access control model for distributed network monitoring. *Computers & Electrical Engineering*, 2012.

[197] Stere Preda, Frédéric Cuppens, Nora Cuppens-Boulahia, Joaquin Garcia-Alfaro, and Laurent Toutain. Dynamic deployment of context-aware access control policies for constrained security devices. *Journal of Systems and Software*, 84(7):1144–1159, 2011.

[198] Stere Preda, Nora Cuppens-Boulahia, Frédéric Cuppens, Joaquin Garcia-Alfaro, and Laurent Toutain. Model-driven security policy deployment: property oriented approach. In *Engineering Secure Software and Systems (ESSOS 2010)*, volume 5965 of *Lecture Notes in Computer Science*, pages 123–139. Springer, 2010.

[199] Wei Shi, Michel Barbeau, Joaquin Garcia-Alfaro, and Jean-Pierre Corriveau. Handling the evil ring attack on localization and routing in wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 17(1-2):87–102, 2012.

[200] Wei Shi, Joaquin Garcia-Alfaro, Michel Barbeau, Jean-Pierre Corriveau, and Meng Yao. Secure Localization in Presence of Colluding Attackers in WSNs. *Pervasive and Mobile Computing, (submitted)*, 2013.

[201] Wiem Tounsi, Nora Cuppens-Boulahia, Frédéric Cuppens, and Joaquin Garcia-Alfaro. Formal Verification of a Key Establishment Protocol for EPC Gen2 RFID Systems: Work in Progress. In Joaquin Garcia-Alfaro and Pascal Lafourcade, editors, *Foundations and Practice of Security*, volume 6888 of *Lecture Notes in Computer Science*, pages 242–251. Springer Berlin Heidelberg, 2012.

[202] Wiem Tounsi, Nora Cuppens-Boulahia, Frédéric Cuppens, and Joaquin Garcia-Alfaro. Privacy-enhanced Filtering and Collection Middleware in EPCglobal Networks. In *Eighth International Conference on Risks and Security of Internet and Systems, (CRiSIS 2013), La Rochelle, France,*. IEEE, October 2013.

[203] Wiem Tounsi, Nora Cuppens-Boulahia, Joaquin Garcia-Alfaro, Yannick Chevalier, and Frédéric Cuppens. KEDGEN2: A key establishment and derivation protocol for EPC Gen2 RFID systems. *Journal of Network and Computer Applications*, 2013.

[204] Wiem Tounsi, Joaquin Garcia-Alfaro, Nora Cuppens-Boulahia, and Frédéric Cuppens. Securing the communications of home health care systems based on RFID sensor networks. In *Eighth Annual Communication Networks and Services Research Conference*, pages 284–291. IEEE, 2011.

[205] Wiem Tounsi, Joaquin Garcia-Alfaro, Nora Cuppens-Boulahia, and Frédéric Cuppens. Protocoles d'echange de cles pour des systemes de surveillance a base de radio-etiquettes. In *5th Conf. on Network Architectures and Information Systems Security (SAR-SSI 2010), Menton, France*, May 2010.

[206] Wiem Tounsi, Joaquin Garcia-Alfaro, Nora Cuppens-Boulahia, and Frédéric Cuppens. Securiser la communication dans les systemes de soins a domicile bases sur les capteurs RFID. Protocoles appliques aux etiquettes a faible cout. In *3eme Atelier sur la Gestion de Donnees dans les Systemes d'Information Pervasifs (GEDSIP'10), Marseille, France*, pages 77–91, May 2010.