INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Security and Privacy in Identification and Mobile Payments

Manuel Nunes Farinha Correia Rego

Dissertation submitted to obtain the Master Degree in

Communication Networks Engineering

Jury

President:	Prof. Luís Manuel Antunes Veiga
Supervisor:	Prof. Pedro Miguel dos Santos Alves Madeira Adão
Co-supervisor:	Prof. Ricardo Jorge Fernandes Chaves
Member:	Prof. Carlos Nuno da Cruz Ribeiro

October 2012

to everyone that made me all I am today

Acknowledgments

First and foremost I would like to recognize, acknowledge and give my most sincere acknowledgment to my parents, Maria Cecilia Rego and Fernando Rego. They gave me everything I could ever need or ask for in academics and in life in general, allowing me to expand my horizons while supporting me in whatever paths that I chose to take. To say I would not be here without them is not only a fact, but also a reflection of my true appreciation for all their efforts through the years that lead to this end of my academic career. Thank you.

Secondly, I would like to give a warm thank you to my supervisor, Professor Pedro Adão and to my co-supervisor, Professor Ricardo Chaves, for all their support through this endeavor. Their patience, overview and support was critical for the completion of this work.

Before I move on, I must also express my gratitude to my grandparents, Maria de Lurdes Nunes and Jose Farinha. While, unfortunately, they could not presence the conclusion of this final stage of my education, their contribution its earlier stages clearly left an impression, one that I believe is reflected on my best qualities as a student.

Moving on, I would like to thank all of my friends that helped, supported and motivated me through this work. It is impossible to name everyone, obviously, as each contribution, even a single detail, contributes to the sum that is this work.

Nevertheless, I would like to give a special and heart-felt thank you to João Louro and Pedro Santos, for all their support and encouragement, to Andreia Amaral for her relentless and crucial help and encouragement, especially in pushing me to a conclusion, to Ana Pereira, for all her guidance with the mathematical component of this work and to Antonio Inacio, for his support, counseling and care. Last, but certainly not least, I must also thank my brother, Antonio Rego, because, in his way, in our way, we help each other through this path of life.

I can only hope to ever repay each and everyone of you.

Resumo

Baseada na proliferação de sistemas de telecomunicacões móveis, o interesse em Pagamentos Móveis como alternativa aos sistemas de pagamento tradicional como dinheiro, cheques ou cartões de crédito/débito tem aumentado de forma bastante significativa dos últimos anos a esta data, como prova o interesse de inúmeras empresas de comunicações móveis, instituições financeiras e potências da Internet como a Google no desenvolvimento deste género de sistemas.

O objectivo deste trabalho passa pelo desenvolvimento de um sistema de Pagamento Móvel off-line, não rastreável e compacto, implementado sobre um sistema de E-Cash anónimo e compacto. Este sistema foi desenvolvido com a Segurança e Privacidade como pilares fundamentais do sistema, mantendo a privacidade dos Utilizadores em qualquer altura durante o uso normal do sistema, permitindo simultâneamente a identificação eficiente e indisputável de utilizadores mal intencionados.

Outra principal preocupação do sistema foi a Portabilidade deste para diferentes meios. Para este efeito o sistema recorre a uma arquitectura modulada, permitindo que o sistema seja portado e adaptado para sistemas distintos com recursos distintos ao seu dispor.

Palavras-chave: Pagamentos Móveis, E-Cash, Segurança, Privacidade, Portabilidade

Abstract

Based on proliferation of mobile telecommunications technology, the interest on Mobile Payments as an alternative to traditional payment methods such as cash, check or credit cards has been raising significantly on the last few years as evidenced by the interest of mobile communication companies, financial institutions and Internet powerhouses like Google on the development of such systems.

The objective of this work is the development of an off-line untraceable and compact Mobile Payment system implemented on a secure and anonymous compact E-cash scheme. This system was designed designed with the foremost concern for Security, Privacy, retaining the Privacy of Users at all times during normal usage, while simultaneously allowing for efficient and indisputable identification of double-spenders.

Another major concern was the Portability of this system to different mediums. To this effect the system relies on a modulated architecture, allowing for the system to be ported and adapted to distinct systems with different resources at their disposal.

Keywords: Mobile Payments, E-Cash, Security, Privacy, Portability

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the early stages of the 21st century, well past the teens of the Information Age, most of us still use forged metal and nickel tokens as a representation of currency. This sound and well tested concept, dated from somewhere around 900 to 600 BC, had its latest revolutionary change in the Chinese Han Dynasty in 118 BC. In an era where information is ever available and has in fact become the major worldwide currency, surpassing in value and meaning of its physical counterparts, hauling around these round pieces of metal is, to the very least, anachronistic.

Mobile payments, on the other hand, create new and unforeseen ways of convenience and commerce and are steadily becoming complement to cash, credit and debit cards, and will surely replace them, sooner or later. This assumption, while seemingly bold, is certainly in the minds of many influential people, as a myriad of mobile communication companies, financial institutions and Internet powerhouses like Google are developing, or already have implemented, mobile payment solutions.

As such, this work takes as another step in the evolution of the most outdated technology in the everyday life of most of us, proposing a mobile payment system primarily focused in Privacy, offering total anonymity to its Users, while still being able to efficiently and doubtlessly identify those that try to undermine the correct functioning of the system.

1.1 Motivation

Privacy, defined by Alan Westing as "*the claim of individuals, groups, or institutions to determine for themselves when, how and to what extend information about them is communicated to others*" [2], will surely be one of the pressing matters in the XXI century, as the information society expands from TV screens and PC monitors into our everyday life.

And if the concern for Privacy is rising in every individual's actions, there is no place where it is more poignant than in the way that people chose to spend their money. Beyond direct connections between receipts and actions of the individual that contracted them, the privacy of each of us is under a bigger thread coming from the snooping and triangulation of each small amount of money. This information is then used, and sold, for the profiling of individuals, allowing those who have access to them to anticipate

and plan with access to details that each of us wanted private.

1.2 Objectives

With this work we aim to accomplish the development of an off-line untraceable and compact Mobile Payment system implemented on a secure and anonymous compact E-cash scheme, with major concerns on User privacy and the cross-technology portability of the system to different mediums.

In this thesis, the current state of the art of Radio Frequency (namely, RFID and NFC) systems is analysed in order to fully understand their capacities and limitations. The present standing of E-Cash and Mobile payment schemes is also dissected, in order to understand what solutions exist today and the differences between such protocols.

A proof of concept system will be implemented in order to conduct a performance evaluation, on the resources that the practical implementation of our proposed Mobile payment system would require. These results are then related to the capabilities of the distinct systems in which it could be implemented.

1.3 Organization

This document is organized in the following chapters:

Chapter 2 (Related Work) In this chapter we present the state of the art in both RFID and NFC systems, going through their specific characteristics, architectural frameworks and other details. This is followed by an in-dept, yet succinct, analysis of the Security and Privacy in RF systems, including an identification of distinct types of threat and the existing solutions for said threats. We conclude this chapter by going through the fundamental concepts in Mobile Payment and overview of already deployed Mobile Payment systems.

Chapter 3 (E-Cash) Here we start by presenting the fundamentals of E-Cash, following this with a detailed description of the Camenish Compact E-Cash Scheme, going through its building blocks, such as the required mathematical concepts and formulas, actors and protocols.

Chapter 4 (Architecture and Implementation) Following the presented E-Cash System in chapter 3, here we provide a comprehensive characterization of how the work of Camenish et al. was constructed into a real-world working concept system. We start by giving a description of the Architecture of the system, with its modules, actors, and the way these mingle together in providing a portable and efficient solutions. We then go through an individual description of each of these components and the details of their implementation.

Chapter 5 (Results) In this chapter we present the results of the feasibility of the implemented proof-of-concept system. These include Performance Tests on the building blocks of the system as well as

on the principal protocols, which are then discussed and related to the capabilities of platforms in which the system could be implemented. This is followed by a system evaluation, featuring inquiries on User expectations for this kind of systems. Finally, we present some considerations on the interface that would be required by production deployment of this system, based on User feedback.

Chapter 2

Related Work

As soon as 1915 there is notice of the development of Identification Friend or Foe (IFF) systems, used in World War II in order to remotely identify aircrafts as friends or foes, a system that is used to this day. Later, in October 1948 [32], Harry Stockman laid the path to what would become RFID with his thesis "Communication by Means of Reflected Power". Surely ahead of its time, the idea would have to wait for the development of the transistor, the integrated circuit, the microprocessor, and communication networks in general in order to be applied.

In the 80's the full implementation of RF technology was deployed, with the first commercial application happening in 1987 in Norway [32], as a toll collection system. These kind of systems meet huge success, such that in the 1990's they were implemented in wide scale in various places through out the world, such as the United States, Australia, China, Hong Kong, Argentina, Brazil, Canada, Japan, Africa, and several countries in Europe.

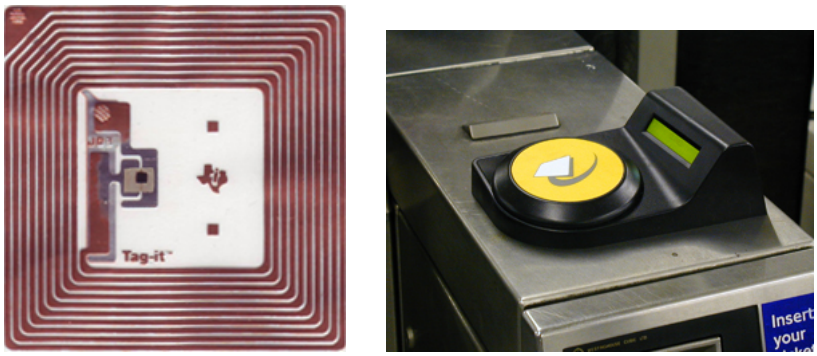
Recently, due to the decrease in cost of RF Tags and the remaining components of their systems, there has been an explosion of RF usage. For instance, analyst firm IDTechEx [8] estimate that 2.31 billion RFID Tags were sold in 2010 worldwide, up from 1.98 billion in 2009 and a value of the entire RFID market to reach 5.63 billion dollars, up from 5.03 billion dollars in 2009. In Japan alone, the market for tags, other system components, and software is expected to grow from 64 million Euro in 2000 to 2.75 billion Euro in 2013, an astonishing 4200% increase [35].

2.1 RFID Systems

RFID systems are composed of three key elements:

- RFID Tag, or *Transponder* - carries object identifying data;
- RFID Reader, or *Transceiver* - reads and writes RFID Tag data;
- Back-end - utilizes the data obtained from the transceiver in some useful manner.

An *RFID Tag* consists of a small microchip, embedded in an integrated circuit, used for storage and performing simple logical operations. The microchip itself can be very small, reaching 0.4mm x



(a) (b)

Figure 2.1: RFID Tag (a) and Reader (b)

0.4mm [53], while the hole Tag will usually also include an antenna, and a connection between both components. The memory present on the microchip might be, depending on the necessities of the systems and required cost of the Tags, read-only, write-once, read-many, or fully rewritable.

RFID Tags can be divided into 3 main categories:

- Passive - When there is no form of power supply in the Tag;
- Semi-Passive - When a battery is used to boost response signal;
- Active - When a battery is used to allow advanced processing abilities and/or to increase range and autonomy of the Tag.

Passive Tags, having to get their energy from the Reader have their communication range limited by both the need for the Reader to power the Tag and by the limited amount of energy that the Tag can collect. However, this “limitation” pays off and by a long way, in a much longer life-cycle due to the lack of a continuous power source, and, most importantly, in a production cost much lower than any of its counterparts [50].

Semi-Passive and *Active Tags*, on the other hand, resort to batteries to improve their communication range. Active Tags themselves may also be equipped with built-in sensors. e.g., temperature, pressure, etc., and have much higher processing capabilities, allowing for significantly more secure transactions. This comes at the cost of a much shorter life-cycle, more fragility, and significantly more expensive hardware, which limits their usage to very specific situations (e.g. high-value work in process inventory in aerospace and automotive manufacturing sectors) and prevents both these solutions from ever reaching the numbers of passive Tags.

It is important to point out, however, that most transponders, either passive, semi-passive, or active, only respond when queried by a transceiver. Active communications by a Tag are only found on very specific scenarios, are most likely to correspond to a rogue system.

RFID Readers consist of a radio frequency module, a control unit and a coupling element to interrogate electronic Tags via radio frequency communication [51], most of them including some internal storage and processing power. They are deployed in strategic locations where the data from the

Band	**Low Frequency**	**High Frequency**	**Ultra High Frequency**	**Microwave**
Frequency	125-134kHz	13.56MHz	865-956MHz	2.45GHz
Read Range	$<$ 0.5m	Up to 1.5m	0.5 to 5m	Up to 10m
Transfer Rate	$<$ 1kbit/s	Aprox. 25kbit/s	Aprox. 30kbit/s	Up to 100kbit/s
Typical Uses	Animal ID, Car immobilizer	Access & Security	Logistics, Animal Tracking	Moving Vehicle Toll

Table 2.1: RFID Operating Frequencies and Characteristics

Transponders needs to be read, e.g., in public transportation, such as the one depicted in Fig. 2.1(b), placed at entrance and exit points of the stations.

Transceivers continuously emit an interrogation signal, creating a zone within which the Transponders can be read. The area of this zone is dependent on both Reader and Tag characteristics, with higher frequency systems achieving larger range than their lower frequency equivalents. This is not due to the usage of higher frequency by itself, but to the fact that in these systems more power is provided to the transmitter, resulting in the mentioned larger range. Active Tags, being self-powered, do not comply with the above statement, having ranges significantly larger than equivalent passive Tags.

2.1.1 Frequencies and Regulations

RFID Systems operate in unlicensed spectrum space, referred to as Industrial, Scientific and Medical (ISM) [51], freely available for use by low-power, short-range systems as designated by the International Telecommunications Union (ITU). Most (if not all) of those systems operate in four main frequency bands, presented in Table 2.1.

RFID systems based on *Low Frequency* and *High Frequency* (HF) make use of the near field communication and the physical property of inductive coupling from a magnetic field [55], with the creation of a magnetic field between the Reader and Tag inducting an electric current in the Tag's antenna, and thus powering the IC.

On the other hand, systems based on *Ultra High Frequency* and *Microwave* resort to far field communication and the physical property of backscattering or "reflected" power [55], by transmitting electric radio waves that are then reflected by the Tag's antenna after the modulation (shifting amplitude or phase of the waves) of the received signal.

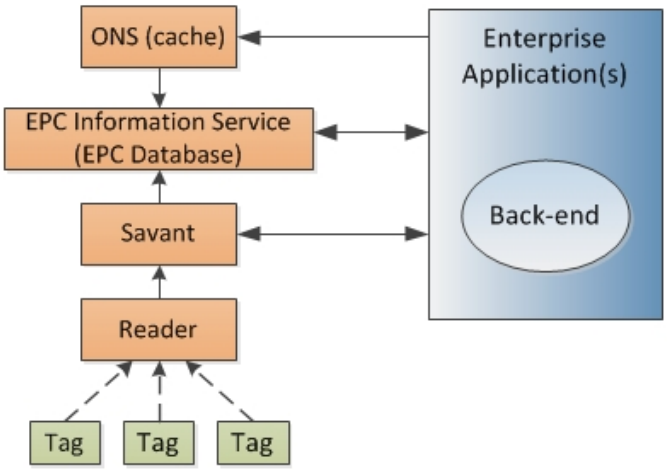


Figure 2.2: Overview of the EPC Architecture Framework

2.1.2 RFID Software

The EPC System was original designed by Auto-ID Center[1] with a minimalistic strategy for the RFID Tags [11], reckoned to be the most replicated component of the system. This approach enables extremely low-cost RFID Tags by moving the complexity to the rest of the system.

We shall now provide an overview of the primary components featured in EPC Architecture:

- Placed between the RFID Reader and the remaining of the network, the *Savant* is a middleware system [34], passing requests from Enterprise Applications to Readers and receiving from them unique Tag identifiers and possibly other data, passing that information back to the Applications. Savants also preform filtering, aggregation, and counting of Tag data, reducing the volume of data prior to sending to Enterprise Applications [15] and thus enabling distributed security and greatly enhanced scalability by providing convenient points for network isolation [51];

- Rigid and simple, much like XML, the *PML* ensures smooth data transfer in between the EPC Network and easy system setup, by standardizing message content. [34];

- The EPC Information Service (*EPCIS*) is the gateway between any requester of information and the Back-end [34], receiving requests from a number of different platforms, implemented with various different languages, and translating them into PML in order to be understood inside the EPC Network. Likewise, it collects Tag read data collected from the Savant available in PML format, and feeds it back to the requesting services in a way they can understand it;

- Lastly, the Object Naming Service (*ONS*), as would be expected from a directory service, has the task of identifying the location of the server hosting the information required by an application;

[1] Auto-ID center was the predecessor organization to EPCglobal. Besides EPCglobal, which manages the EPC network, there is a sister organization, Auto-ID Labs, that manages and funds research on EPC technology

Figure 2.3: NFC System

2.2 Near Field Communication

As RFID was the next step after Bar-codes, Near Field Contact (NFC) technology, is its natural successor. Building upon the existing RFID systems and in fact borrowing most of its technical aspects, NFC differs in the principles of its architecture and communication, potentiating RFID technology beyond its limitations.

Initially developed in 2002 by *NXP Semiconductors* and *Sony*, NFC lunged forward with the establishment of the NFC Forum, in March 2004, in which Nokia and Philips joined the two founders in order to improve the standardization of Near Field Communication systems. In 2006, a mere 18 months after its founding, the Forum formally outlined the architecture for NFC technology [41]. The same year saw the release of the first NFC phone, the Nokia 6131.

From then until now, the interest in NFC kept growing, with industry giants (e.g. Google) and major telecommunication companies in Europe (e.g. Portugal Telecom, in Portugal) presenting solutions that will soon be available to the general public.

The only major barrier to widespread usage of NFC is the limited number of NFC enabled smartphones in the market. This barrier, however, will quickly be overcome, both by the natural evolution of the devices offered and owned by the general public and by the development of solutions (e.g. TMN Wallet, by Portuguese mobile network operator TMN) that enable an NFC chip to be coupled with devices that previously did not have access to this technology.

2.2.1 NFC Systems

NFC, like RFID, is a short-range, standards-based wireless connectivity technology, employing magnetic field induction to enable communication between electronic devices in close proximity.

NFC operates in the International Standard ISO/IEC 18092, known as Near Field Communication Interface and Protocol (NFCIP-1), operating at 13.56 MHz, the same frequency band as High Frequency RFID Chips (Table 2.1). However, as a trade-off for a reduced Read Range of 20 cm instead of the 1.5

m offered by HF RFID, NFC technology allows for data transfer rates of 106 kbit/s, 212 kbit/s and 424 kbit/s, much higher than those presented by RFID Tags with even higher rates expected in the future. In addition to this, being coupled to battery-powered devices translates into much higher processing power and storage capacity and fewer, though still unavoidable, concerns about power consumption.

Unlike RFID, which is based in a fixed reader-tag structure, an NFC device is able to impersonate two different roles, *initiator* or *target* with the NFC communication protocol being based on a message and reply concept. Therefore, when one device (A) sends a message to another (B), it is then expected for B to send back a reply. It is not possible for B to send any data to A according to its own initiative. In this example A has the role of initiator, with B assuming the part of the Target.

