MOBILE AUTHENTICATION WITH NFC ENABLED SMARTPHONES

Electrical and Computer Engineering
Technical Report ECE-TR-14

AARHUS
UNIVERSITY
DEPARTMENT OF ENGINEERING

DATA SHEET

Title: Mobile Authentication with NFC enabled Smartphones
Subtitle: Electrical and Computer Engineering
Series title and no.: Technical report ECE-TR-14

Authors: Rawad Kilani and Kenneth Jensen
Department of Engineering – Electrical and Computer Engineering,
Aarhus University

Internet version: The report is available in electronic format (pdf) at the Department of Engineering website http://www.eng.au.dk.

Publisher: Aarhus University©
URL: http://www.eng.au.dk

Year of publication: 2012 Pages: 101
Editing completed: March 2013

Abstract: Smartphones are becoming increasingly more deployed and as such new possibilities for utilizing the smartphones many capabilities for public and private use are arising. This project will investigate the possibility of using smartphones as a platform for authentication and access control, using near field communication (NFC). To achieve the necessary security for authentication and access control purposes, cryptographic concepts such as public keys, challenge-response and digital signatures are used. To focus the investigation a case study is performed based on the authentication and access control needs of an educational institutions student ID. To gain a more practical understanding of the challenges mobile authentication encounters, a prototype has successfully been developed on the basis of the investigation. The case study performed in this project argues that NFC as a standalone technology is not yet mature to support the advanced communication required by this case. However, combining NFC with other communication technologies such as Bluetooth has proven to be effective. As a result, a general evaluation has been performed on several aspects of the prototype, such as cost-effectiveness, usability, performance and security to evaluate the viability of mobile authentication.

Keywords: electronics, power electronics, optics and photonics, biomedical devices and applications, communication systems, digital signal processing, embedded systems, software engineering and systems

Supervisor: Rune Hylsberg Jacobsen

Please cite as: Rawad Kilani and Kenneth Jensen, 2013. Mobile Authentication with NFC enabled Smartphones. Department of Engineering, Aarhus University. Denmark. 101 pp. - Technical report ECE-TR-14

Cover photo: Rawad Kilani and Kenneth Jensen

ISSN: 2245-2087

MOBILE AUTHENTICATION WITH NFC ENABLED SMARTPHONES

Rawad Kilani and Kenneth Jensen

Aarhus University, Department of Engineering

Abstract

Smartphones are becoming increasingly more deployed and as such new possibilities for utilizing the smartphones many capabilities for public and private use are arising. This project will investigate the possibility of using smartphones as a platform for authentication and access control, using near field communication (NFC). To achieve the necessary security for authentication and access control purposes, cryptographic concepts such as public keys, challenge-response and digital signatures are used. To focus the investigation a case study is performed based on the authentication and access control needs of an educational institutions student ID. To gain a more practical understanding of the challenges mobile authentication encounters, a prototype has successfully been developed on the basis of the investigation. The case study performed in this project argues that NFC as a standalone technology is not yet mature to support the advanced communication required by this case. However, combining NFC with other communication technologies such as Bluetooth has proven to be effective. As a result, a general evaluation has been performed on several aspects of the prototype, such as cost-effectiveness, usability, performance and security to evaluate the viability of mobile authentication.

Acknowledgements

This Master's Project was conducted between August and December of 2012 at Aarhus school of engineering. We would like to thank Morten Høy Thellefsen, Ivan Damgård, Bjarke Parner and Søren Harbo Jensen for sharing their insight, perspective and providing interesting discussion within the subject of this document.

We particularly would like to thank Assoc. Prof. Rune Hylsberg Jacobsen for his guidance and advice prior and throughout the duration of this project.

Finally we would like to thank Logica for introducing us to the subject of this document and providing us with assistance and guidance.

Kenneth Jensen

Rawad Kilani

Table of Contents

List of figures

Acronyms and abbreviations

API Application Programming Interface

CA Certificate Authority

CIA Confidentiality, Integrity and availability

CPU Central processing Unit

HTTP Hypertext transfer protocol

HTTPS HTTP Secure

ICC Integrated circuit card

ID Identity

MAC Message authentication code

MAC address Media Access Control address

MITM Man in the middle

MNO Mobile network operator

NFC Near field communication

OTP One time password

PC Personal computer

PDA Personal digital assistant

PKI Public key infrastructure

PIN Personal identification number

POS Point of sale

RAM Random access memory

RFID Radio frequency identification

ROM Read-only memory

RSA Rivest-Shamir-Adlerman public key cryptosystem

SE Secure element

SIM Subscriber identity module

SMS Short message service

SSL/TLS Secure socket layer/Transport layer security

UI User interface

UICC Universal integrated circuit card

UMTS Universal mobile telecommunications service

USB Universal serial bus

QoS Quality of Service

1 Introduction

1.1 Problem Description

This Master's Project addresses the problem of authenticating identity using smartphones. To make this problem more concrete a scenario have established with the consulting firm Logica. The scenario will be based on the education systems student identification needs.

Currently the majority of the education institutions in Denmark rely on student identification cards for identification purposes. This includes authorization of access to school facilities and resources, but also access to external services offered by third parties such as shops and stores offering rebates.

Authentication is to be understood as the action by the School or third parties (hereafter denoted as the authenticator) confirming that they are interacting with a legitimate student, and vice-versa, i.e. the student making sure that she is interacting with the authenticator. Additionally, authentication is not restricted to the entities as presented above, but also to the information exchanged between them. Meaning that a party must be able to determine whether the information is seemingly coming from the other party, does in fact originate from it.

The importance of authenticating identity lies in the fact that the authenticator must only allow legitimate students access to services. The basis of this is establishing the identity of the student. For the purposes of identification using a device such as the smartphone that is already available to students and a part of their daily life could make authentication simpler. By simplifying the authentication process using smartphones several potential gains can be achieved such as cheaper authentication and more secure authentication. This is especially relevant because the smartphone is increasingly offering new technologies and features, which simplifies the required user interaction.

The scope of this project consists of presenting a general survey of the state of the art in mobile identity authentication, followed by development and assessment of a Near Field Communication (NFC) based authentication mechanism. NFC is an emerging technology based on Radio Frequency Identification (RFID) that allows devices to communicate over short distances (max 10 centimeters). This technology has already been incorporated into some commercially available mobile phones, and services are already provided such as public transportation and ticketing systems [3].

1.2 Motivation and Goals

The concept of a mobile student identity card arises from the major technological developments of the 21th century. We are currently living in a world where young people expect technology to accommodate their needs of easier access to public and commercial services. Tax can be reported via the internet, bank accounts can be accessed through smartphones. The next step is linking electronic identification with the physical realm, thereby making it possible to implement access control. A device capable of providing such a link to physical objects is the smartphone. There are already smartphones being used to access public services such as public transportation and ticket purchasing systems [1].

In Denmark there is a large growth in the amount of smartphones among the populace. 2011 to 2012 has seen an increase of smartphones of about 15%, reaching a market penetration of 50% [2]. This large increase over just a year indicates smartphones replacing the older feature phones. The increase of smartphones will pave the path for new ways of accessing public services. The smartphone as a platform has huge potential, it can provide pervasive access to the internet through 3G and 4G networks, and more importantly it is a device people carry around almost everywhere they go. Furthermore, the smartphone is a device which is still evolving. Currently new smartphones are entering the market with technology such as NFC (Near field communication). NFC will allow the smartphone to wirelessly interact with physical objects or terminals. The Interaction with physical objects will enable the phone to simplify the daily life for its users. NFC can seamlessly receive and transfer information which would otherwise be tedious work for the user to input.

NFCs capabilities enables the smartphone to become a platform for identification, which in turn can be used to access public services such as school services, transportation services and eventually even health care services. The smartphone has the potential to become a multi-identity platform, however to keep the project and document within a scope suitable for a master thesis, we envision the smartphone in a Mobile Student Identification Card (MS-ID) scenario. The MS-ID scenario is a more maintainable scenario and is a suitable proof of concept for smartphones potential capability to become a multi-identity platform, which can eventually replace the plethora of physical identification cards available today.