An NFC system can also operate in two distinct modes of communication, active and passive. In *passive communication*, the initiator generates an RF field that, by induction, powers the other device. This behavior, similar to the one found in RFID systems, allows for passive receptors that can be embedded in convenient places, like movie posters, outdoor advertisements, etc. In this mode, the data is sent using a weak load modulation, encoded using Manchester coding with a modulation of 10% [19]. If the transmission rate is 106 kBaud a subcarrier frequency is used for the modulation, with the base RF signal being modulated at 13.56 MHz for transmission rates greater than said value [19].

In *active communication*, both initiator and target communicate by alternatively generating their own RF fields, which implies that both are active and as such require a dedicated power source. In this mode, data is sent using amplitude shift keying (ASK). If the transmission rate is 106 kBaud, the coding scheme used it the so-called Miller coding. If the transmission rate is greater than 106 kBaud the Manchester coding scheme is applied [19].

These modes and roles of NFC systems can be mingled, with the exception of the combination of Initiator and Passive.

It should also be mentioned that NFC communication is not limited to a pair of devices, with one initiator being able to talk to multiple targets. Broadcasting, however, is not possible and, as such, the initiator must select one receiving device before each message is sent. Said message is ignored by the remaining devices and only the selected target device is allowed to answer.

Communication between NFC devices is accomplished according to the NFC Data Exchange Format (NDEF) specification. This specification defines the NDEF data structure format as well as rules to construct valid messages as an ordered and unbroken collection of NDEF records. Furthermore, NDEF also defines the mechanism for specifying the types of application data encapsulated in NDEF records [42].

The Simple NDEF Exchange Protocol (SNEP) is a request/response protocol specified by the NFC Forum that enables the exchange of NDEF messages between devices in NFC peer-to-peer mode [44]. SNEP protocol works on top of Logical Link Control Protocol (LLCP) which, in turn, provides a connection oriented reliable transport service [43].

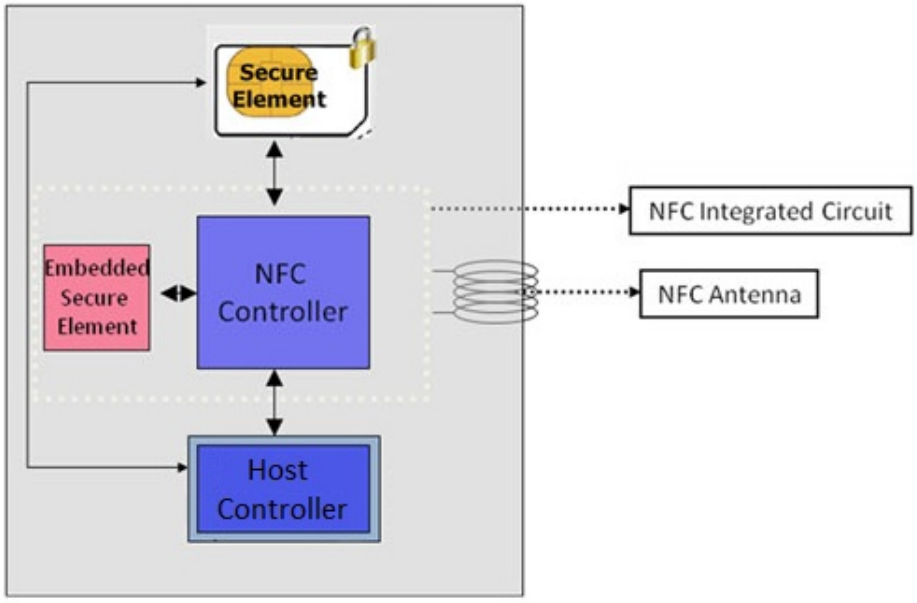


Figure 2.4: NFC Architecture

2.2.2 NFC Architecture

The main components of the NFC Architecture are the NFC Controller, the Secure Element, the Antenna and the Host-Controller [37]. The later is the Application Execution Environment (AEE), where the application lies and is executed (e.g. the mobile phone's operating system). We shall now describe the remaining components in some detail.

NFC Controller Acting as a modulator/demodulator between the analog Air Interface and the digital Host-Controller and Secure Element, the NFC Controller also links the later ones. Communications with the Host-Controller are held through interfaces such as Serial Peripheral Interface (SPI), Inter-Integrated Circuit (IIC) and Universal Serial Bus (USB). On the other hand, the channel with the Secure Element is granted by the NFC Wired Interface or the Single Wire Protocol. When in Card Emulation mode, the NFC-Controller also handles the interface that allows for the Secure Element to be powered with the energy retrieved from the Air Interface. The NFC-Controller can be connected to more than one distinct form of Secure Element.

Secure Element Seeing that most NFC applications (e.g. payment and authentication systems) require data to be securely stored, the Secure Element has to be able to execute cryptographic functions and implement a secure environment to execute security-relevant software. Given the flexibility of NFC systems, there are a number of ways to implement a Secure Element, such as:

- *Software based* - Being the most flexible and independent solution, software has the disadvantage of never being as immune to manipulation as hardware based options;
- *Device integrated* - The most host dependent but also the most reliable solution, it can be either a part of the Host or a built in chip in the NFC system. The biggest disadvantage faced by this

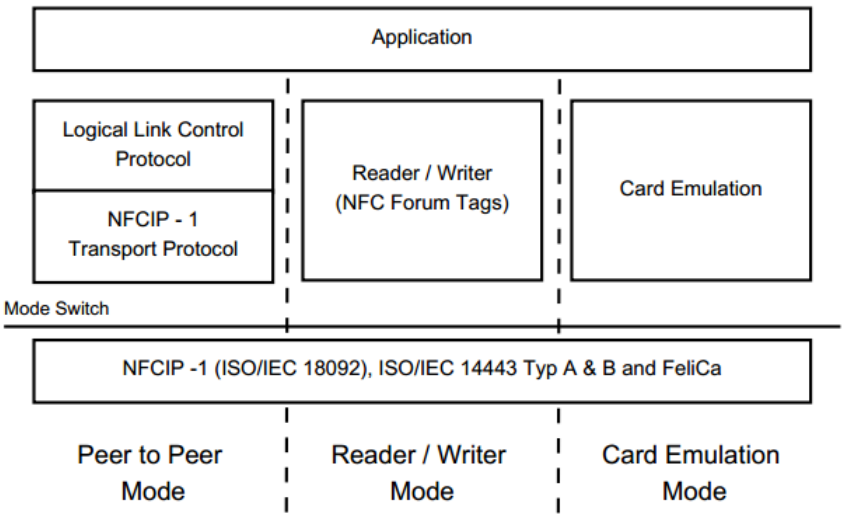


Figure 2.5: Overview of NFC Operation Modes

alternative is the very reduced, or troubled, portability of the data if the user migrates to a different device;

- *Changeable hardware* - The best compromise between reliability, usability and costs. It can be implemented with resort to a Secure Memory Card (SMC), which combines secure smart card functions with a normal memory card, or through a Universal Integrated Circuit Card (UICC), the SIM in mobile phones. In this later case, resorting to the SIM card allows the implementation on NFC even on mobile phones which originally did not provide NFC connectivity with reduced cost and trouble to the user.

2.2.3 NFC Operational Modes

In order to extend the functionality and increase compatibility, NFC systems offer different operation modes that serve different purposes and use different communication protocols (Fig. 2.5). These modes are known as Peer-to-peer Mode, Reader/Writer Mode and Card Emulation Mode [28].

Peer-to-peer Mode Enables communication between two NFC devices (ISO 18092), each of them being able to assume the role of initiator or target. This mode allows the establishment a bidirectional connection between both devices to exchange data (e.g. Bluetooth pairing information, contact information, etc.).

Reader/Writer Mode Implies that one of the devices is, typically, in passive mode. This allows for an active device to interact, with an NFC Tag - thus called reflecting the Reader/Tag protocol borrowed from RFID - that can be embedded in, for instance, a movie poster or advertisement. The NFC Tag contains a short piece of information (e.g. Internet Address) or performs a simple action on the device (e.g. connection to a Wireless or Bluetooth network), usually in order to initiate a more complex process

(e.g. allowing the user to buy movie theater tickets). This mode is fully compatible with ISO 14443 and FeliCa [2] (ISO 18092) technology, allowing for compatibility with existing RFID infrastructures.

Regarding the *Reader/Writer* mode the NFC Forum has defined four distinct formats in order to cope with the broadest possible range of applications and device capabilities:

- *Types 1 and 2* (ISO/IEC 14443 A), manufactured my Innovision and NXP and resorting o Topaz and MIFARE technology, respectively, have small memory capacity (1 and 2 kilobytes) and operate at relatively low speed (106 kbit/s), which means they are low-cost and suitable for single-use applications.
- *Type 3* (JIS X 6319-4), developed by Sony is based on FeliCa, and has larger memory (up to 1MB) and higher transfer speed (212 kbit/s) being suitable for more complex applications, but also more costly.
- *Type 4* (ISO/IEC 14443 and ISO/IEC 7816-6) is based on Smart card technology and specifies memory of up to 64KB, with transfer speeds of between 106 kbit/s and 424 kbit/s per second, making it suitable for multiple applications.

Card Emulation Mode In this mode an NFC system behaves like a smart card (ISO 14443) able to communicate with a RFID readers. This makes possible the emulation of one or more RFID Tags, enabling the user to interact with a number of previously existing infrastructures, such as contactless payment or admission control, with one NFC device. This emulation is accomplished either in application or in a Secure Element.

Combining the *Card Emulation* and *Reader/Writer* modes it is possible to implement a lightweight equivalent to the peer-to-peer mode, discarding the elaborate protocol stack implied by it. With the correct hardware implementation it even becomes possible for this mode to be used when the NFC device is powered off or with very low energy consumption [33].

2.2.4 NFC Security Mechanisms

As more and more solutions are developed with resort to NFC systems there is an increasing requirement for security solutions in these systems. Answering this need Ecma Internacional, an international standards organization, has released the NFC-SEC: NFCIP-1 Security Services and Protocol [13] and NFC-SEC-01: NFC-SEC Cryptography Standard using ECDH and AES Reference [12].

NFC-SEC (Fig. 2.6) is the first part of the *NFC Security standard series* and provides the common framework. This framework was created for securing the data exchange in peer-to-peer Mode and is placed above NFCIP-1 (Physical and MAC layers) and below higher level protocols. It defines the necessary extensions to NFCIP-1: sequence protocols and other basic conditions.

Two distinct services are offered in NFC-SEC, Shared Secret Service (SSE) and Secure Chanel Service (SCH). SSE defines methods for establishing a shared secret (key) with application specific

[2] FeliCa is a RFID smart card system developed by Sony, already deployed in Public transportation systems in Japan, China, USA, Singapore and Thailand

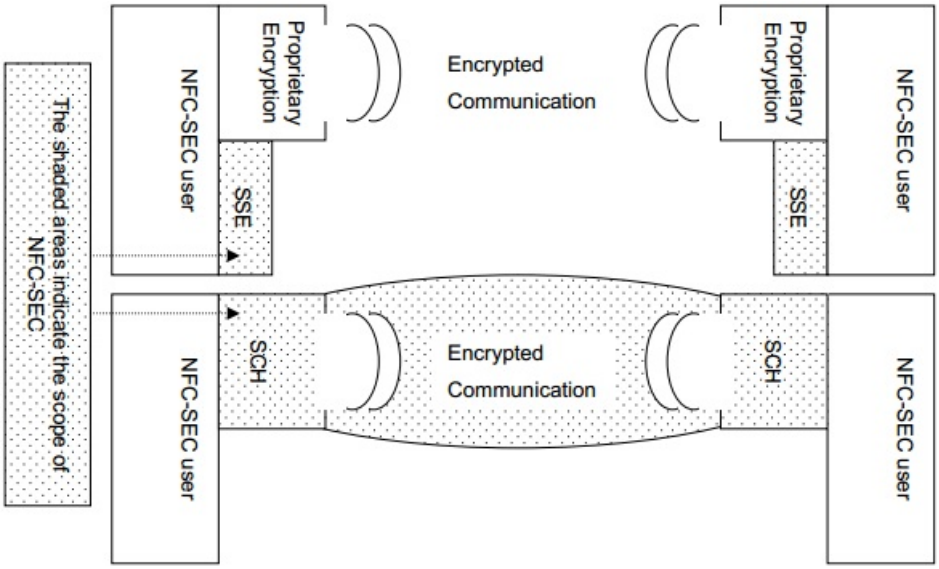


Figure 2.6: NFC-SEC overview

encryption methods while SCH not only provides the connection setup but also a secure and encrypted communication path for data messages [13].

Other parts of the NFC Security standard series, named NFC-SEC-XX define specific cryptographic mechanisms. Currently, only NFC-SEC-01 is officially available, specifying the Elliptic Curves Diffie-Hellman (ECDH) algorithm for secure key exchange and the Advanced Encryption Standard (AES) (ISO/IEC 1803-3) for data encryption. It addresses NFC communications which should be secured, and where no key is shared a priori [37] [12].

Since NFC-SEC is only targeted at peer-to-peer mode it offers no protection for an NFC device operating in Reader/Writer or Card Emulation mode. In these scenarios we can resort to Signature Record Type, which offers the possibility to assure the integrity and authenticity of an NDEF record/message, by signing it with a Signature Record.

These records contain a *Signature field*, holding either the actual signature or an URI reference, a *Certificate Chain* field, that contains the certificates necessary to authenticate the Signature Record and the authenticity of up to 14 records, and a Version Field, which is not yet very useful as there is only one version of the record, with devices that implement the Signature Record Type Definition defined by the NFC Forum having to ignore any version other than 1 (0x01).

2.3 RF Security and Privacy

Having analyzed the characteristics and architecture of RFID and NFC systems, the most important characteristic of these systems is yet to be mentioned. Unlike bar-code technology that preceded it, RF devices have the ability to be read without line-of-sight contact and without precise positioning [24]. While easy to infer, because the RF systems are based on wireless communication, this *detail* is what gives RF most of its utility. One can easily picture how much the logistics and other everyday usage can improve by upgrading from Bar-codes, that require optical scanning and thus careful positioning of the

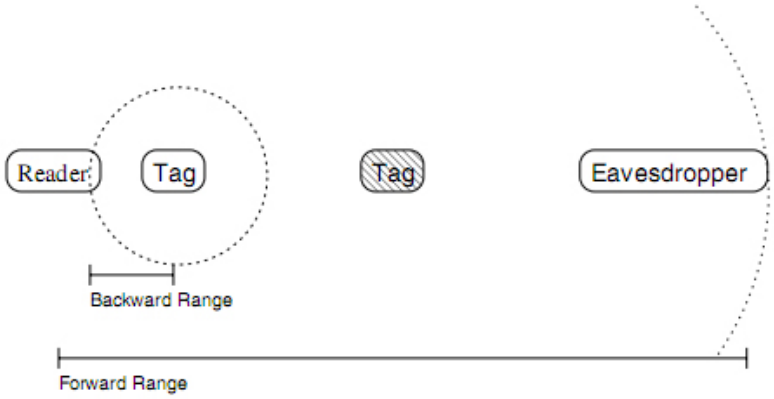


Figure 2.7: Forward & Backward RF Channels

scanned objects, to RFID and NFC systems, that can be read much faster, from greater distances and without such concerns.

The flip-side of this is that by not requiring line-of-sight, systems can be read without any evidence, creating a range of privacy threats from unauthorized data access, to snooping on communications between devices, and location tracking of physical objects and their association with people. This is aggravated by the fact that not only most of these are small enough to be covertly concealed in the environment but also that radio waves, used in the transmission of data between them, can travel through fabric, plastic, thesis and other materials, which makes it possible to embed them in floor tiles, doorways or even sheets of paper.

This translates into the urgent need to seriously consider Security and Privacy issues in the development any RF system. As such, a detailed study on the state of the art in said topics was carried out, the results of which are be presented as follows.

2.3.1 Forward and Backward Channels

Similarly to the concepts used in general data transmission, the *Forward and Backward Channels* in RF communication consist of two sides of the communication between RF systems of different roles, presenting a unique feature with noticeable consequences.

Due to the characteristics of RF systems, the range of the Forward Channel, i.e., the forward Range, is much larger than its counterpart of the Backward Channel, i.e., the backward Range. This is due to the known limitations in the range of mobile RF hardware, even semi-passive or active solutions operating in high frequency, in contrast to the capabilities of receptors, where the weight and power consumptions are usually not a concern.

In Security and Privacy concerns this peculiarity has very serious consequences. Even if a recipient has a limited range, thus supposedly reducing the danger of eavesdropping, any ill-intended user can easily intercept communications occurring in the Forward Channel (as depicted in Fig. 2.7), increasing the feasible distance between said attacker and the victims, and thus easing their concealment.

2.3.2 RF Privacy Threats

Relating to the increase in usage of RF Tags with unique IDs a number of threats to privacy arise as identified by Garfinkel et. al [18].

- *Association threat* When an individual, or some of its personal data, is associated (possibly in an involuntary or clandestine manner) with a certain RF serial number.
- *Constellation threat* Even if there is no particular identity connected to a Tag, a collection of Tags carried by an individual form a unique RF shadow or constellation, allowing the tracking of individuals without their identities being known.
- *Transaction threat* Relating to the Constellation thread, when a Tagged object moves from one constellation to another it is easy to infer a transaction between the two individuals associated with the constellations.
- *Breadcrumb threat* Expanding the Association threat, as individuals collect Tagged items they are - inadvertently - building an items database associated with their identity. The Breadcrumb threat arises because when the individuals dispose of the Tagged items, with their association to the items remaining valid, resulting not only in a direct Privacy violation but also in inadvertent consequences, e.g., if the item is latter used in a crime or other malicious act.

All of these underline the urgent call for privacy concerns in the development of RF systems, because rogue users are not even required to acknowledge the identity of the user in order to breach into his privacy. Evidence of this was recently uncovered by the generalized adoption of biometric Passports by most of the Western World (i.e. the EU and the US) which faced severe privacy breaches in early deployments. [29]

2.3.3 Existing RF Privacy Solutions

Existing privacy solutions for RFID systems take the most diverse forms, aiming to make up for the Privacy Threats presented in Section 2.3.2. It is important to notice that these have been developed with the low-cost RFID Tags in mind, with the limitation of these systems as a main concern.

Physical Approaches These approaches do not require any kind of computing power since they are based on the physical alteration of the Tag. Some examples are the Tear-off Antennas, proposed by Karjoth et. al [26] and the Granularity Reduction, proposed by Inoue and Yasuura [21], that are based on the alteration of the RF Tags, in order to, respectively, reduce their range and the information contained in them.

Other approaches to this sort of security include the isolation of the Tag, by the use of a Faraday Cage [23], the requirement of a confirmation by Optical contact, as proposed by Juels et. al [23], or recurring to Distance Analysis, as analyzed by Fishkin et. al [14].

Privacy Management Systems Instead of trusting on deployed RFID systems and their supposed privacy mechanisms RFID users might recur to their own privacy-enforcing solutions. Several approaches have already been proposed, such as *Watchdog Tag* [16], *RFID Enhancer Proxy* [24] and *RFID Guardian* [49]. While each of those proposes a slightly different approach, all of these Privacy Management Systems are, essentially, proxies, allowing the user to inspect and manage it's own RFID privacy.

Pseudonym Throttling Proposed by Juels et al. [22], Pseudonym throttling is a simple approach to RF Privacy. The concept relies on having the Tag store a short list of random identifiers - the pseudonyms - which are known by authorized Readers to be equivalent. In every query the Tag receives it cycles through the list of pseudonyms, transmitting the next one in the list, returning to the beginning when the list is depleted. Inoue et. al [21] proposed an improvement on this system through a concept of Ownership transfer, in which the detailed information about the Tag's owner would be stored in a private database. Osaka et. al [48] further built up on this approach by proposing a *Trusted Center* in which the IDs would be stored.

Cryptography Even with the limited resources of RF Systems, some of the existing privacy solutions are based on basic cryptographic concepts. Some examples of this are the Hash Lock, proposed by Weis et al. [56], based on one-way hash functions and the Hash Chain, proposed by Lamport [31] and inspired in a combination of the Hash Lock and Pseudonym mechanisms. There are also symmetric-key authentication systems such as the HB^{+} Protocol, proposed by Juels and Weis [25], adapted to cope with said limitations.

In NFC Systems, the integration of RF chips with powerful mobile systems might lead one to believe that these solutions are made irrelevant. However, even with more complex solutions at out the disposal, these constrained approaches provide us with an additional layer of security and privacy that must be considered as it might prove to be able to tackle problems not solvable even by the most powerful of encryptions (e.g. the Constellation Thread presented in section 2.3.2).

2.3.4 RF Security Threats

Besides the privacy issues detailed before, RFID and NFC systems are based on an Air Interface, and, as such, are liable to a number of attacks. These have been listed by both Haselsteiner et al. [19] and Kerschberger [28] and can be divided in attacks to the device, which we shall name as Physical Attacks, and attacks to the Air Interface.

Unlike RFID, where the Tag is, in most cases a very low cost device, in NFC the system (or part of it) is often embedded in a device which has moderate to high value. *Physical attacks*, such as the system host being stolen and/or destroyed, must then be considered in the development of any solution using those systems. Not only the losses can be far greater (e.g. a rogue user gaining access to a mobile

payment system connected to a user's bank account), but also because as the device itself is valuable it becomes more prone to arouse the interest of others.

Regarding the *Air Interface*, the already mentioned contactless nature of RF devices can allow for a number of different types of attack, such as:

- *Denial of Service (DDOS)* - Accomplished either through the jamming of the signal, performed by an emission of a disturbance signal on the frequency used by RF systems or by an generation of collisions/answers for every device in the its range, blocking legitimate communications;
- *Eavesdropping* - If two devices are using RF waves to communicate a well positioned rogue user can listen to the conversation. If the device is sending data in active mode, eavesdropping can be done up to a distance of 10 m [19] and, when in passive mode, as the attacker does not need the power of the active part of the communication for answering, it is possible to amplify weak signals for up to 30 - 40 cm [28]. We should note that these values depend heavily on the characteristics of both devices and, as such, should be taken as an illustration.
- *Man in the Middle* - When two devices are tricked into a three party communication without their knowledge, with the attacker intercepting and possibly modifying messages between the two original parties. This attack is, however, difficult to accomplish because the three devices must be in range while the attacker has to shield direct connection between the original recipients;

2.4 Mobile Payments

Mobile payments, defined by Karnouskos et al. [27] as any payment where a mobile device is used in order to initiate, activate, and/or confirm this payment. These mobile devices may include mobile phones, PDAs, wireless tablets and any other device that connect to mobile telecommunication network and make it possible for payments to be made.

This technology enables new and unforeseen ways of convenience and commerce, such as the procurement of travel, hospitality, entertainment and others, where mobile payments can become a complement to cash, credit and debit cards, or even replace them. Mobile payments can also be used for payment of bills with access to account-based payment instruments such as electronic funds transfer, Internet banking payments, direct debit and electronic bill presentment [5].

The development and deployment of Mobile Payments has been quite uneven throughout the world. In a number of countries we can find deployed technology and successful business cases implementation, such as Japan, South Korea, and other Asian countries, where several successful mobile payment solutions have already been launched (e.g., Mobile Suica, Edy, Moneta, Octopus) [47]. On the other hand, in the so called Western World (i.e. Europe and North America) the development of mobile payments has not been as successful, with the exception of some exceptional cases such as Austria, Croatia and the Scandinavian countries [5].

A major difference in these distinct regions of Mobile Payment implementation are the technologies used. In Asia the major focus is in RFID technology [52], which could be explained by the ubiquity of

contacless cards (IC cards) for payment transactions in this area. In Europe and North America the mobile payments are mostly based in SMS (Short Message Service), USSD (Unstructured Supplementary Service Data), WAP (Wireless Application Protocol) or IVR (Interactive Voice Response), in order to facilitate the uptake of mobile payments by using existing technologies in the current customer base. [47]

This leaves some crucial room for the development of NFC systems. While, as mentionet before, NFC systems are capable of interaction with existing RFID systems and therefore will face an easy transition in the Eastern markets, they also offer a much wider range of possibilities than the systems currently used in the West. In these times, where multi-function smartphones have (or will soon) become the norm [54], NFC has a worldwide open door, eagerly anticipating the solutions if can offer.

2.4.1 Mobile Payment Types

Mobile payments may be classified based on a number of criteria, namely the interaction type, basis of payment and transaction type.

Interaction Type Based on the nature of the transactions made through a mobile payment system we can split these systems in remote and proximity based payments. *Remote payments* imply that the information about each transaction is transmitted remotely to the user, usually through a Web Page that initiates the payment process (e.g. mobile banking). *Proximity payments*, on the other hand, are characterized by the fact that the user interacts directly with the merchant through the use of small range communication technologies (e.g. NFC, Bluetooth, Barcodes, etc.), avoiding the need to insert transfer details in the mobile phone.

Basis of Payment Mobile payments can be classified as account based or wallet based. In *account based* solutions the transaction is connected to an account held by the user (e.g. in a bank or telecommunication company). The amount due is debited on said account and credited to the merchant. These solutions have the advantage of a simpler implementation, due to the fact that they only require the association with a previously existing user account. *Wallet based* systems, on the other hand, require the user to create an account in the mobile payment system, dubbed a wallet, and deposit an usually limited amount of money. All debts and credits are then accomplished through that account. Wallet based solutions are usually perceived as safer by the users, due to the fact that the amount of currency involved is limited, in comparison to the balance of a Bank account.

Transaction Type There are three different types of transaction in mobile payment systems:*On-line*, where the controlling entity (i.e. the Bank) is constantly connected to the merchants and every payment must to be approved by said entity; *Off-line*, in which merchants initially accept payments and only latter submit them to the entity, having the guarantee that the payment will be honored by the bank or will lead to the identification, and therefore punishment, of the double-spender; and *Semi-Offline* systems, that

connect to the entity from time to time and, at such occasions, synchronize transactions that have been meanwhile stored in internal memory;

2.4.2 Mobile Payment Business Models

While the technical characteristics of mobile payment systems might be the primary focus of academic attention, we must not forget that other than the user, mobile payment systems must also provide value to other stakeholders along its supply chain. Each of these players has different needs, often conflicting with the expectation of others. We shall now list these stakeholders and their expectations [5].

- *Consumers* - Trust, privacy and security at the lowest possible (preferably inexistent) cost of usage and offering ubiquity, interoperability in a customized service;
- *Merchants* - Fast transaction time with high security at the lowest possible cost and featuring customization, real time monitoring and integration with existing services;
- *Financial institutions and Banks* - Network operator independent solutions with payment applications designed by the institution featuring branding opportunities, better volumes and opportunity to create, or reinforce, customer loyalty;
- *Mobile Network Operators (MNOs)* - Generation of new income due to traffic increase, increasing average revenue per user (ARPU) and customer loyalty, while becoming an attractive partner to content providers;
- *Mobile Device Manufacturers* - Low time to market and increase in ARPU, with large market adoption of embedded mobile payment applications;
- *Software and technology providers* - Low time to market and mass adoption of applications and/or technologies;
- *Governments* - Revenue through taxation and the definition and use of standards;

The regulatory environment is also one of the main focus of the business models for mobile payments. There are three types of mobile payment markets according to the financial rules and regulations followed by each country [30].

- *Highly regulated markets* (e.g. India) - Where only Banks are legally allowed to offer mobile payment services. In this scenario, only account based mobile payment systems are possible;
- *Moderately regulated markets* (e.g. USA) - Where private companies can operate mobile payment services as licensed money transmitters;
- *Minimally regulated markets* (e.g. Kenya) - Where mobile network operators are allowed to handle subscribers' cash with a MNO account and where MNOs can accept and disburse cash from its outlets;

These many possible combinations of stakeholders and regulatory environments lead to the development of different business models, such as:

- *Operator centric model* - In which the MNO independently provides mobile payment systems;
- *Bank operator centric model* - Where banks deploy mobile payments through the leveraging of existing financial debit and credit card infrastructure;
- *Peer-to-peer model* - Allows for payments to be made between customers or between customers and merchants;
- *Collaborative model* - The collaboration of several stakeholders (e.g. MNOs, Banks, etc.) in the providing of a mobile phone solution to end users;

2.4.3 Mobile Payment Solutions

We shall now briefly go through and describe some Mobile payment systems already deployed throughout the world.

Osaifu-Keitai A mobile wallet system, developed by NTT Docomo, a Japanese mobile phone operator. Resorting to RFID FeliCa smart cards, Osaifu-Keitai offers its customers access to several services, such as electronic money, credit and debit cards, member cards, transports tickets, coupons and vouchers [38]. Depending on the specific service the amount of financial transactions this system may be debited from the phone bill or from a credit or debit bank account [38]. The success this system accomplished in Japan has certainly made it a reference in RF based mobile payment systems.

mFerio An NFC based mobile peer-to-peer payment system developed in the Singapore Management University, with the aim of replacing cash based transactions [1]. It does not require any additional connectivity or infrastructure beyond the cell phones of the participants, and was designed with usability and security in mind, with those criteria being given a considerate amount of though and evaluation [1];

Online Wallets Several online companies have developed mobile payment platforms, such as PayPal (PayPal Mobile), Amazon (Amazon Payments) and Google (Google Checkout) [45]. These platforms enable users with accounts in said companies to perform mobile transactions, recurring to remote payments. Paypal Mobile also offers peer-to-peer transactions between users and while it encourages users to physically bump supported smartphones, the transaction is still accomplished remotely [45].

M-Pesa Developed by Kenya's largest mobile network operator, Safaricom, M-Pesa brings financial services to hard to reach areas while enabling users to utilize their mobile phones to deposit and withdraw money, transfer money and purchase airtime [20]. In order to facilitate the interoperability in such a particular market, M-Pesa is based in SIM toolkit, a standard software on all SIM cards [20].

Square A mobile payment service based in the credit card payments through mobile phones, Square allows users to either swipe their cards on a reader or insert their credit card number to issue a payment. The data is then encrypted, transmitted via an Internet connection and processed by Square's back-office, which, in turn, communicates with the payment network to complete the transaction [45].

Chapter 3

E-Cash

Conceived by Chaum in the beginning of the Eighties [7], electronic cash has been wide and extensively studied ever since by numerous authors. The core idea behind the concept being that while an entity , e.g., a bank, is responsible for distributing coins and afterwards collecting or receiving them for deposit, the withdrawal and spending protocols are designed in such a way that makes it impossible to identify when one particular coin was spent, i.e., making any tracing or identification of the spender impossible.

That is, of course, unless any user double-spends a token, which is a problem in the electronic world due to the easiness in duplicating data, in which case the scheme must allow for the revokement of the anonymity of the rogue user. Merchants must also be prevented from depositing the same token more than once.

Most E-Cash schemes are said to be *divisible*, which means that users can withdraw coins of 2^L value and spend said coin in several transactions, by dividing the value of the coin. The main goal of this is to allow users to efficiently spend a coin of monetary value 2^l, with $0 \leq l \leq L$, i.e., much more efficiently than repeating a spending protocol 2^l times. This idea was first implemented in a practical way by Okamoto [46] and subsequently improved by Chan et. al [6], both proposals providing anonymity of users but not unlinkability since it was possible to track several spends from a single divisible coin. Unlinkability was archived later by Nakanishi et. al [40] and further improved by the same author [39]. However, these schemes still required a trusted third party, a deficit that Canard et. al later overcame [4].

The above mentioned schemes however, have their main focus on assuring that anonymity and unlinkability are assured. Camenish et. al, on the other hand, proposed a secure *off-line* anonymous compact E-Cash scheme [3] aimed to address the efficiency issue in a concise use of resources.

3.1 Camenish Compact E-Cash Scheme

This scheme, developed with a clear focus on efficiency, was proposed in 2006 by Jan Camenish, Susan Hohenberger and Anna Lysyanskaya, allows a user to withdraw a wallet with 2^l coins, such that the space required to store these coins, and the complexity of the withdrawal protocol, are proportional

to l, rather than to 2^l.

As such, in this proposal, a wallet containing k coins can be withdrawn and spent with $O(l + k)$ complexity, while the it also takes $O(l + k)$ to store all the coins, based on the *Strong RSA Assumption* [17] and the *Decisional Diffe-Hellman Inversion*(*y*-DDHI) [10] assumptions in the random-oracle model. This is achieved through the usage of Pseudo-random functions and we shall delve into the mechanisms that are involved in this process.

Before we do, however, it is important to define the fundamental building blocks on which said mechanisms are based.

3.2 Complexity Assumptions

The security of our e-cash systems is based on the following assumptions:

Strong RSA Assumption Given an RSA modulus n and a random element $g \in \mathbb{Z}_n^*$, it is hard to compute $h \in \mathbb{Z}_n^*$, and integer $e > 1$ such that $h^e \equiv g \mod n$. The modulus n is of a special form pq, where $p = 2p^{'} + 1$ and $q = 2q^{'} + 1$ are safe primes. [17]

***y*-Decisional Diffie-Hellman Inversion Assumption (y-DDHI)** Given a random generator $g \in G$, where G has prime order q, the values $(g, g^x, ..., g^{(x^y)})$ for a random $x \in \mathbb{Z}_q$, and a value $R \in G$, it is hard to decide if $R = g^{1/x}$ or not. [17]

3.3 Global Parameters

Let 1^k be the security parameter and let l be any value in $O(\log k)$. In the subsequent protocols we require the existence of two groups:

- $\mathbf{G} = \langle \mathbf{g} \rangle$, where n is a special RSA modulus of $2k$ bits, $\mathbf{g}$ is a quadratic residue modulo n, and $\mathbf{g} \in \mathbf{G}$. This group is used in the Pedersen commitments and RSA-based CL signatures;

- $G = \langle g \rangle$, where g is an element of prime order $q = \Theta(2^k)$, and h is an element in G. We assume that DDH is hard in G. This group is used in the Pseudo-random function (PRF) used in coin generation.

3.4 E-cash Primitives

Having presented the all the building blocks, we shall now enumerate, characterize and describe the main protocols used to implement the E-cash scheme designed by Camenish et. al [3].

BKeygen (1^k,*params*) This algorithm is a key generation algorithm for the bank B. It takes as input the security parameter 1^k and, if the scheme is in the common parameters model, it also takes as input these parameters *params*. This algorithm outputs the key pair (pk_B,sk_B).

UKeygen (1^k,*params*) This algorithm is a key generation algorithm for the user U. Since merchants are a subset of users, they may use this algorithm to obtain keys as well. It takes as input the security parameter 1^k and outputs the key pair (pk_U,sk_U, or (pk_M,sk_M) if used by the Merchant M.

Withdraw (*U*(pk_B,sk_U,*n*), *B*(pk_U,sk_B,*n*)) - In this protocol the user *U* withdraws a wallet *W* of n coins from the bank *B*. The user's output is the wallet *W*, or an error message. *B*'s output is some information T_W which will allow the bank to trace the user should this user double-spend some coin, or an error message. The bank maintains a database *D* for this trace information, to which it enters the record (pk_U, T_W).

Spend (*U*(*W*, pk_M),*M*(sk_M, pk_B, *n*)) - A user *U* gives one of the coins from his wallet *W* to the merchant *M*. Here, the merchant obtains a serial number *S* of the coin, and a proof π of validity of the coin. The user's output is an updated wallet *W′*.

Deposit (*M*(sk_M, *S*, *p*, pk_B), *B*(pk_M, sk_B)) - A merchant *M* sends to bank B a coin (*S*,π=(*R*,*T*,ϕ)). If ϕ verifies and *R* is fresh (i.e. the pair *(S,R)* is not already in the list *L* of spent coins), then *B* accepts the coin for deposit, adds (*S*,π) to the list *L* of spent coins, and credits the account of pk_M; otherwise, *B* sends *M* an error message.

Identify (*params*, *S*, π_1, π_1) - This algorithm allows to identify double-spenders using a serial number *S* and two proofs of validity of this coin, π_1 and π_2, possibly submitted by malicious merchants. This algorithm outputs a public key pk_U and a proof Π_G.

VerifyGuilt (*params*,S,pk_U,Π_G): Parse Π_G as (π_1,π_2) and each π_i as (R_i,T_i,ϕ_i). Run *Identify*(*params*, *S*, π_1, π_1) and compare the first part of its output to the public key pk_U given as input. Check that the values match. Next, verify each ϕ_i with respect to (*S*,R_i,T_i). If all checks pass, accept; otherwise, reject.

3.5 Pseudo-random Function

Another of the fundamental building blocks of the proposed E-Cash system are the pseudo-random functions proposed by Dodis and Yampolskiy [10]. Their construction is the following:

- For every *n*, a function $f \in F_n$ is defined by the tuple (*G,q,g,s*), where *G* is a group of order *q*, *q* is an *n*-bit prime, *g* is a generator of *G* and *s* is a seed (i.e. a random element) in $\mathbb{Z}_q$. For any input $x \in \mathbb{Z}_q$ (except for *x*=-1 mod *q*), the function $f_{G,q,g,s}()$, which we simply denote as $f^{DY}_{g,s}$(ï¿½) for fixed values of (*G,q,g*), is defined as:

$$f_{g,s}^{DY}(x) = g^{1/(s+x+1)} \tag{3.1}$$

This construction is secure under the y-DDHI assumption in G. [17]

3.6 CL Signatures

In order for the E-Cash system to accomplish the identification of double-spenders without comprimising the privacy of legitimate users, Camenisch et al. devised a secure signature scheme based in the Pedersen commitment scheme. This Pedersen commitment requires that, in order to commit the values $(v_1, ...v_m) \in \mathbb{Z}_q^m$, one should pick a random $r \in \mathbb{Z}_q$, define g_0 and g_i as the generators of group G and set:

$$C = PedCom(v_1, ..., v_m; r) = g_0^r \prod_{i=1}^{m} g_i^{vi} \tag{3.2}$$

This effort by Camenisch and Lysyanskaya resulted in a secure signature scheme with two protocols:

- An efficient *signature* protocol between a user and a signer with keys (pk_S, sk_S). The common input consists of pk_S and C, a Pedersen commitment. The user's secret input is the set of values $(v_1, ..., v_l; r)$ such that $C = PedCom(v_1, ..., v_l, r)$. As a result of this protocol, the user obtains a signature $\omega_{pk_S}(v_1, ..., v_l, r)$ in his committed values, while the signer does not learn anything about them. The signature has size $O(l \log q)$;
- An efficient *proof of knowledge* of a signature protocol between a user and a verifier. The common inputs are pk_S and a commitment C. The user's private inputs are the vales $(v_1, ..., v_l; r)$ and $\omega_{pk_S}(v_1, ..., v_l, r)$ such that $C = PedCom(v_1, ..., v_l, r)$.

These signatures are secure under the strong RSA assumption. [17]

3.7 Coin Generation

As mentioned before, one of the advantages of the E-Cash scheme used in our solution lies in its efficiency. As such, one of the goals of the scheme is to adapt single-use electronic cash schemes so that a coin can be used at most 2^l times. The trivial solution would be to obtain 2^l coins. However, this would be inefficient since 2^l may be quite large (e.g., 1024), therefore undermining the performance,of the scheme.