The immediate benefits of the MS-ID system will be a reduction in the expenses paid by school authorities for creating and distributing plastic student ID cards containing RFID chips and magnetic stripes. Furthermore, the benefit for the students will be increased usability with regards to not having to print out an enrollment confirmation, and remove the need for carrying a card only usable for student ID. From the shop owners perspective the MS-ID will provide a trustworthy mechanism to authenticate students eligible for rebates. It is rather easy to forge a student ID as there is no uniformity among student ID cards, and as such it can be very difficult for shop owners to confirm the authenticity of the ID card.

To examine the possibilities of an MS-ID application, this document will provide a proof of concept which will enlighten viability of mobile authentication using smartphones. The proof of concept will be developed using existing standards and technology rather than dealing with a radically new solution to the

authentication problem, and use existing infrastructure wherever possible instead of proposing radical and expensive changes.

1.3 Document Outline

The rest of this document is organized as follows: Chapter 2 introduces information security fundamentals and cryptographic concepts and terms regarding information security, with special focus on authentication. Chapter 3 scopes and describes Mobile Student ID (MS-ID), identifying its actors, associated goals and vulnerabilities in the form of a threat model which will be used to assess the effectiveness of the MS-ID security system presented in chapter 6.

Chapter 4 introduces the most relevant technologies and standards used in mobile authentication, focusing on those used by the proposed solution presented in chapter 6, and provides a security assessment of these. Chapter 5 presents a survey of mobile authentication, identity and security techniques, mechanism and considerations, by outlining their strengths and limitations in the light of goals presented in Chapter 3.

Chapter 6 is the core of this document as it describes and discusses the proposed authentication solution using NFC enabled smartphones. Chapter 7 discusses the subject of this document and it examines the attributes of the prototype. Chapter 8 recapitulates the lessons and conclusions obtained from this project.

2 Overview of Information Security

This chapter will attempt to clarify concepts, definitions and terms for information security used in the following chapters. This will only be a brief introduction and for more comprehensive and detailed coverage the authors refer to the main sources of this chapter [4, 5, 6, 7, 8, 25, 35].

Points of coverage will be to introduce the fundamentals of information security in section 2.1, while explaining the concepts and terms used within cryptography to understand notations in section 2.2. Section 2.3 provides insight into access control and means to achieve it by using authentication.

2.1 Information Security: Fundamentals

Security is defined by William Stallings in [4] as "*The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity availability and confidentiality of information system resources (includes hardware, software, firmware, information/data and telecommunications)".*

The definition itself does not provide a lot of information without a context, and opens more questions than it answers. To discuss this subject we need a clear context which can be defined by answering four fundamental questions: *What* is being protected? *From whom* is it being protected? *Against* what is it being protected? And *How* is it being protected?

To have any meaningful discussion about security the answer to the four questions must be absolutely clear. It is important to keep in mind that these answers can be very subjective as the *entity* answering is not necessarily a person. It can be an organization, or a government which must answer these questions to give security a meaning. As a consequence, security can have a very different meaning from individual to individual and definitely a different meaning to organizations compared to a person. Anderson presents an example in [7]: *"To a corporation it [security] might mean the ability to monitor all employees' email and web browsing; to the employees, it might mean being able to use email and the web without being monitored"*.

What is being protected is called an asset. An asset can be a person, organization, infrastructure, etc. The asset or object does not need to be physical, it can be information that is not necessarily physically bound. The most common and important asset in security systems is information, which must be protected from being revealed, tampered with or made unavailable. Even though information is usually the most important asset it is rarely the only asset. Someone or something normally creates information and someone or something transmits and reads it, to finally make use of it. This communication is usually facilitated by or passes through a computer device.