The idea underlying our system is that the values s and t implicitly define several (pseudo-random) serial numbers S_i and blinding values B_i, respectively. In other words, we need a pseudo-random function $F_{(.)}$ such that we can set:

$$S_i = F_s(i) \tag{3.3}$$

$$B_i = F_t(i), 0 \leq i \leq 2^l - 1 \tag{3.4}$$

The user then gets 2^l pseudo-random serial numbers with the corresponding double-spending equations defined by (s, t). Here, the double-spending equation for coin i is:

$$T_i = g^{sk_U}(B_i)^R \tag{3.5}$$

With R being chosen by the merchant. This leaves us with a very specific technical problem. The challenge is to find a pseudo-random function such that, given a commitment to (skU, s, t), a commitment to i and the values S_i and T_i, the user can efficiently prove that she derived the values S_i and T_i, correctly from sk_U, s, and t, i.e. $S_i = F_s(i)$ and $T_i = g^{sk_U}$.

3.8 E-cash Protocols

3.8.1 Withdraw

A user *U* interacts with the bank *B* as follows:

- *U* identifies himself to the bank *B* by proving knowledge of sk_U;
- The user and the bank In this step, the user and bank contribute randomness to the wallet secret s; the user also selects a wallet secret t. This is done as follows: U selects random values s', t $\in \mathbb{Z}_q$ and sends a commitment **A'**= PedCom(sk_U, *s',t;r*) to *B*. *B* sends a random *r'* $\in \mathbb{Z}_q$. Then *U* sets *s=s'+r'*. *U* and *B* locally compute $\mathbb{A} = \eth^{\backprime}\mathbb{A}'$ = PedCom(sk_U,*s'+r',t;r*) = PedCom(sk_U,*s,t;r*);
- *U* saves the wallet W = (sk_U,*s,t,*σ_B*(*sk_U*,s,t),J*), where *s,t* are the wallet secrets, σ_B(sk_U,s,t) is the signature of the bank and J is an *l*-bit coin counter initialized to zero;
- *B* records a debit of 2^l coins for an account pk_U.

3.8.2 Spend

A user *U* interacts with the merchant *M* as follows:

- *U* identifies himself to the merchant *M* by proving knowledge of sk_U;
- *M* chooses a random *R* $\in \mathbb{Z}_q$ such that R$\neq$0 and sends it to U;
- If *J*$>J_0$+J_1,the user aborts as all the coins have already been sent;
- The user computes the serial number S = $f^{DY}_{g,s}(J)$ and a (now fixed) security tag *T* = $pk_U f^{DY}_{g,s}(J)^R$; With these values, user computes ZPKPOKϕ and sends it to the merchant;
- If ϕ verifies, *M* accepts the coin (*S,(R,T,*phi*)*) and uses this information at deposit time;
- *U* updates his counter *J*=*J*+1. When *J*$> 2^l - 1$, the wallet is empty;

3.8.3 Double-Spending Identification

However efficient the E-Cash scheme is, it must allow one to expose double-spenders to outside third parties in an undeniable fashion. In short, when a User spends a coin more times than allowed he must by identified and this act must be proven to anyone in a sound fashion. The Compact E-Cash scheme by Camenish et al. provides this characteristic, while granting that the user's anonymity is based on *computation*, rather than *trust* assumptions.

Following the steps of the coin generation, if S_i and T_i are computed through the above mentioned constructions they are members of G rather than of $\mathbb{Z}_q$. This leaves us with the following protocol: to withdraw a coin, a user obtains a signature on (sk_U, s, t). During the spending protocol, the user reveals S_i and the result of the double-spending Equation 3.5 where sk_U is the user's secret key and $pk_U = g^{sk_U}$ the corresponding public key. Now, with two double-spending equations for the same coin:

$$T_1 = g^{sk_U} B_i^{R1} \tag{3.6}$$

$$T_2 = g^{sk_U} B_i^{R2} \tag{3.7}$$

We can infer the following:

$$\left(\frac{T_2^{R_1}}{T_1^{R_2}}\right)^{(R_1-R_2)^{-1}} = \left(\frac{g^{uR_1+R_1R_2\alpha}}{g^{uR_2+R_1R_2\alpha}}\right)^{(R1-R2)^{-1}} = g^{\frac{u(R1-R2)}{(R1-R2)}} = g^u = pk_U \tag{3.8}$$

The result of the calculation of the two double-spending equations is pk_U, the public key of the User that has double-spent, sufficient proof for identification of said user.

Chapter 4

Architecture and Implementation

Considering all the previous research, the following presents the proposed solution. The main goal of this work is to offer users an efficient mobile payment system that assures user privacy at all times and that, while designed to be used on an NFC enabled system, it is capable of fitting smoothly with existing systems, offering *interoperability* without the need of costly hardware updates, while at the same time being portable to related systems (e.g. Java Cards) and/or different wireless means (e.g. Wi-Fi, Bluetooth).

We accomplished this through the implementation of the off-line divisible and unlinkable electronic cash scheme, presented in Section 3, embedded in a multi-layered architecture, with independent *Security* and *Communication* modules, as depicted in Fig. 4.1.

This architecture enables implementation in a vast array of platforms, with more or less computing capacities, because these modules can be moved around the system (e.g. in a system with very limited resources, most security related computing can be moved to the back-end structure).

The proposed system considers the three main actors, namely: User, Bank and Merchant. These actors perform distinct roles in the system while sharing the architecture that bonds them together, namely the Security and Communication modules.

The User is able to withdraw a *Wallet* from the Bank, which enables the generation of *Coins* that are consumed through their spending with the Merchant. The Merchant, having received said coins, can then interact with the Bank, depositing them. Lastly, the Bank has the task of accommodating the requests from Users and Merchants, while also being responsible for the identification of double-spenders with non-negligible probability. Both the Bank and the Merchant handle incoming connections in a multi-threaded approach with a dedicated *Clerk* thread, allowing both Agents to stay receptive to new connections.

The Security module is responsible for all the encryption and authentication procedures necessary for the secure functioning of the system. This includes the creation and management of the asymmetric keys of each Agent, the encryption and decryption of messages sent between them, and the authentication procedures that precede each of the interactions.

The Communication module handles the establishment of the connection between Agents and the

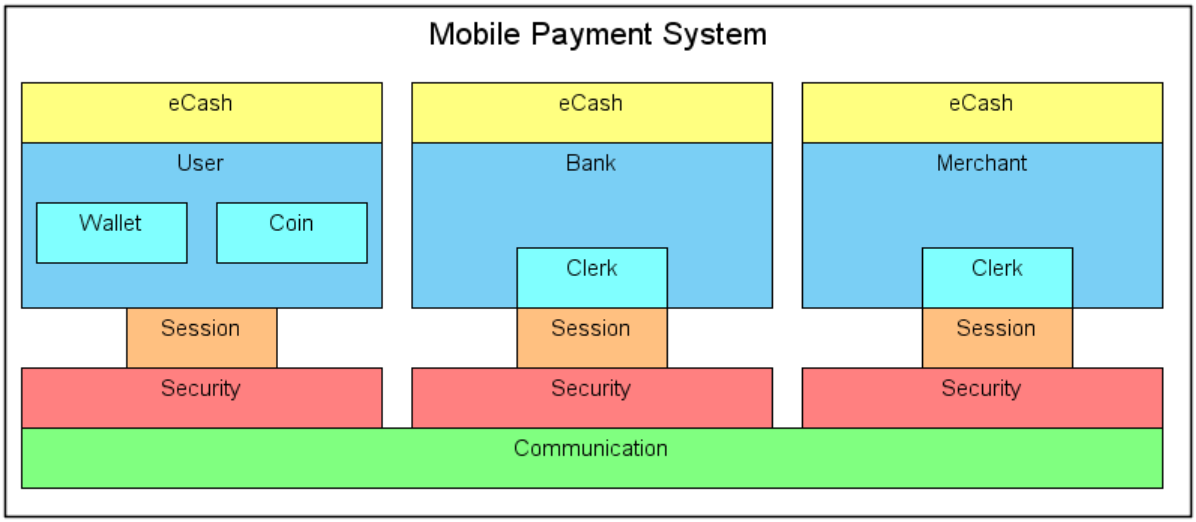


Figure 4.1: Overview of System Architecture

subsequent transmission of messages, in the most transparent way possible. The details of these connections are abstracted in a *Session* object, allowing for a change in the protocol and medium used without any disturbance to the layers above.

Lastly, the E-Cash module comprehends the operations necessary to the implementation of the compact E-Cash presented in Section 3, namely the Pedersen Commitment, the CL signature and the Pseudo Random Function.

According to the above motivations, the solution was implemented in Java, a programming language notorious for its portability. In our implementation each Actor and Module corresponds to a Java class. In this section we characterize the functionalities and responsibilities of each of these and how they were implemented, followed by a detailed description of the protocols that mediate the communication between Actors.

4.1 Implementation

4.1.1 User

The User implementation, depicted in Fig. 4.2, features two operations: *withdraw*, through which the User connects to the Bank and accomplishes the withdrawal (i.e. generation) of a *Wallet* object; and *spend*, where, the User interacts with the Merchant, using *Coins* generated through the *Wallet*.

The *Wallet* features the attributes defined in the E-Cash protocol [3], namely:

- *skU* - The User's private key;
- *t* - A wallet secret generated by the User;
- *s* - A wallet secret resulting of the combination of two secret generated by User and Bank;
- *j* - An *l* bit coin counter. This was chosen to be represented by the *short* data type that allow for

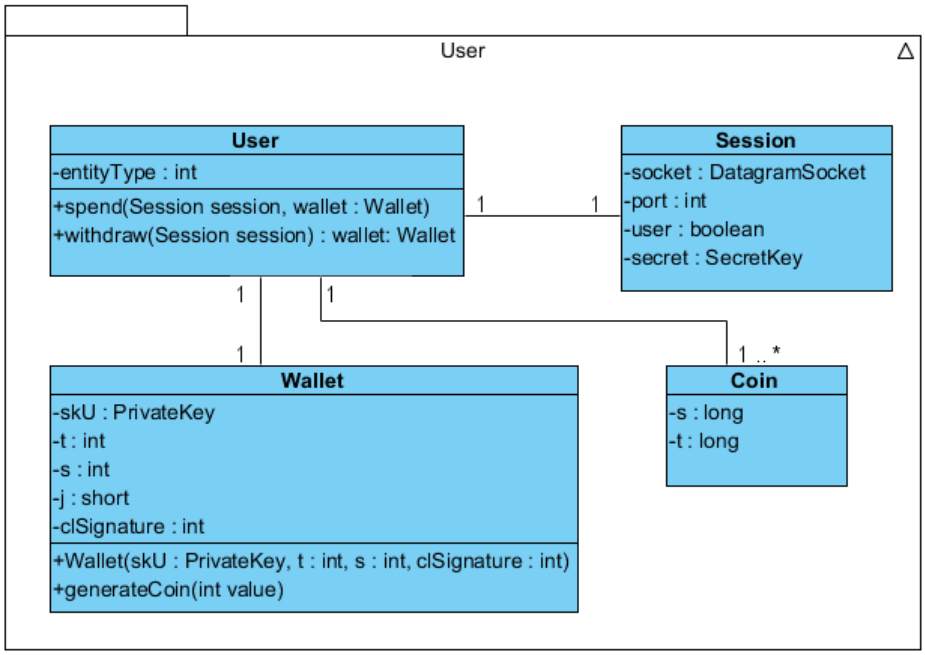


Figure 4.2: Diagram of the User implementation

a 2^4 bit counter with a maximum value of 32.767, sufficient for its purpose and to enabling the efficiency and compactness characteristics of the E-Cash scheme;

- *clSignature* - A signature generated in the withdraw protocol;

The *Coins* are generated by the method *generateCoin*, part of the *Wallet* object, and are characterized by, S and T, two numbers that are a product of that process.

In order to abstract the Communication details, the User relies on a *Session* object, used as a parameter on both spend and withdraw protocols. Lastly, the *entityType* attribute is used to define the role of the Agent.

4.1.2 Bank

According to the implemented E-Cash protocol, the Bank is realized through the structure illustrated in Fig. 4.3 and allows for a number of operations.

- *Withdraw* - Where the Bank, at the request of a User, generates a Wallet. The number of coins present in the *Wallet* is debited frothe respective User account.
- *Deposit* - The Bank receives a *Coin* sent by a Merchant, saving it in the *spentCoins* list, in order to be able to detect a future double-spending of said coin.
- *Identify* - Ran when a double-spend situation is detected (i.e. the Bank receives a Coin that was already present in the *spentCoins* list), this method allows for the identification of the Public Key (*pkU*) of the User that double-spent. Receiving two Coins (S, R, T) as input, the Bank then computes the Double Spending equation (3.8), outputting the Public Key of the User that double-spent.

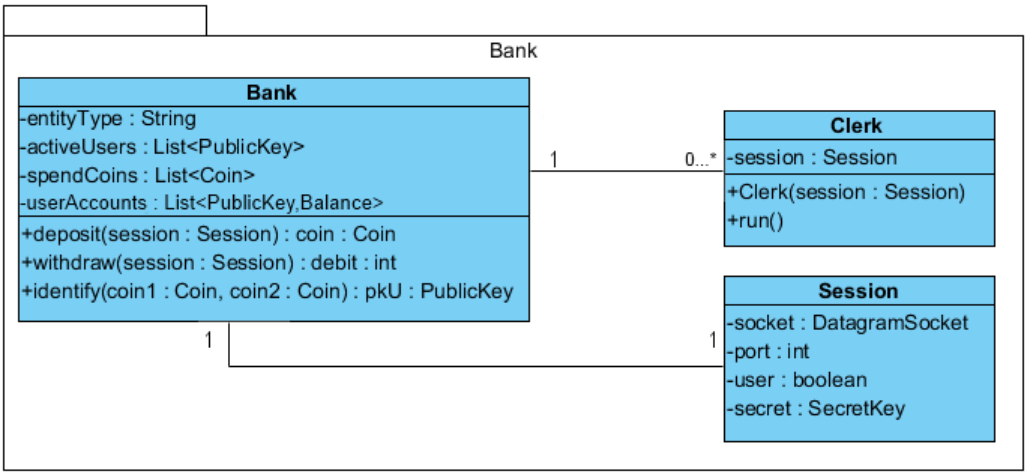


Figure 4.3: Diagram of the Bank implementation

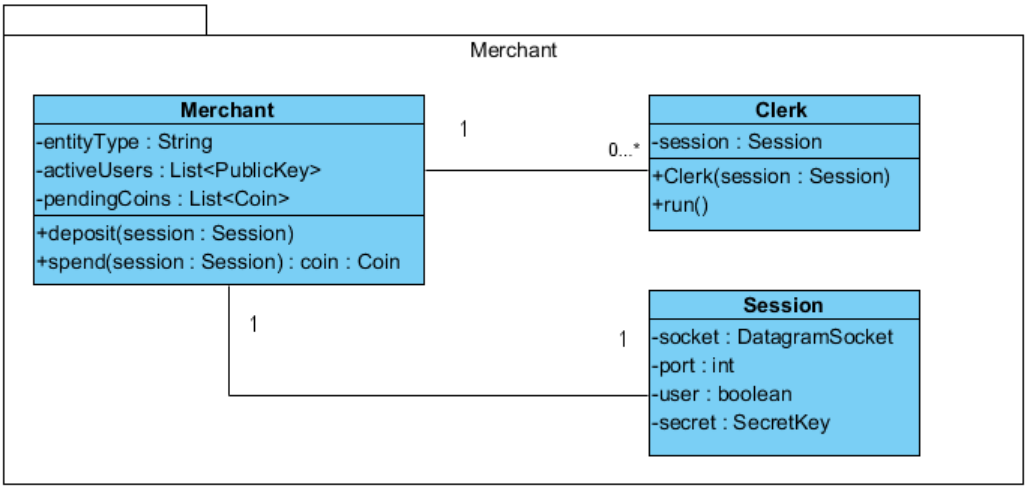


Figure 4.4: Diagram of the Merchant implementation

Similarly to what happens in the User, the *Session* is used to abstract the connection details. However, while the User relies on single threaded request operations, the Bank must be able to offer a multi-threaded response to incoming connections. This is accomplished with resort to a *Clerk* object. As soon as a *Session* is generated (i.e. a connection is established), the Bank starts a new and independent *Clerk* thread, which is bonded to the *Session* passed as an argument to its constructor.

4.1.3 Merchant

As far as the structure of the implementation (depicted in Fig. 4.4) goes, the Merchant is very similar to the Bank, featuring a *Session* and a *Clerk*, with the same characteristics mentioned before. The Merchant then provides two operations:

- *Spend* - Where the User gives one of the coins from his Wallet to the Merchant. The output of this function is a coin, that is then added to the *pendingCoins* list;
- *Deposit* - Executed immediately after the Merchant receives a Coin from the user, this method

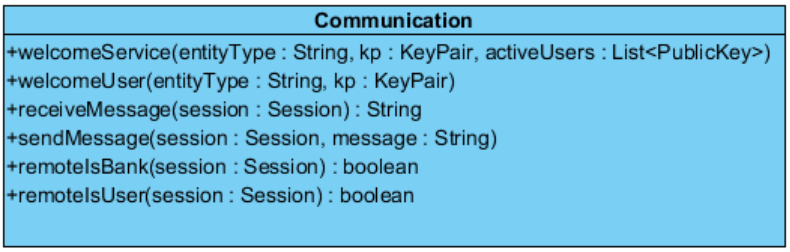


Figure 4.5: Diagram of the Communication Module implementation

allows for the deposit of this Coin into the Bank. If the Bank is unavailable, the Coin is stored in the *pendingCoins* list, to be deposited in the future. Vice versa, if the deposit is successful, the Merchant then proceeds to deposit any coins previously stored in the list.;

4.1.4 Communication Module

The Communication Module (Fig. 4.5) mediates all the contact between the different Agents.

These transmissions begin with he initial handshaking and connection establishment between Services (i.e. the Bank or the Merchant) and Users (i.e. the User, or the Merchant when connecting to the Bank), with each side of that procedure fulfilled, respectively, by the *welcomeService* and *welcomeUser* methods. We detail the protocol that regulates this procedure in Section 4.2.1.

Next, we have the *receiveMessage* and *sendMessage* methods, used in the transmission of messages. While the methods themselves are not transparent to the used medium, as it is necessary to resort to primitive methods that are dependent on said medium, these details are abstracted to the Agents thanks to the *Session* object. These methods resort, respectively, to the *decryptMessage* and to *encryptMessage* methods that are contained in the Security module in order to safely receive and transmit the message.

Lastly, the *remoteIsBank* and *remoteIsUser* also reflect this abstract approach. Used after the connection is established through the *welcomeService* and *welcomeUser* services, by the User and the Bank, respectively, in order to ascertain the role of the Agent that they are connected to. The User might be connected to the Bank (for a withdraw) or the Merchant (to spend a coin). Likewise, the Bank might be receiving a connection from the User (for a withdraw) or the Merchant (for a deposit).

4.1.5 Security Module

This module, depicted in Fig. 4.6 is responsible for all security related matters, namely the generation of the asymmetric keys and handling of the cryptography of messages sent in the various protocols. Also a component of the security module is the authentication of Agents in the face of the Services they will be using.

The *keyGeneration* method is used to generate the Asymmetric Keys on which our solution is based. This is done using the *KeyPairGenerator* primitive supplied by Java. The Key Generating algorithm used is *RSA* while recurring to the *SecureRandom* primitive, supplied with the *SHA1PRNG* algorithm,

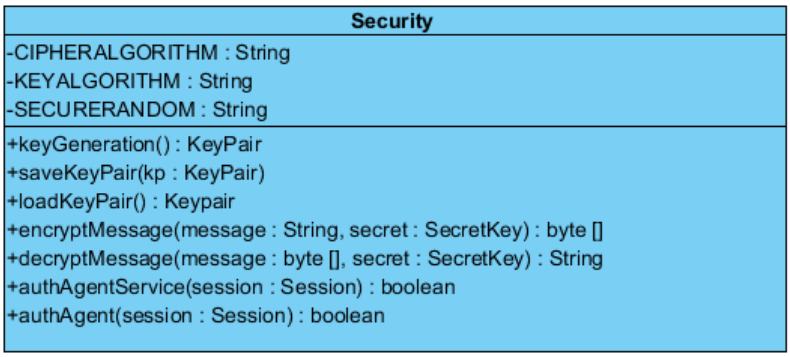


Figure 4.6: Diagram of the Security Module implementation

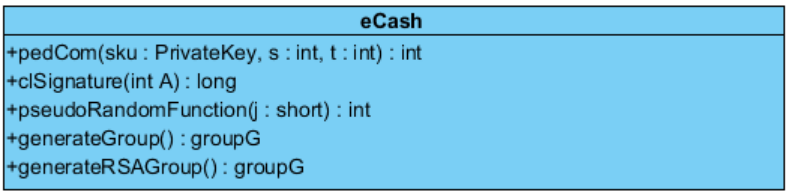


Figure 4.7: Diagram of the E-Cash Module implementation

to provide adequate and secure randomness to the key creation.

The generated Keys are saved and loaded through the *saveKeyPair* and *loadKeyPair* methods using the *java.security.KeyStore* class, using the *JCEKS* instance, implemented by the *SunJCE* provider, storing the Keys in the *JCE* SealedObject format.

The messages are encrypted and decrypted by the *encryptMessage* and *decryptMessage* operations. This is accomplished through the *javax.crypto.Cipher* library, using the *RSA/ECB/PKCS1Padding* transformation.

Lastly, the *authAgentService* and *authAgent* methods implement the Challenge-response mechanism used to authenticate the Agents as part of the connection establishment procedure. This Challenge-response accomplishes two very important goals, authenticating the Users, by proving that its Secret Key is a resultant of the Public Key sent to the Service, while also ensuring the freshness of the connection.

4.1.6 E-Cash Module

Finally, the E-Cash module (Fig. 4.7) realizes all the operations necessary for the implementation of the Compact E-Cash module defined by Camenish et al.[3], namely the Pedersen Commitment, the CL signature, the Pseudo Random Function and the Group generators. These are essentially the implementation of the mathematical functions defined in [3] that have been described in Chapter 3, using the operators defined in the *java.lang.Math* package. However, some requirements and specificities for each should be noted.

generateGroup This method is used to generate a group $G = \langle g \rangle$ that will be used in the Pseudo Random Function procedure. This group, also known as a *Schnorr group*, obliges to the following

specification:

- p and q denote large primes such that q divides $p-1$, G_q is the unique subgroup of $\mathbb{Z}_q^*$ of order q, and g is a generator of G_q.

Considering this, the workings of the *generateGroup* method are trivial. Firstly, *p* is generated by using the *BigInteger java.math.BigInteger.probablePrime* library. Then, we generate a number of *q* random primes, until one satisfies $p \mod q = 1$, i.e. $(p-1)\%q = 0$.

Improving on this, since $(p-1)\%q = 0 \equiv p = rq+1$, we simply increment the multiplication factor r and test the resulting value of p with java.math.BigInteger.isProbablePrime method, asserting the number is prime within a $(1 - 1/2^{50})$ probability. By doing this, we circumvent the very heavy prime generation, allowing for a faster result with significantly less computation.

The final step in this procedure is getting the generator g. For this, we can just use any random integer modulo p, since the probability that q is *not* a divisor of the order of a random non-zero integer modulo p if $1/q$ is so small that it should never be hit in practice. [1] Nevertheless we still test the generator. The procedure then follows: we start by creating a random integer u modulo p and compute $g = u^{(p-1)/q} \mod p$. If $g = 1$, a new value of u must be chosen and the procedure restarted. Otherwise, it can be shown that g has order exactly q, and thus u is a good generator for this group.

Due to the fact that this procedure is, still, computationally intense, *generateGroup* is run alongside the *keyGeneration* method in the first time the User starts the application, generating a *Group* object that is stored locally. This relegates this computation for the first initialization of the application, allowing for a fluid experience in all later uses.

generateRSAGroup The generation of the RSA groug, while being based in the same principles, has some significant differences, compared to the previously described Group generation. The groups generated by it must follow:

- **g** is a quadratic residue modulo *n*, with the latter being a special RSA modulus of *2k* bits. An RSA modulus $n = pq$ is called special if $p = 2p' + 1$ and $q = 2q' + 1$ where p' and q' are also prime numbers.

On a first though, this could be attained by generating a prime number p' with $2k/(2-1)$ bits (because the length of $n = pq$ must be 2k bits) in an analogous way to the calculations on *generateGroup*. This would go on until we found a prime number that satisfied said condition, or if p' got longer than $2k/(2-1)$, forcing for the generation of another random prime and the restart of the process. This would then be repeated for q'.

Improving on this, we firstly compute p on the above mentioned way disregarding the $2k/(2-1)$ bit limit. After p is reached, we compute q with $2k - p$ bits. This improves the algorithm not only by eliminating a possible great number of costly random prime number calculations, but also due to the fact that q becomes smaller, which translates in a reduced computing cost.

[1] The overall security of the Diffie-Hellman protocol relies on the practical impossibility of obtaining events which substantially more probable than this.

Following comes the calculation of the generator, for which we use the same procedure described in the *generateGroup* method. However, for the usage of this RSA group (CL Signatures), a group with multiple generators is required. In order to get to these, we generate random prime numbers and ascertain if their are relatively primes to g by computing the value of the *Greatest Common Divisor*(GCD) between the two numbers. If the value of the CGD equals 1, then the random prime number under testing is also a generator of the RSA Group.

pseudoRandomFunction This method receives the 2^8 bit counter [2] from a Wallet and is responsible for the generation of a pseudo random number that will be used in the creation of Coins, upon the execution of the *Spend* protocol. The function used to accomplish this is *f*$(J) = g^{1/(s+j+1)}$. *g* a generator of *G* and s, a seed in is a seed in $\mathbb{Z}_q$ are obtained through the *generateGroup* method. J is the serial number of the key for which said random value is required.

clSignature This method implements the Pedersen Commitment used as a commitment for the random values generated on occasion of the *Withdraw* protocol. The formula used for this is$C = g_0^r \prod_{i=1}^{m} g_i^{vi}$, with $(g_0]..., g_m)$ being generators of a special RSA modulus group, r being an element of said group and $(v_0, ..., v_m)$ as the committed values. As input, this method receives the Private Key of the User making the withdrawal and two random values calculated in this process.

The CL signature protocol by Camenish and Lysyanskaya [36] is implemented through the *pedCom* method. Used on occasion of the *Withdraw* protocol, this method is used as an efficient proof of knowledge of the signature protocol between User and Bank in the withdrawal process. The function used is exactly the same of the *clSignature* method, but this time around the input is the result from that previous commitment, with $(g_0, ..., g_m)$ as generators of a special RSA modulus group, r being an element of said group and $(v_0, ..., v_m)$ and $\sigma_{pk_S}(v_0, ..., v_m)$ being the inputs.

4.2 Protocols

4.2.1 Connection Establishment

The establishment of the connection itself is accomplished by two distinct methods, named *welcomeService* and *welcomeUser*, used, respectively, by the Services and the Users. These follow an *active communication model*, in which the Services passively wait for an User to connect. The following describes the procedures of this protocol, depicted in Fig. 4.8.

1. Independently of the role (i.e. Service or User), the system starts by creating a Key Pair using the *keyGeneration* operation in the Security module, or loading if it had already been previously created through *loadKeyPair*. The Service then stands on a passive wait for the connection of an User.

[2] The 2^8 limit value for the bit counter is due to the choice of using the Java *byte* data type, the smallest integer data type in Java, sufficient to hold the necessary amount of tokens.

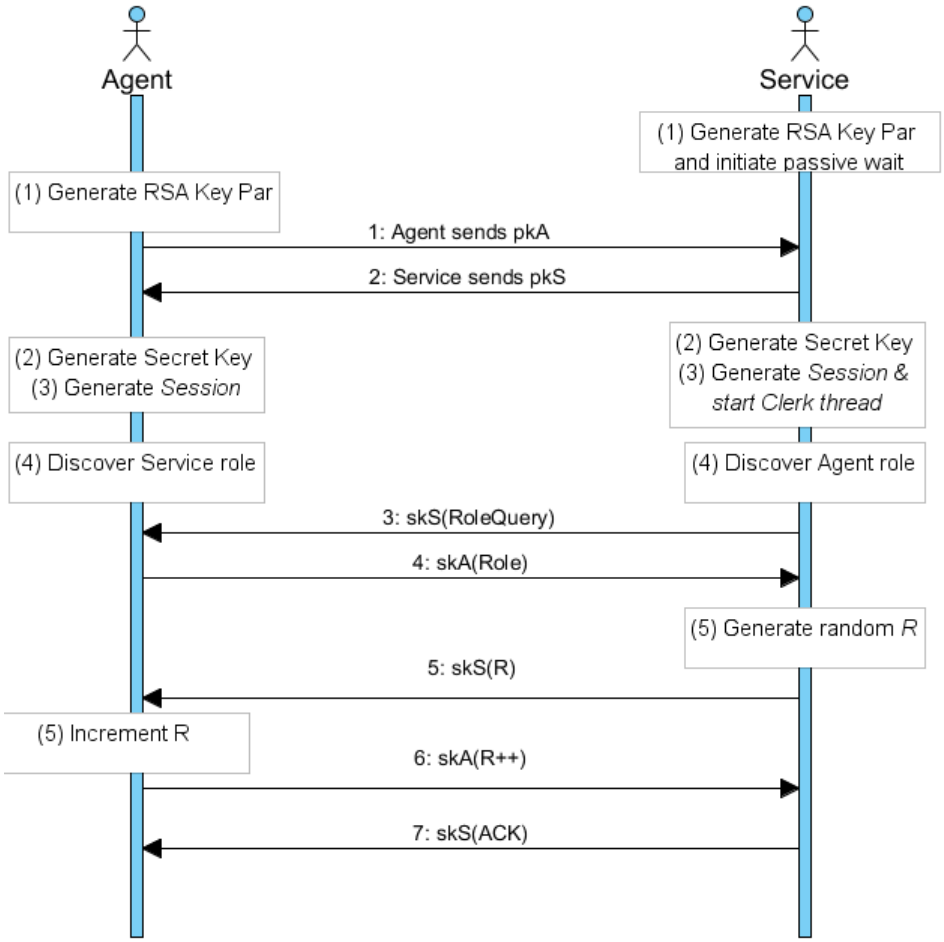


Figure 4.8: Overview of Connection Establishment

When an User decides to initiate a connection a search is conducted in order to determine which Service is available for connection. This is accomplished in the prototype through the search of port ranges in the Service for one that is waiting for a connection. While not strictly necessary, this was done in order to replicate the behavior of RF systems, allowing for a simulation of the physical approach between initiator and target.

As a Service is found, the initial handshake and establishment of the connection must be negotiated with distinct primitives dependent on the interface used. As such, this component of the protocol will always have to be adapted as the solution is ported. In the presented prototype this translates into the establishment of an *UDP* connection, using the *DatagramSocket* primitives provided by Java.

The process thereon follows the *Diffie-Hellman Key Agreement Algorithm* [9]. Through this algorithm two users can exchange a secret key over an insecure medium without any prior secrets. This is appropriate, seeing the exposed nature of the RF communication fields and the considerations about these made previously in this work.

2. The User starts the communication with the desired Service, by sending its Public Key (message 1 on the diagram), to which the service responds in a likewise form (message 2). Using the *KeyAgreement* Java primitive, both parts then initialize the Diffie-Hellman protocol by supplying their own Private Key to the *init* method and the Public Key of other party to the *doPhase* method. With these arguments, the *generateSecret* method is then used, recurring to the *HmacMD5* algorithm, in order for both parties to independently obtain the *SecretKey* which is then used to cypher all communications.

It is important to note, however, that the purpose of the Key Pair is not limited to the generation of the secret. It is, in fact, fundamental for the operation of our solution, since it is through them that eventual double spenders are identified.

3. When the connection is established, a distinct *Session* object is created by each party. In this object we store all the details essential to the communication in the interface (in this case, the address of the other party and the port used). The Service then starts a dedicated *Clerk* thread, to which the *Session* object is provided.

4. At this point, both Service and User must ascertain the role of each other, doing so through the *remoteIsBank* and *remoteIsUser* methods. In the presented solution the *remoteIsUser* works by directly asking the User for its role (through messages 3 and 4), while the *remoteIsBank* is based in the ascertainment of the port on which the communication is taking place. In a real world implementation of this system a different resolution would be necessary, due to security reasons. One of such resolutions could be accomplished by maintaining a list of Services and comparing the received Public Key to the ones present in said list.

5. Lastly, the Agent must answer to a Challenge-response sequence implemented in the *authAgentService* and *authAgent* methods, ran, respectively, by the Service and User. The Service starts by generating a 24 bit long random number R, using the methods defined in *java.util.Random*

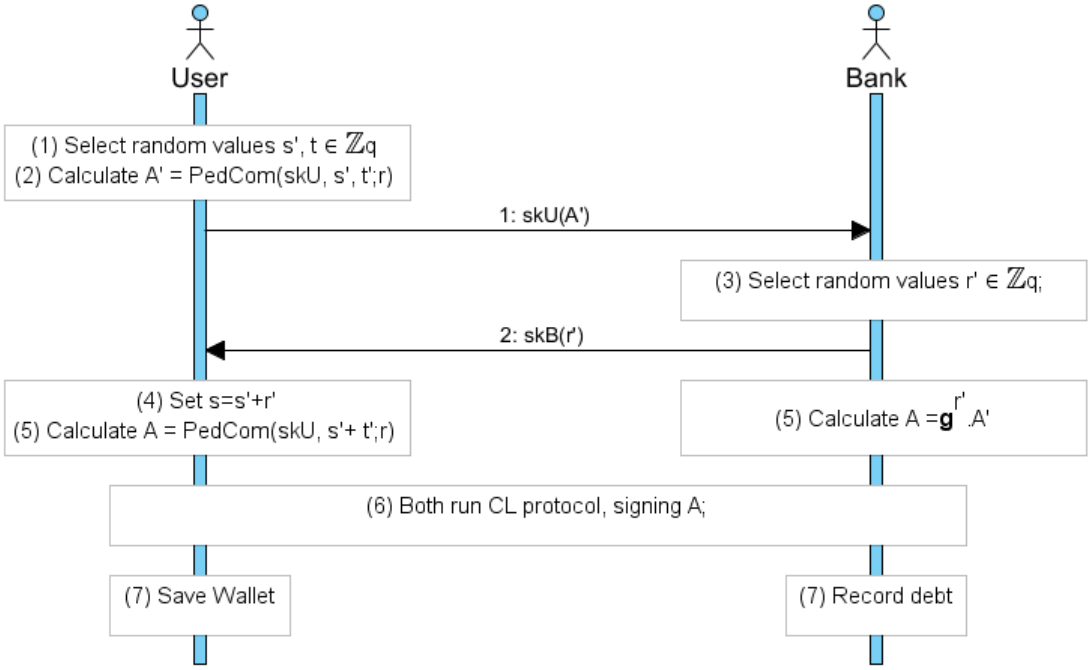


Figure 4.9: Overview of Withdraw Protocol

and sending the message to the User (message 5), using the aforementioned encryption methods. The User receives and decrypts this message, proceeding to increment it by a previously set amount and forwarding the result to the Service (message 6). The Service then verifies if the result is according to the expected, allowing the protocol to follow suite if it does and notifying the User accordingly (message 7).

4.2.2 Withdraw

In this protocol the the User withdraws a wallet from the Bank. A depiction of the various steps can be seen in Fig. 4.9 and we shall now provide a detailed explanation of its workings.

1. With the connection already established, this protocol begins with the User selecting two random values (s' and t) from the previously calculated *Group* object;
2. The User then runs the Pedersen Commitment method, *pedCom* from the E-Cash module, taking as input its private key sk_U and the s' and t values, obtaining A'. The User then sends this value to the Bank (message 1.), encoding it with the Secret Key obtained from the Diffie-Hellman algorithm s_K;
3. At this point, the Bank also contributes randomness, selecting a random r' from its own *Group* object, sending it to the User (message 2.) also through Secret Key encoding;
4. The User sets $s = s' + r'$;
5. At this point two distinct operations occur: the User calculates A by running the Pedersen Commitment method with input (sk_U, $s' + t'$); and the Bank calculates $A = \mathbf{g}^{\mathbf{r}'} A'$; Doing this, both parties

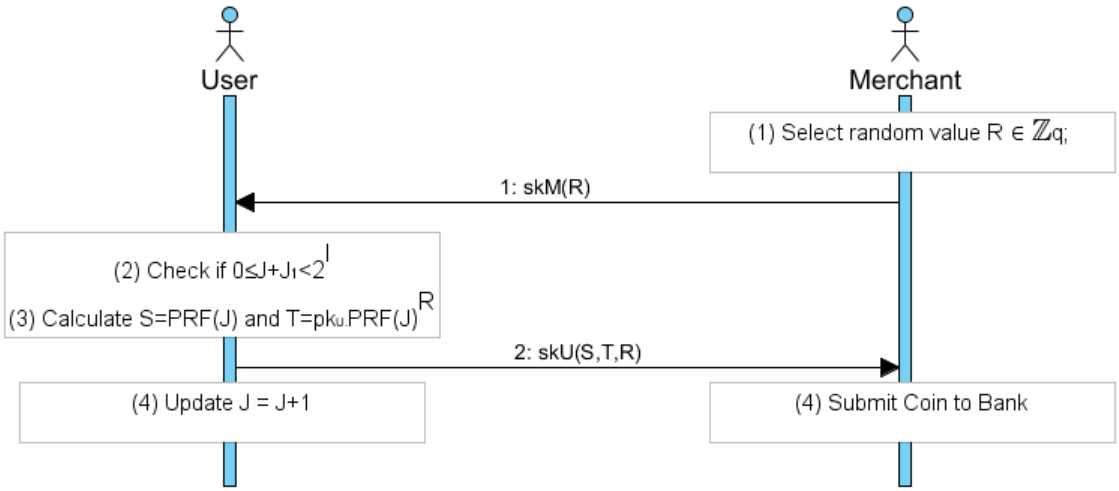


Figure 4.10: Overview of Spend Protocol

reach A, without the need for the User to explicitly share s' and t;

6. Both parties then run the *clSignature* method with A as the argument, obtaining $\sigma_B(sk_U, s, t)$;

7. Finally, the user saves the obtained Wallet $W = (sk_U, s, t, \sigma_B(sk_U, s, t), J)$ by creating an instance of a *Wallet* object, while the Bank records a debit in the User account;

4.2.3 Spend

After a User has finished a withdrawal, obtaining a Wallet, it is then possible to connect with Merchants to spend the coins in said Wallet. The spending protocol is detailed in Fig. 4.10 which shall now be expounded:

1. With the connection already established, this protocol starts by with the Merchant selecting a random R value from its *Group* object and sending it to the User, encoded with the Diffie-Hellman provided Secret Key (message 1);

2. The User then checks the Coin counter. If the value of the Coin that is going to be spend J_1 added to the present value of the counter J obliges to $0 = J + J_1 < 2^l$ then the operation is authorized. If not, it is aborted;

3. The User now calculates the values of S and T, through the *pseudoRandomFuncion* present in the E-Cash module. The input for each calculation is, respectively, J and J^R; After this calculation, the User generates the *Coin* object, with the values of S, T and R, and sends it to the Merchant, encoded with the Secret Key (message 2);

4. With the protocol completed, the User updates the value of the Coin counter J and the Merchant starts the *Deposit* protocol, submitting the received coin to the Bank;

4.2.4 Deposit and Double-Spending Identification

After receiving the Coin from the User, the Merchant tries to connect with the Bank in order to start the *Deposit* procedure. If this attempt is successful, the Coin is stored in the *pendingCoins* list, held by the Merchant, waiting for a future opportunity. If it does succeed, the Merchant then initiates the *Deposit* protocol, using a FIFO strategy with the *pendingCoins* list.