Computer devices are a part of the security system and are also considered assets. These devices may be very complex and can contain vulnerabilities which can be exploited by attackers. It is a difficult task to determine vulnerabilities in a complex system and rarely leads to a complete list. Therefore there is a relation between complexity and security. A heavily complex system will usually contain more loopholes for attackers to exploit and therefore be less secure.

The security definition provided by Stallings in [4] suggests that there is no such thing as perfect security. Security is limited by a cost factor and practicality, if a system is too complicated or cumbersome for the end-user, it does not matter if it's safe. The users will reject it and the system will be neglected, as stated in the definition of security by Stallings. Security systems only need to achieve the security objectives at an affordable level. Perfect or absolute security seems like a pipe dream since security almost always includes assumptions, i.e. a trusted party in communication, or some other weakness. This document aspires to achieve and present a practical and realistic level of security, as opposed to ideal and abstract security.

After discussing the first question it is now time to continue to the second question: ”From whom is the asset being protected?”. Defining an attacker is achieved by creating an attacker model which consists of discovering who the attacker is as well as its capabilities. A computer is almost always used in an attack, but it is only a tool or a target, while the source of the attack is always a human. An attacker is not necessarily a concrete or specific person it can be a group or class of people. To clarify, a system can be secure against attacks from external agents (non-employees), but be vulnerable to attacks by or involving employees. Thus we can observe that security is not absolute, but restricted by a set of assumptions not only relating to the environment but also the attacker and its capabilities.

The attacks mentioned above are directly related to the question “Against what is the asset being protected?” and to the actions mentioned in the security definition presented at the beginning of this section. An attack is any action that compromises the security of protected assets. These compromises are often referred to as threats, and are normally defined in terms of violations of one or more security goals. These security goals are related to the goals mentioned in the security definition confidentiality, integrity and availability, popularly known as CIA.

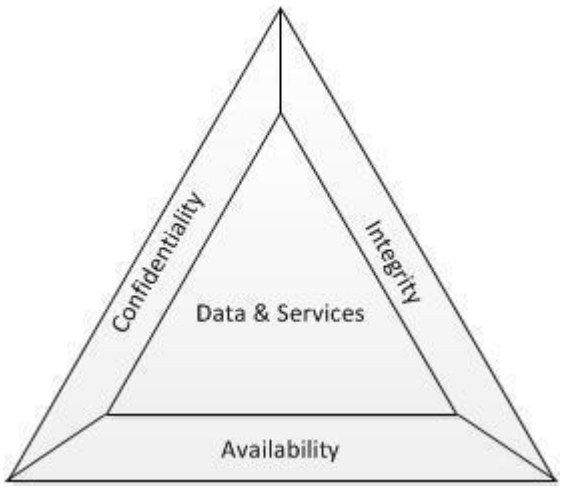

Figure 2.1: The Security Requirements Triad (adapted from [4])

These three goals embody the fundamental security objectives for data, information and computing services. We have implicitly introduced these concepts above when it was stated that is might be required for information to be protected against being revealed (confidentiality), tampered with (integrity) or made unavailable (availability). Further goals can be added or removed depending on the needs of the security system involved. For instance if the information exchanged between entities does not have to be private, the confidentiality goal can be excluded. However if the identity of the parties involved in the communication needs to be protected the goal of privacy can be introduced. Non-repudiation provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication. Finally, accountability is closely related to non-repudiation, but rather than relating to transmitting and receiving of a piece of information, it is related to the execution of an action. Accountability is defined in [4] as "*the requirement of actions of an entity to be traced uniquely to that entity".*