The *Deposit* protocol is very much straightforward, in the sense that it merely resumes to the Merchant sending a Coin to the Bank. The Bank, upon receiving the Coin, checks its own *spentCoins* list for the presence of the now received coin. If the Coin is fresh (i.e. not present in the list) the Bank accepts the coin for deposit and adds the coin to the list. If the coin has appeared before, the *Identify* algorithm is triggered.

In this situation, the Bank runs the *identify* method, using both coins (the one on the list and the one received from the merchant), which outputs the Public Key of the fraudulent user.

Chapter 5

Results

In order to infer the feasibility and usability of the proposed solution a prototype version was implemented in Java. This prototype was then evaluated using a Smartphone, holding the User's side application, and Laptop computer, running the Bank and the Merchant. The general characteristics of both these devices are described in Section 5.1. This assessment comprehends both an analysis of the resource and time consuming initialization of the system in Section 5.2, including the generation of the RSA Key Pair necessary to the establishment of a secure communication channel and the Group generation required for the operation of the E-Cash system, followed by diverse takes on the Withdraw, Deposit and Spending protocols in Section 5.3. The obtained results are then analyzed in order to evaluate the performance of the implemented E-Cash system.

In Section 5.4 we extrapolate from this analysis into considerations on the Portability of the implemented E-Cash system. This includes an overview on how different hardware (i.e. Smartphones, Java Cards and RFID Systems) and Wireless technologies (i.e. Wi-Fi, Bluetooth and NFC) would be able to cope with the porting of the proposed system.

Later on we go through the assessment on user expectation for a Mobile payment system in Section 5.5 and some Interface considerations based on our proof of concept implementation in Section 5.6.

5.1 Testing Environment

Our testing environment comprehends both a Smartphone and a Laptop computer, with their characteristics detailed in the Table 5.1.

Device	**Nokia N900**	**Laptop Computer**
CPU	900 MHz Cortex-A8	2.53GHz Intel Core 2 Duo T9400
RAM	256MB	4096MB
OS	Maemo 5 (Linux based)	Windows 7

Table 5.1: Proof of Concept System

The Smartphone used was a Nokia N900, once a flagship device of the Finnish company, released in 2009. Running a Linux based operating system, this device shows its age when compared to more recent devices, holding a single core CPU and a limited amount of RAM, while even midrange devices released more recently feature more than one core and hold, at least, double the amount of memory. We can therefore consider that the results obtained in this device are a worst-case scenario, which holds our considerations on user experience from the obtained results.

The laptop is used to run both the Bank and Merchant. A full scale implementation of this system would feature a full scale Back Office, with dedicated resources and services way beyond those of a commercial laptop. Even so, the results obtained in our tests reflect the difference in resources between a mobile device and a dedicated system.

The results herein presented were obtained as a result of 10.000 iterations of the same calculation (unless otherwise noted), in order to discard disturbances from other running processes, while, specifically in the Smartphone, reflecting the resources that an actual implementation would have at its disposal.

5.2 System Initialization

This section presents the results of the performance tests ran on the various processes required in the initialization of the Mobile payment system. These operations comprehend the Key Pair generation (Section 5.2.1), the Group generation (Section 5.2.2) and the RSA Group generation (Section 5.2.3).

5.2.1 Key Pair Generation

Firstly we start by evaluating the performance of the Key Pair generation. This procedure comes before any other execution as it is required for the establishment of a secure communication channel. As detailed in Section 4.1.5, our system uses an RSA Asymmetric Key Pair, generated via the KeyPairGenerator primitive supplied by Java.

We tested the generation of Key Pairs for Key sizes varying from 512 to 2048 bits.

The results obtained in the PC, depicted in Figure 5.1 suggests an average time for the generation of the 2048 bit Key Pair of 0,7 seconds, a value significantly higher to the 0,01 and 0,08 seconds for 512 and 1024 bit, respectively. However, bearing in mind that this Key Generation process would be executed in a Back Office, featuring significantly more resources than a simple Laptop computer, this translates into the differences between the different Key sizes to become intangible. The largest Key Size considered (2048 bits) would then be the preferable option.

One hinder to this option could be prompted by the existence of less powerful Back Offices for Merchants. However, since this Key Generation would be ran only once, or at considerable time intervals, this is a neglectable drawback as proper scheduling of the process would make it transparent to users.

On the Smartphone, as expected, the performance is reduced as a reflection of the limited available resources. A 512 Key size translated to 0,55 seconds generation time, with 1024 and 2048 Key Pairs

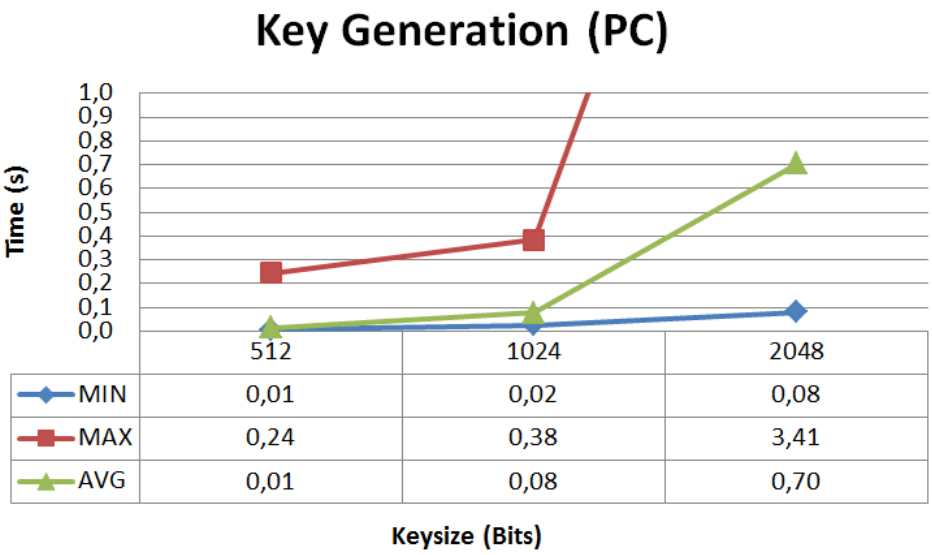


Figure 5.1: Key Pair Generation in PC

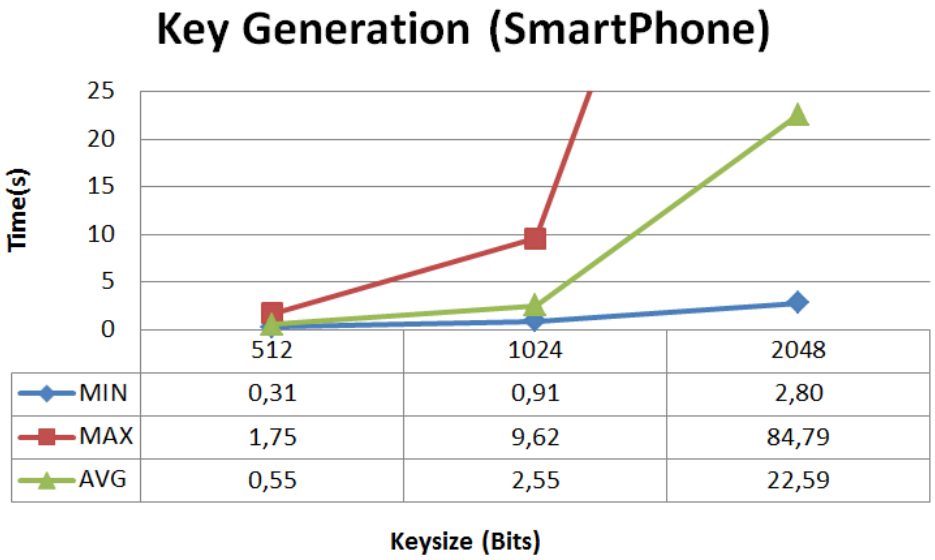


Figure 5.2: Key Pair Generation in Smartphone

being generated in 2,55 and 22,59 seconds, respectively.

In terms of user experience, the 0,55 seconds needed to generate the 512 bit Key Pair would be an acceptable value. Eventually, however, one could use this smaller Key size for the in the first time the mobile system is executed in the system, with a 1024 bit Key being generated as a background process, thus improving security in a transparent way to the user and without impairing system usability.

5.2.2 Group Generation

We shall now evaluate the performance values of the Group Generation procedure. With its generation described in Section 4.1.6 the Group is a primary building block for our proposed E-cash system, having, likewise to the Key Pair, to be computed before any other protocol takes place.

This procedure was evaluated for varying modulus length values. These values contribute to the

randomness factor of the values generated in the various protocols, therefore for the scalability of the system higher Modulus length are better as the possibility for conflicts is reduced.

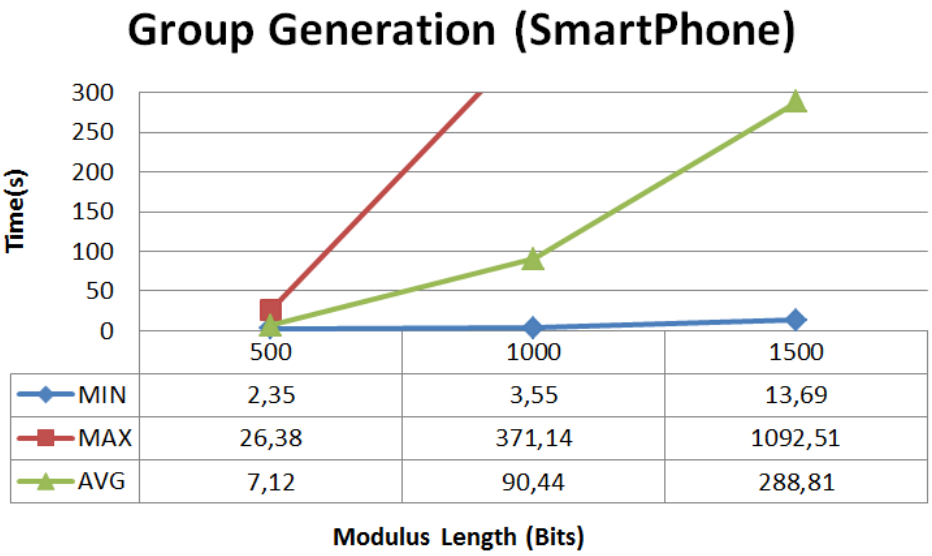


Figure 5.3: Group Generation in Smartphone

On the Smartphone, the results show an average value of 7,12 seconds for a Modulus Length of 500 bits, which is already a considerable delay in terms of usability. The following values, 90 seconds for 1000 bits and 288 seconds (5 minutes) for 1500 bits are simply untenable for a mobile application. Given this resuts, we conclude that it would be difficultly be feasible to perform this calculation on a hand-held device.

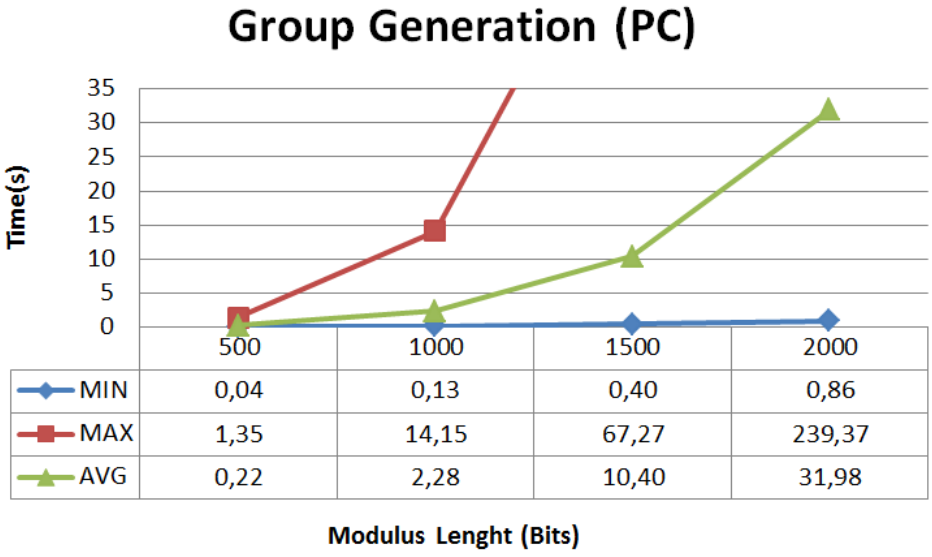


Figure 5.4: Group Generation in PC

The results for the Group computation in the PC are also considerably high, with values reaching 10,4 and 31,98 seconds for 1500 and 2000 bit long Modulus. The value of 0,22 seconds for 500 bits is, on the other hand, mostly insignificant, with the 2,28 seconds for 1000 bits seemingly being the most appropriate compromise between computational cost and scalability.

However, in a production system, this calculation would be held in a dedicated mainframe, which would greatly reduce the time needed for its concretion. As such, facing these results, we propose for the Group generation to be performed in the Back Office with the results being sent to the user upon the first usage of the system, which is creates no problem since these values are not secret.

Given the importance of scalability in a system of this nature, we also assert for the maximum possible value to be used, seeing that the Group computation could be accomplished beforehand with the resulting values transmitted as they deemed necessary.

5.2.3 RSA Group Generation

The RSA Group, another building block of the Mobile Payment system, as discussed in Section 4.1.6. In contrast to the preceding Group generation, this Group is only necessary for the Bank and Merchant. As such, only the Laptop evaluation is performed.

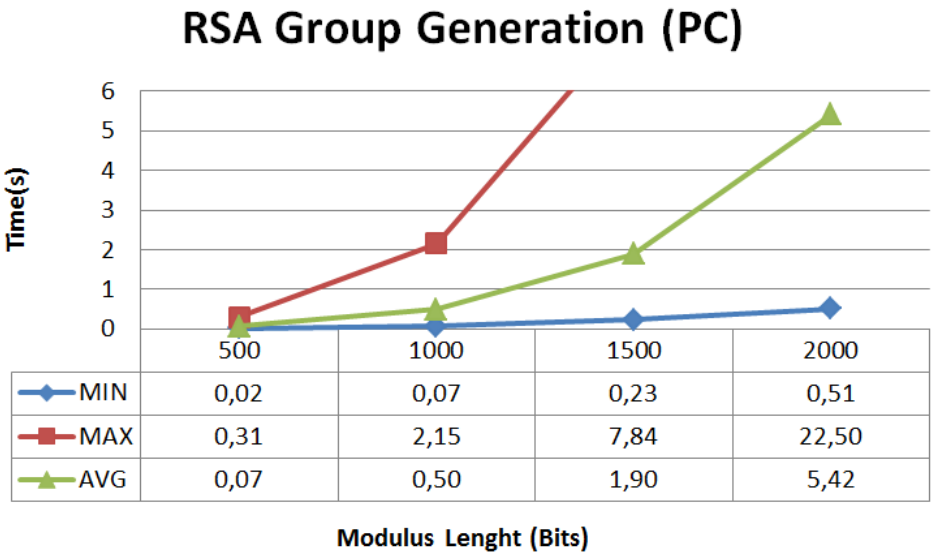


Figure 5.5: RSA Group Generation in PC

Despite supposedly requiring a more complex calculation than the normal Group generation, the results we obtained in the RSA Group generation gave results in straight contradiction with this assumption. 0,07 seconds for 500 long Modulus, and with 0,5; 1,9 and 5,42 seconds for the remaining values of, respectively, 1000, 1500 and 2000 bits.

Comparing the value of the 2000 bit calculation, we are faced with a 26,56 difference between the two procedures. This is due to the optimization that we implemented on the procedure, that essentially split the Modulus length in two separate values, with the product of these values resulting in the desired modulus size. Even repeating the process, these results show that it was considerably faster to calculate two distinct smaller values, than a bigger one.

As mentioned above, the calculation of this group is destined for the Back office, so the values obtained in that situation would be negligible, in terms of affecting the performance of the system. Nevertheless, seeing such an improvement in the calculation, it would be unreasonable not to apply the solution to the Group generation.

Likewise to the reasoning in the Group computation, the maximum possible Modulus Length should be used, improving scalability seeing that the RSA Group computation could be accomplished beforehand, with the resulting values transmitted as they deemed necessary. This solution is also valid for the point raised in the Key Generation, concerning possible limited resources being available to Merchants.

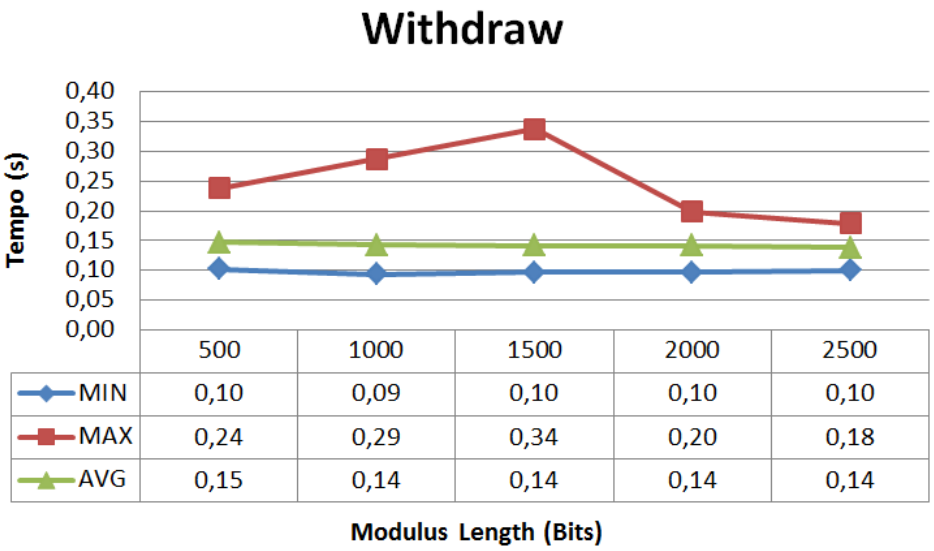
5.3 Protocols

Having drawn conclusions from the most computationally consuming processes in our system, we now move to performance tests performed on the main protocols of the Mobile payment system: Withdraw, in Section 5.3.1; Deposit, in Section 5.3.3 and Spend, in Section 5.3.2.

5.3.1 Withdraw

The Withdraw protocol features an interaction between User and Bank, described in Section 4.2.2, where the User withdraws a wallet with n coins from the Bank. As the Key Pair, Group and RSA Group have been previously determined, the necessary computation of the protocol is confined to the obtaining of random Group elements, a straightforward process given the pre-existing groups and simple algebraic operations between these values.

We started by measuring the the execution time of the Withdraw process with varying lengths of the Modulus length used in the generation of both RSA and normal groups.



	500	1000	1500	2000	2500
MIN	0,10	0,09	0,10	0,10	0,10
MAX	0,24	0,29	0,34	0,20	0,18
AVG	0,15	0,14	0,14	0,14	0,14

Figure 5.6: Withdraw Protocol

The variation was shown to have no influence whatsoever in the performance of the Withdraw process, with the average value remaining pretty much stable in the 0,14 seconds mark through the different iterations of the test. Seeing that the variation of the Modulus length is still quite significant, this seems counter intuitive. However, the explanation lies in the implementation of the system.

The Withdraw protocol, as described by Camenish et. al [3], requires, at certain points, for the exponentiation prime numbers (Group generators and group elements). While in a theoretical standpoint these are straightforward, on a practical situation, such as our prototype implementation, the exponentiation of numbers with a large bit count was deemed completely unfeasible.

Our solution to this problem was to recur to the Modulus value of the Key Pair in each Agent, and apply it in a Module operation to the number that would need to be exponentiated. Beyond severely cutting the computational load, this also resulted in the actual values obtained to be in an average range of 6500 bits, regardless of the Modulus length. In a production environment this would be addressed, for example, through the usage of hardware tailored to handle such calculations.

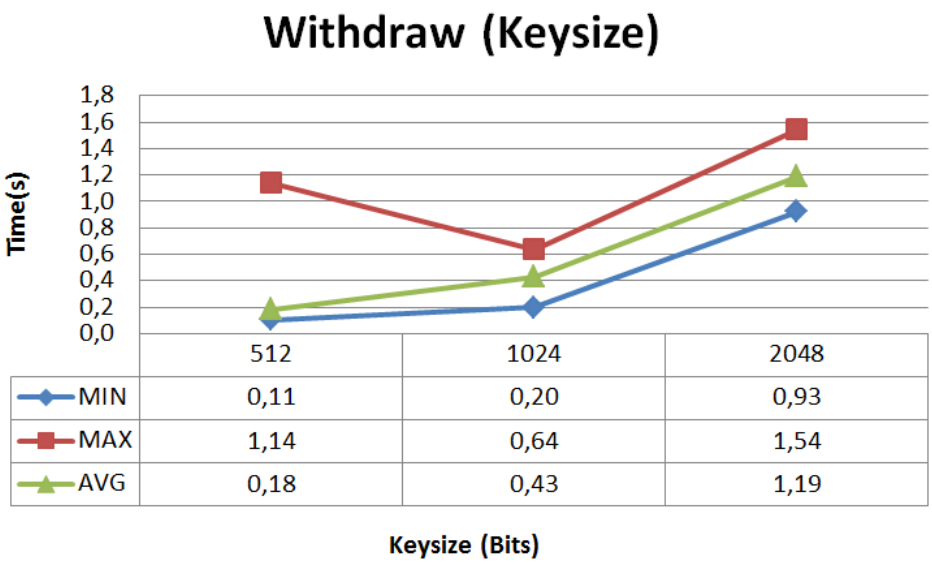


Figure 5.7: Withdraw Protocol

We then testes the variation of the Withdraw protocol based on varying values of Key size. The results obtained, with time values increasing from 0,18 seconds, to 0,43 seconds and finally to 1,19 seconds, for, respectively, 512, 1024 and 2048 bit Key sizes, might seem evident, seeing as a bigger Key size implies a bigger computational effort in the encrypting and decrypting of messages.

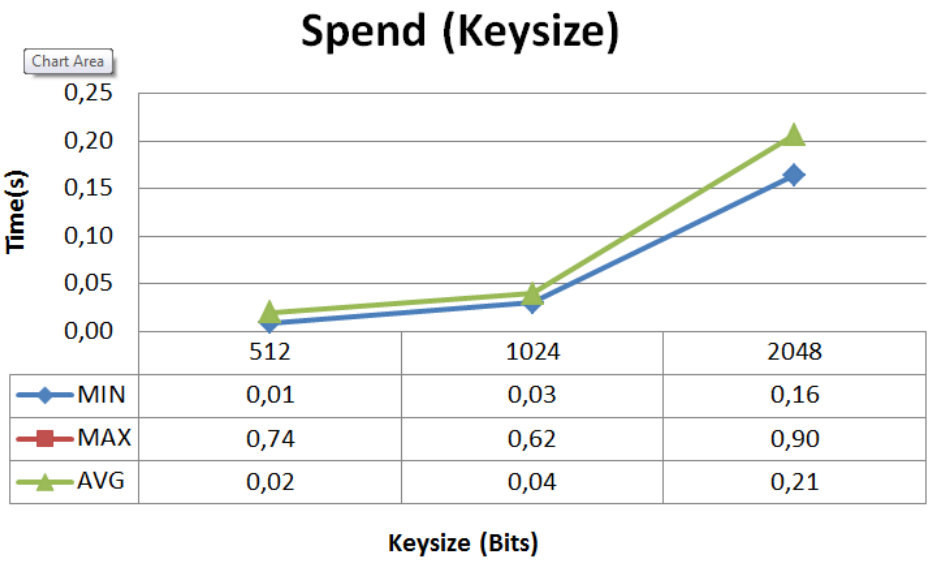
However, since, as mentioned above, we are dealing with rather large numbers, a characteristic of RSA encryption comes into play. Given a determinate n Key size, the RSA algorithm can only encrypt data blocks that go up until $(n/8) - 11$ bits (the 11 bits are used for padding). This translates into the need to split a message (i.e. a number) into several small blocks, encrypting, sending them, and having the other Agent to receive, decipher and concatenate all the blocks. Nevertheless our obtained results show that a single bigger computational effort surpasses a bigger number of smaller calculations.

As for the conclusions of this analysis, we find 1,19 seconds to be an acceptable time for the Withdraw process, remembering that the device used for the testing of the prototype is far from the state of the art and thus affirm that an 2048 Key size would be the most adequate for the system.

5.3.2 Spend

We now evaluate the performance on the Spend protocol. This protocol, described in Section 4.2.3, features the generation of a coin (or a number of coins) by the User, and the transmission of said coin to the Merchant.

We started by measuring the execution time of the Spend protocol, with varying lengths of Key size used in the generation of the RSA asymmetrical Keys, featuring the generation and transmission of a single coin.



	512	1024	2048
MIN	0,01	0,03	0,16
MAX	0,74	0,62	0,90
AVG	0,02	0,04	0,21

Figure 5.8: Spend Protocol

Unlike the Withdraw protocol, which requires heavy processing by the User and the Bank, with the transmission of the generated values, the Spend protocol only requires the User to receive the Merchant ID, a relatively small piece of data that requires no calculation and the transmission of the generated coin to the Merchant.

As such, the time for each Key size was verified to be shorter than the equivalent for the Withdraw protocol, with 0,02 seconds, 0,04 seconds and 0,21 seconds for 512, 1024 and 2048, respectively. These values are completely acceptable values for this procedure, offering a good user experience.

However, this test was performed, as mentioned, with using a single coin. As such, because the usage of multiple smaller tokens will probably be desirable, we asserted the execution of the Spend protocol with increasing values of coins being generated and transmitted.

The obtained results, the average values of the execution of the Spend protocol with varying values of Coins and Key sizes, allow us to draw some interesting conclusions.

Firstly, if we follow the previous recommendations on the usage of 2048 bit Key size, the maximum value of tokens that can be generated for a single transaction, without compromising User interaction is near 10, which resulted in an average 0,91 second execution time. For 1024 Keys allowing for the generation of 50 coins translated into an average of 0,68 seconds, translating into a more flexible option. Porting these numbers into a real world example, e.g. the purchase of a 6 Euro movie ticket, the 2048

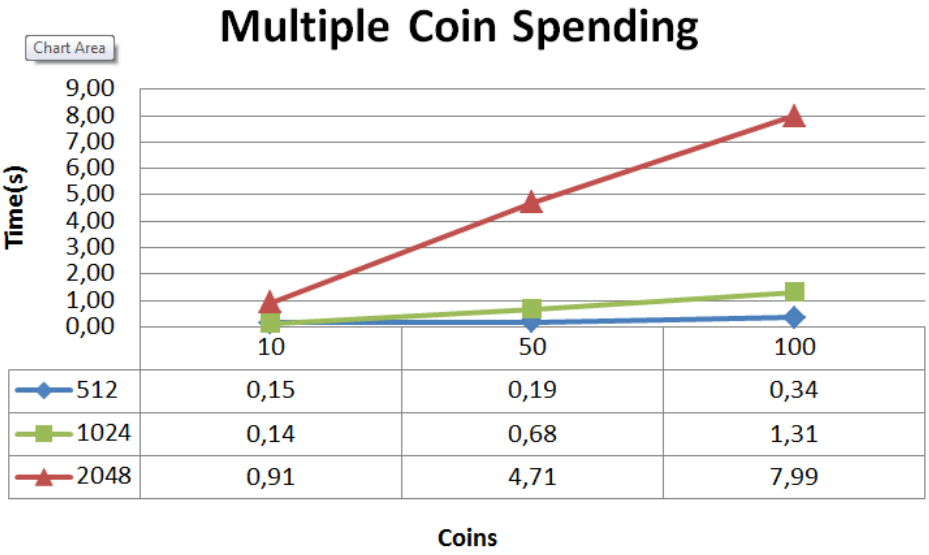


Figure 5.9: Multiple Coin Spending

bit Key size would allow for 50 Euro-cent tokens, while a 1028 Key size could lower the individual coin value to 20 Euro-cents.

Another interesting conclusion comes from comparing the obtained results with the ones presented in Table 5.2, taken from the isolation of a single coin generation process.

Device	**Smartphone**	**PC**
MIN (s)	0,005	0,00009
MAX (s)	0,224	0,13
AVG (s)	0,013	0,00013

Table 5.2: Coin Generation

The value obtained for the generation of a single coin in the Smartphone is 0,013 seconds. The equivalent value attained with on the Laptop of 0,00013 seconds might seem to indicate that it would be a sound idea for the token generation to be handled by the Back Office, or even by the Smartphone in anticipation of the actual use (e.g. as a coin was obtained). However, faced with this data it becomes clear that the most costly part of the Spend process is not the generation of the coin, but its encoding and transmission. While on 512 bit Keys this overhead is almost seamless, with 10 coins being generated and transmitted in a little over 10 times the indicated time for the generation of a single coin. On 2048 bits, on the other hand, this difference escalates to 70 times. If we take the values for 100 coins, with a 2048 bit Key size, the obtained value is 600 times superior to a single coin generation.

Beyond the obvious limitation of the tokens being generated using the Merchant ID as a component of the calculations, these results clearly discard the remote or scheduled generation as a significant approach to the optimization of this protocol.

5.3.3 Deposit

The Deposit protocol, described in Section 4.2.4, only implies calculations on previously generated values, with the deposit procedure itself being the costlier part of the process. Namely, when a coin is deposited, the Back office must check a list of existing coins for a duplicate coin, triggering the identify mechanism if one is found.

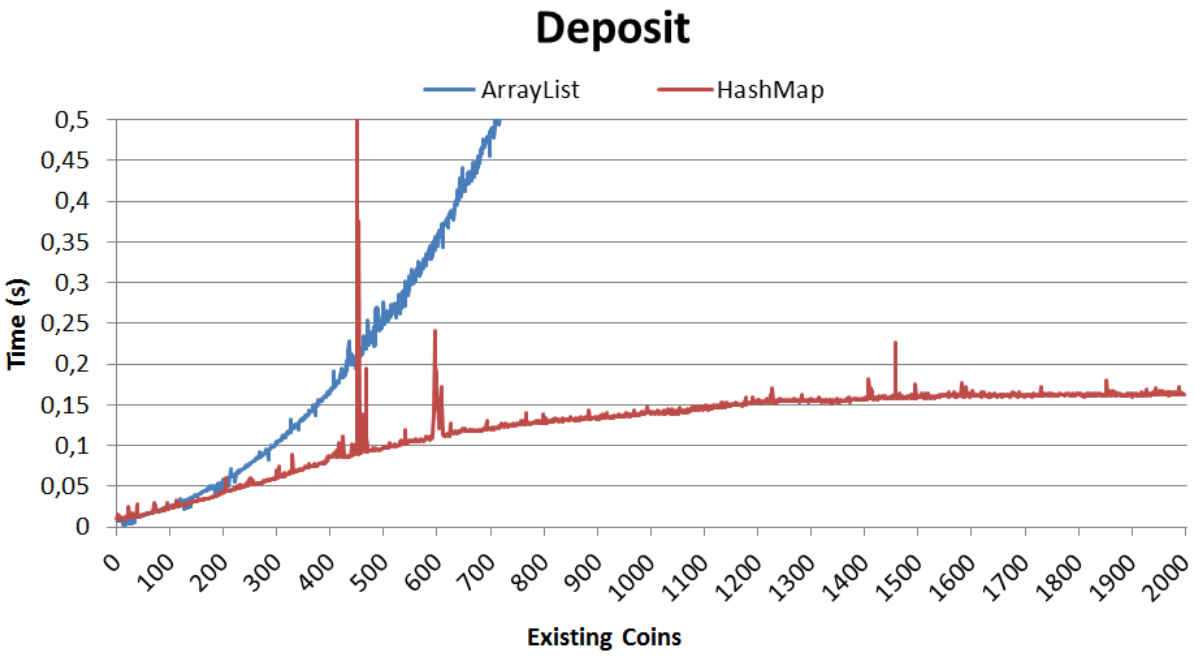


Figure 5.10: Deposit Protocol

On an initial phase, we implemented this list of coins by means of an ArrayList. As expected, and as depicted in Figure 5.10 this is not at all a scalable alternative. The list was then moved to a HashMap, with the sum of the two distinct coin identifiers S and T being set as the Key and the Coin object as the value.

The performance of this implementation is clearly adequate for is purpose, with the time spent in the deposit procedure climbing to 0,15 seconds and stabilizing in that value, for up to 2000 existing coins. Its important not to forget that our E-Cash system relies on *divisible* coins, which means that 2000 coins can imply a wide range of transactions. When facing a deployment to a production system, with distributed and dedicated storage systems would reduce this value even further.

Eventually, for improved performance the Bank entity could held a distinct coin list for different coin values, with high-value coins being given more relevance and computing power, for instance. Another idea would be to implement an expiration date on the coins, dependent on their value, which could reduce the amount of data that would be necessary to hold.

5.4 System Portability

The Mobile Payment system developed in this thesis was designed in a module interface with sights set on offering as much portability and interoperability as possible. Adding to this the research conducted on RF technology it is then possible to analyze and present various directions and systems in which the presented system could be applied and implemented in the future.

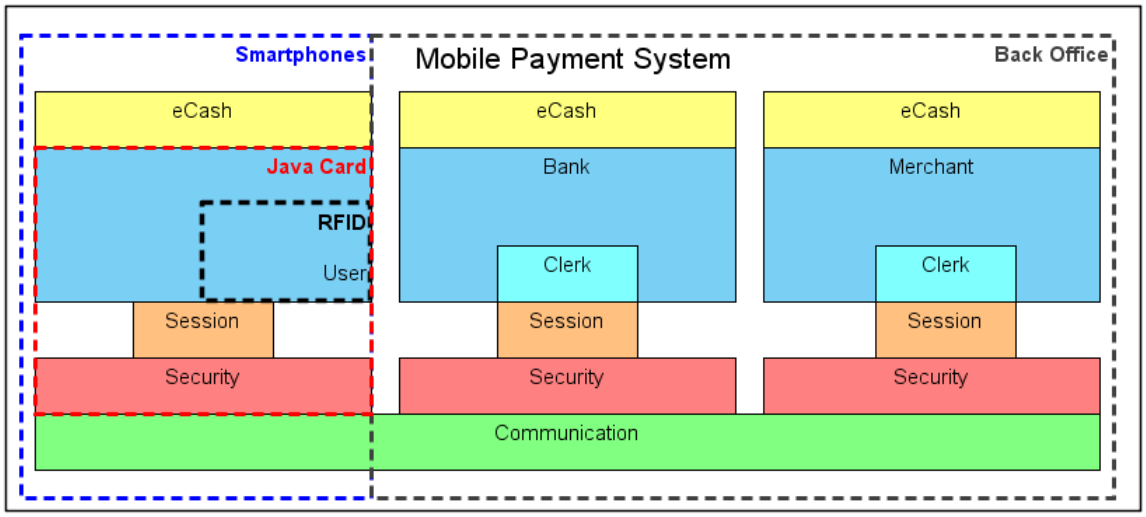


Figure 5.11: Portability of the Solution across systems

5.4.1 Hardware

While our concept solution was based on the usage of a Smartphone as the medium through which the Users interact with the system, the intended portability of the interface and of the Java language translates into the possibility for this system to be adapted to very distinct devices. As such, we now go into the considerations of using Smartphones, as well as Java Cards and RFID Tags, as said medium.

Smartphones Being the base used in the development of our proof of concept solution and seeing that in a short period of time they will become the *natural habitat* for mobile payment systems, smartphones are the most obvious recipient for the system presented in this thesis.

As depicted in Fig. 5.11, a smartphone implementation of our scheme should be able to independently handle all the components necessary to grant the functionality required for the User role. As per the tests taken in Section 5, it has been shown that such a system would be usable, albeit, as mentioned on said Section, the some of the processing could be shifted to the Back Office, in order to improve usability. The enhancement of mobile phone hardware would only concur with this claim, and devices posterior (i.e. with more capable hardware) to the one used for the testing of the concept might be able to handle all the processing independently.

Additionally to these capabilities, the presence of Universal Integrated Circuit Card (UICC) could be taken into an opportunity to store sensible information (i.e. the Private Key of the User) and/or allow for

mobile phones which originally did not provide NFC connectivity to be adapted and, as such, also being able to be used in our mobile payment system.

Java Cards A stepping stone between RFID Cards and Smartphones, Java Cards merge the processing capabilities of the Smartphone, albeit a very limited amount of them, with the disposability of RFID Cards, although they are much more costly than these.

Using a Java Card for our Mobile Payment system would translate in the possibility of all of the Users details to be held in the card itself, rather than the Back Office, since Java Cards have a considerable amount data storage space and, more notably, the capacity to independently generate (and hold) a full RSA Key Pair.

Most, if not all, the eCash computation, however, hold have to be produced in the Back-Office, since it is way beyond the capabilities of a Java Card. Less computational heavy details, such as the generation of random numbers could be feed by seeds generated on the Java Card, improving on the Security of the system.

RFID As per our extensive analysis of this medium, it has been established that the processing capabilities of RFID Tags are very limited. As such, the adaptation of our proposed solution to comprehend the usage of RFID would require all computation and most of the characteristics of the User to be held by the Back Office, as depicted on Fig. 5.11.

This would translate into the Withdraw and Spend protocols having to be run single-handedly by the Bank and Merchant, respectively, with the RFID card holding only a small identifier for the User in question.

One manner in which this could be implemented would require Users to interact with the system in order to obtain the Tags, in a manner similar to the one used in contact-less payment systems of public transportation services. This could happen either through the input of money into a terminal, or through requiring users to accomplish a registration that would bind their account in the mobile payment system to a source of currency (i.e. a Bank account).

As for the data to be actually held up in the Tag itself, several options are also available. The allocation of all of the User's data into the Back Office would mean that the Tag would only have to hold the 2^l counter necessary to limit the spending to the User's balance, although this, if coupled with the direct charging of Cash into a terminal would fall outside of the scope (and the advantages) of the system that we are proposing, due to the lack of privacy in the generation of tokens.

A more consensual translation of our Compact E-Cash system would imply the need for the Tag to, alongside with the counter, hold a secret of the User, that would then enable the direct translation of all the protocols. This secret might be the full RSA Key Pair (i.e. the Public and Private keys, pkU and skU), only the Private component of the Key Pair, or a fraction of the private component, depending on the data holding capacity of the Tag to be used.

The usage of such a dispensable and limited medium, however, would require other considerations to be held, such as the limit on the balance that the card could hold and/or the possibility of canceling

a previously issued card, accounting for the fact that Users could (and would) lose the medium more frequently, and the presence of an expiration date on the card, especially necessary if the card is not able to hold the full RSA Key.

5.4.2 Wireless Technologies

On what concerns the medium used for the communication, the resort to different technologies would provide us with both opportunities and threats to consider, such as Wi-Fi, Bluetooth, NFC and RFID. We shall now provide an insight into each of these options, except for RFID, as this the portability of our system to technology was already assessed in Sector 5.4.

Wi-Fi The mean on which the prototype was developed, Wi-Fi has a throughput capacity vastly superior to any of the alternatives. The fact that most, if not all, smartphones offer Wi-Fi connectivity and the relative widespread presence of Hot-spots might lead to believe that, perhaps more than one would expect, Wi-Fi is a truly feasible mean to implement Mobile Payments. Various solutions already exist, but virtually all of these resorts to a *remote payment* interaction type.

A solution based on Wi-Fi *proximity payment*, like the one developed in the presented prototype, could benefit from the ubiquity of Wi-Fi, while the reduction on the range of the Wi-Fi Hotspots would create smaller areas on which the mobile payment system could be used. This would resolve the main issue of the Wi-Fi connectivity for such a system, the fairly large range from which the communication between legitimate Agents and Services could be intercepted.

Furthermore, we feel that the security measures used, namely the encryption of all the transmissions in a Diffie-Hellman agreed Security Key and the authentication procedure that precedes any protocol are sufficient, considering the reduced values transfered in Mobile Payment systems.

Bluetooth With the early versions of Bluetooth communication allowing data rates of up to 1 and 3 Mbits/s (version 1.2 and 2.0, respectively). Taking into account the results obtained in Section 5.3, we consider that the throughput capacity of Bluetooth is completely able to deal with the amounts of data required for the communication in our proposed E-Cash system.

Furthermore, the piconet ad-hoc network is very close concept to the Reader/Tag duo of RF communication. Since our prototype was developed with this communication style in mind, the porting of the solution to Bluetooth technology would mostly be limited to an adaptation of the Communication layer.

NFC On NFC, as we noted in Section 2.2.1, allows for throughput rate of up to 424kbits/s. While being sufficient for the initial establishment of the communication (i.e. the exchange of Public Keys between Agent and Service, the role assertion and the Challenge-response protocol), this procedure should to be prompted to the User as a preamble to the Withdraw and Spend protocols, in order to reduce appearance of time needed to accomplish the protocols.

The Withdraw and Spend protocols, on the other hand, as noted in Sections 5.3.1 and 5.3.2, respectively, require an amount of data throughput that NFC would never be able to handle. There are

two general possible workarounds for this issue: the Withdraw and Spend protocols could be ran mostly via the Back Office, with the NFC device contributing only with the data required for the calculations (i.e. group generators, group elements); alternatively, the NFC platform could be used to expedite the establishment of the communication via a more capable connection, such as Wi-Fi or Bluetooth.

5.5 System Evaluation

The presented solution is meant to be the groundwork for a Mobile Payment system, rather than a system meant to be produced and pushed to end-Users. As such, the interface itself inspired little consideration seeing that the concretization of it could be done in a system with substantially different characteristics (e.g. an Apple device with very strict regulations on the interface system) or with no interface whatsoever (e.g. a Java Card or an RFID Tag).

Even so, some inquiries were realized that could render useful insight into the user expectations for this kind of systems. In these queries, while not adding to a large amount of individuals, we managed to target a wide range of the population that would use these systems, with the age of said individuals ranging from 20 to 50 years old. The total number of answers amounts to 73, with 70% of these belonging to the 19-25 age group (46 individuals), 21% being aged between 25-35 years old (14 individuals) and the remaining 7% belonging to the 35-55 interval (5 individuals).

It is important to have in consideration that the economic possibilities of these individuals reflect those that would be the short to medium term Users of such a system.

5.5.1 Wallet Currency

Age Group	**20-25**	**25-35**	**35-55**
Favored Maximum Amount of Currency (EUR)	20	35	25
Average Expected Balance (EUR)	5.5	15	N.A.
Mininum Average Payment (EUR)	0,5	0,5	N.A.
Maximum Average Payment (EUR)	7,5	16	N.A.
Relevance of Criteria (0-5)	5	5	N.A.

Table 5.3: Wallet Currency Query Results

On this query it was asked to potential users what was the maximum amount of currency that a Wallet should hold, what would be their regular balance and how relevant was this payment system to them. We present the results of these queries in Table 5.3.

Based on this data its possible to draw some conclusions:

- In what concerns the *Maximum Amount of Currency* in the system our results show that the age group between 25 and 35 is the one that expects a higher amount. This is due to the fact that this

segment has both a higher buying power than younger users and the higher confidence in such a system than the last section of users;

- The *Average Expected Balance* and the *Average Payments* reflect distinct usage patters from each analyzed segment. Younger users would use the system essentially for small transactions and keep a lower balance, which would imply more frequent withdrawals; Users from 25 to 35 would keep a higher balance, in order to be able to rely more on the system and would diversify their payments, using the system both for small and medium purchases (i.e. 5 to 10 EUR); From the elder segment we could not get any significant data, due to the unawareness of these users relatively to the possibilities of these kind of systems.

- On the last item, the *Relevance* of these amounts as system characteristic, there was a clear consensus, with all age groups stating that this would be one of the top criteria in the adoption of any Mobile payment system.

5.5.2 Mobile Payment Types

The aim of this evaluation was to discover the Mobile Payment type that users favored more. As mentioned in section these can be characterized on a basis of Interaction Type (Remote or Proximity), Basis of Payment (Wallet or Account Based) and Transaction Type (On-line, Off-line and Semi-Offline). Seeing that the focus of this thesis is on RF systems, and seeing that in the Transaction Type is transparent to eyes of users, both the Interaction Type and the Transaction Type were disregarded.

As such, we present the results obtained in an inquiry about Basis of Payment in Mobile Payments in Table 5.4.

Age	**20-25**	**25-35**	**35-50**
Account Based (%)	59%	7%	20%
Wallet Based (%)	26%	57%	20%
Unclear (%)	15%	14%	60%

Table 5.4: Basis of Payment Results

As it happened before, we reach some interesting conclusions.

- In the lower range of the age spectrum that was contacted, the preference clearly goes to the *Account Based* payment type. Users explained that this is due to the increased convenience of not having the necessity to *top up* the account, something that would happen frequently due to the usage patterns reflected on the previous survey;

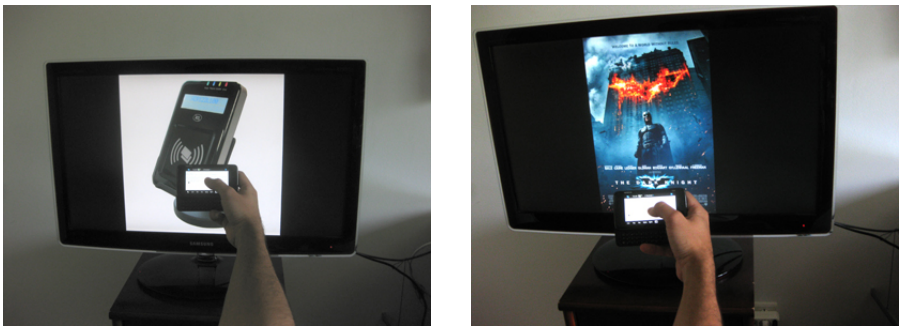
- On the middle tier, the preference clearly went to the Wallet based solutions, with Users proclaiming that this would make it easier to trust such a system. This is not related to the security of the

used E-Cash scheme or of the communications involved, but due to the fact that they are concerned about what would happen if the device was lost or stolen. In this case, the existence of a Wallet, with a limited amount of currency available, helps to ease the state of mind of the users;

- Lastly, users on the top tier, while giving Walled based solutions a preference, are still unclear on the usage of such a systems, leading to indecision on this matter;

5.6 Interface considerations

While the interface on the presented solution was far from a final product, as mentioned previously, a usage test was still realized. Due to said lack of development on the interface it is impossible to provide any accurate metrics. However, feedback gathered from users still provided some interesting insights into concerns that should be held in interface design for such systems.



(a) (b)

Figure 5.12: Simulation of Withdraw (a) and Spend (b) protocols

For this test, two usage scenarios were simulated. Firstly, users would have to withdraw a Wallet, by keeping a mobile phone with the solution in RF range (5-15cm) to a virtual Bank entity, simulated through the display of a NFC target on a screen (as depicted in Fig. 5.12(a)). Then, users would be prompted to spend this currency on the purchase of a movie ticket. This was simulated through the display of a Movie poster on a screen (as depicted in Fig. 5.12(b)) with users having to perform the spend protocol while keeping the phone in RF range (5-10cm) to the simulated poster. Doing so, we gathered the following suggestions:

- The interface must be as clear as possible. Given that the system would handle real currency, users clearly expressed that the interface must be unequivocal at all times, allowing for the user to be able to discern, without any doubt, the amount of money that was being handled both in the withdrawal and spending;

- The system must be implemented on a portrait screen orientation. While this is pattern for most smartphones, our system was limited to a landscape orientation. All users expressed their dislike for this, claiming that this orientation clearly interferes with the interaction with the system (i.e. holding the smartphone in one hand and using the other for input);

- Users, especially the elder ones, also expressed concern for the need to keep the system within reach of the NFC target. Their worries were that a momentary increase in distance might lead to interruptions in the correct execution of the protocols, which might be prejudicial for them (e.g. spending the money on the movie ticket and failing to receive the ticket, or getting a debt recorded on the Bank without receiving the Wallet). To this effect, we consider to be of the utmost importance to implement a state keeping feature on every protocol, allowing a session to persist for some amount of the time should the system lose contact with the RF field until the user, prompted to correct this situation, accomplishes it.

Chapter 6

Conclusions

With the current interest in Mobile Payments by big companies such as mobile communication companies, financial institutions and Internet powerhouses like Google this area will undoubtedly experience a very intense development in a short term perspective. However, while the systems developed by these companies will surely be proficient and able to offer Users the advantages of replacing physical currency by a digital medium, it is very unlikely that User Privacy will be a concern.

In fact, we can go out in a limb and assert that User Privacy will not only be a low valued priority, but its absence will be a considerable advantage to the companies implementing such systems, seeing that information straight from Users on their shopping and general spending habits is a very powerful and valuable asset.

Our solution, on the other hand, goes against the grain in the setting of User Privacy as a main priority. While it effectively deems the system mostly invaluable for any company we firmly believe that this is the right thing to do, especially in this period in History were personal Privacy is turning from a fundamental right into a mirage.

The analysis undertaken into the state of the art of RF systems allowed for a clear insight into the capabilities of such systems, especially into the Security and Privacy concerts that affect such systems. Moreover, the evaluation of existing Mobile Payment state of the art, business models and solutions only reinforced the importance of our focus on Privacy.

The proof-of-concept that was implemented was extensively evaluated, proving that a development into a full-scale system would to be clearly feasible. And not only on the medium used in the concept, as our analysis showed that the Portability of the system to be indisputable.

Bibliography

[1] R. Balan, N. Ramasubbu, and K. Prakobphol. mFerio: the design and evaluation of a peer-to-peer mobile payment system. *Security*, pages 291–304, 2009.

[2] R. Boguslaw. Privacy and Freedom. *American Sociological Review*, 33(1):173, February 1968.

[3] J. Camenisch and S. Hohenberger. Compact e-cash. *Science*, 2005.

[4] S. Canard and A. Gouget. Divisible e-cash systems can be truly anonymous. *Advances in Cryptology-EUROCRYPT 2007*, pages 482–497, 2007.

[5] M. Carr. Mobile Payment systems and services: An introduction. *Mobile Payment Forum*, pages 1–12, 2007.

[6] A. Chan, Y. Frankel, and Y. Tsiounis. Easy come - easy go divisible cash. *Advances in Cryptology - EUROCRYPT 1998*, pages 561–575, 1998.

[7] D. Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto*, volume 82, pages 199–203, 1983.

[8] R. Das and P. Harrop. RFID forecasts, players and opportunities 2005-2015. *IDTechEx report*, 2005.

[9] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

[10] Y. Dodis and A. Yampolskiy. A Verifiable Random Function with Short Proofs an Keys. *Public Key Cryptography*, 2005.

[11] D.N. Duc, J. Park, H. Lee, and K. Kim. Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning. In *The 2006 Symposium on Cryptography and Information Security*, pages 97–102. Citeseer, 2006.

[12] ecma International. NFC-SEC-01: NFC-SEC Cryptography Standard using ECDH and AES. (June), 2010.

[13] ecma International. NFC-SEC: NFCIP-1 Security Services and Protocol. 2010.

[14] K.P. Fishkin and S. Roy. Some methods for privacy in RFID communication. *Security in ad-hoc and sensor networks*, 27(June):42–53, 2005.

[15] C. Floerkemeier, D. Anarkat, T. Osinski, and M. Harrison. PML core specification 1.0. *Auto-ID Center Recommendation*, 15(September), 2003.

[16] C. Floerkemeier, R. Schneider, and M. Langheinrich. Scanning with a purpose - supporting the fair information principles in RFID protocols. *Ubiquitous Computing Systems*, pages 214–231, 2005.

[17] E. Fujisaki and O. Tatsuaki. Statistical zero knowledge protocols to prove modular polynomial relations. *Advances in Cryptology CRYPTO 97*, 1997.

[18] S. Garfinkel, A. Juels, and R. Pappu. RFID Privacy: An Overview of Problems and Proposed Solutions. *IEEE Security and Privacy Magazine*, 3(3):34–43, May 2005.

[19] E. Haselsteiner and K. Breitfuß. Security in Near Field Communication (NFC) Strengths and Weaknesses. *Proceedings of Workshop on RFID Security*, 2006.

[20] N. Hughes and S. Lonie. M-PESA : Mobile Money for the "Unbanked" Turning Cellphones into 24-Hour Tellers in Kenya. *Innovations*, (March):63–81, 2007.

[21] S. Inoue and H. Yasuura. RFID privacy using user-controllable uniqueness. In *RFID Privacy Workshop*. Citeseer, 2003.

[22] A. Juels. Minimalist Cryptography for Low-Cost RFID Tags. *New York*, 2003.

[23] A. Juels and R. Pappu. Squealing Euros: Privacy protection in RFID-enabled banknotes. In *Financial Cryptography*, pages 103–121. Springer, 2003.

[24] A. Juels, P. Syverson, and D. Bailey. High-power proxies for enhancing RFID privacy and utility. In *Privacy Enhancing Technologies*, pages 210–226. Springer, 2006.

[25] A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology - CRYPTO 2005*, pages 293–308. Springer, 2005.

[26] G. Karjoth and P. Moskowitz. Disabling RFID tags with visible confirmation: clipped tags are silenced. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, volume 23710, pages 27–30. ACM, 2005.

[27] S. Karnouskos and F. Fokus. Mobile payment: a journey through existing procedures and standardization initiatives. *Communications Surveys & Tutorials, IEEE*, pages 44–66, 2004.

[28] M. Kerschberger. *Near Field Communication - A survey of safety and security measures*. PhD thesis, 2011.

[29] K. Koscher, A. Juels, T. Kohno, and V. Brajkovic. EPC RFID tags in security applications: passport cards, enhanced drivers licenses, and beyond. In *ACM Conference on Computer and Communications Security*, pages 33–42. Citeseer, 2009.

[30] D. Kumar, T. Gonsalves, A. Jhunjhunwala, and G. Raina. Mobile payment architectures for India. *2010 National Conference On Communications (NCC)*, (1):1–5, January 2010.

[31] L. Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, November 1981.

[32] J. Landt. Shrouds of Time: The history of RFID. *Scientist*, 2001.

[33] J Langer. Anwendungen und Technik von Near Field Communication (NFC). *Technik von Near Field Communication*, 2010.

[34] K.S. Leong, M.L. Ng, and D.W. Engels. EPC network architecture. In *Auto-ID Labs Research Workshop*. Zurich, Switzerland, 2004.

[35] C. Loebbecke. Towards item-level RFID in the Japanese publishing industry. *of the 13th Americas Conference on*, 27(2004):1–8, 2007.

[36] A. Lysyanskaya and J. Camenisch. A signature scheme with efficient protocols. *Security in communication networks*, 2003.

[37] G Madlmayr, J Langer, and C Kantner. NFC devices: Security and privacy. *Reliability and Security*, 2008.

[38] S. Mainwaring and W. March. From meiwaku to tokushita!: lessons for digital money design from japan. *of the twenty-sixth annual SIGCHI*, pages 1–4, 2008.

[39] T. Nakanishi, M. Shiota, and Y. Sugiyama. An unlinkable divisible electronic cash with user's less computations using active trustees. In *Proc. ISITA2002*, pages 547–550. Citeseer, 2002.

[40] T. Nakanishi and Y. Sugiyama. Unlinkable divisible electronic cash. *Information Security*, pages 179–206, 2000.

[41] NFC Forum. The NFC Forum. *http://www.nfc-forum.org/aboutus/*.

[42] NFC Forum. NFC Data Exchange Format (NDEF) Technical Specification. *History*, 2006.

[43] NFC Forum. Logical Link Control Protocol Technical Specification. 2011.

[44] NFC Forum. Simple NDEF Exchange Protocol Technical Specification. 2011.

[45] M. Nzualo. MobiPag : A System for Interoperable Peer-to-Peer Proximity Mobile Payment Applications. 2007.

[46] T Okamoto. An efficient divisible electronic cash scheme. *Advances in Cryptology - CRYPTO 95*, 1995.

[47] J. Ondrus and Y. Pigneur. An Assessment of NFC for Future Mobile Payment Systems. *International Conference on the Management of Mobile Business (ICMB 2007)*, pages 43–43, July 2007.

[48] K. Osaka, T. Takagi, K. Yamazaki, and O. Takahashi. An Efficient and Secure RFID Security Method with Ownership Transfer. *2006 International Conference on Computational Intelligence and Security*, pages 1090–1095, November 2006.

[49] M.R. Rieback, B. Crispo, and A.S. Tanenbaum. RFID Guardian: A battery-powered mobile device for RFID privacy management. In *Information Security and Privacy*, pages 184–194. Springer, 2005.

[50] S. Sarma. Toward the five-cent tag.pdf. *Technical Report MIT-AUTOID-WH-006*, 2001.

[51] S. Sarma, S. Weis, and D. Engels. RFID systems and security and privacy implications. *Cryptographic Hardware and Embedded Systems-CHES 2002*, pages 1–19, 2003.

[52] Supply Chain Diggest. The Five Cent RFID Tag is Here, the Five Cent Tag RFID is Here! Well, Almost. *Supply Chain Diggest*, pages 1–2, 2009.

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

[54] K. Wagstaff. Smartphones Outnumber Feature Phones in U.S. for First Time, 2012.

[55] R. Want. The magic of RFID. *Queue*, 2(7):40–48, 2004.

[56] S. Weis, S. Sarma, R. Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. *Security in Pervasive Computing*, pages 50–59, 2004.