To answer to the fourth question which is *How assets are protected* we need to define a security system. A security system is the concrete way of protecting assets. A security system has achieved its purpose if it prevents all possible attacks, based on the assumptions made of the attackers and the environment. Note that it might be infeasible to create such a system due to limited budgets. Instead of hoping for a perfect system, a process known as risk assessment can be engaged. The purpose of risk assessment it to determine which risks the security system will counter and which risks it will accept, based on the severity of the risks. The assessment consists of a list of risks accompanied by the respective consequences of the risk; these can be categorized and evaluated by level of financial and physical damage. The notion of risk is closely linked to the notion of threat. A risk is the measurable damage inflicted by the occurrence and the probability of a threat. Once the risks have been identified and the threats are mapped to risks, it will present an overview from which it is easier to decide which risks should be mitigated and which ones should be accepted. Normally the biggest risks are mitigated, this does not mean that they are completely eliminated but rather that the probability and the impact of a risk is reduced to an acceptable level, which again highlights our security definition. Risk can be diminished in other ways such as transferring it by relaying the responsibility to insurance companies and some risks can be so small that implementing countermeasures in the security system simply exceeds the costs of accepting the risk.

There are three main approaches a security system can implement to mitigate risks: prevention, detection and recovery. The approaches can be further deconstructed into parts, referred to as countermeasures. The countermeasures may consist of procedures or regulations, but they can also have a technical nature, usually implemented using access control. Access control determines who should have access to what, and is often realized using cryptography.

2.2 Cryptographic terms and concepts

Access control is usually implemented using cryptography as stated in previous section. Because cryptography is a necessity for a security system to achieve its security goals, it is important to establish a baseline of understanding the terms, concepts and primitives used in cryptography. This will not be an exhaustive explanation of all terms and concepts used in cryptography and will be limited to terms and concepts used later in this document.

Hash functions (H) accept a variable-length block of data (M) as input and produce a fixed-size hash value h.

Hash functions are denoted: $h = H(M)$

Symmetric encryption transforms plaintext (M) into cipher text using a secret key (K) and an encryption algorithm (E). Using the same key and a decryption algorithm (D) the plaintext is recovered from the cipher text C.

Symmetric encryption is denoted: $C = E(K, M)$ and $M = D(K, C)$

Asymmetric encryption also known as public key encryption is a cryptosystem in which encryption and decryption is performed using two keys – a public key(PU) and a private key(PR). Public key encryption transforms plaintext into cipher text using one of the two keys and an encryption algorithm. Using the opposing key and a decryption algorithm, the plaintext is recovered from the cipher text. Subscripts are used to specify which party the keys belong to, e.g. PU_A specifies that it is A's public key.

Encryption is denoted: $C = E[PU_A, M]$

Decryption is denoted: $M = D[PR_A, C]$

Message authentication codes (MAC's) also known as a *keyed hash function*, accepts messages of a variable-length message M and a secret key K, and produces a fixed-size output h which is secure against attackers who do not know the secret key. MAC's protect the authenticity and the integrity of the message.

Generation of message authentication codes is denoted:

$$h = E(K,H(M))$$

Signatures are based on much the same concept as message authentication codes however, signatures are generated using asymmetric keys rather than symmetric keys. This enables verification by entities which

only possess one of the two keys composing a key pair. Signatures protect the authenticity and integrity of the data they are generated from.

Generation of signatures is denoted: $h = E[PR_A, H(M)]$

Public key certificate (also just called certificates), consists of a public key, identifier of the key owner, a signature generated by a trusted third party and the third party's identity. This third party is also referred to as the certificate authority (CA). The user A can in a safe manner retrieve his certificate by presenting his public key (PU) and his identity to the CA, which then issues a certificate (C_A). The certificate signature generation is denoted by:

$$C_A = E[PR_{auth}, H(T \,||\, ID_A \,||\, PU_A \,||\, ID_{CA})]$$

Explained as the encryption by the CA's private key of the hash value generated of the data contained within the certificate. H is used to denote a hash function which is calculated over the data within the brackets (). It is used to generate a fixed length representation of the data contained in the certificate. T is a timestamp which validate the certificate. ID_A is the identity of the owner of the certificate and ID_{CA} the identity of the certificate authority. PU_A is the public key of the owner of the certificate and PR_{auth} the private key of the certificate authority. The symbol '||' is used as a separator.

By generating an asymmetrical encrypted hash value of the certificate data also called a signature the integrity and authenticity of the data is secured. Anyone with access to the CA's public key can verify that the certificate has been generated by the CA. This is done by decrypting the signature with the public key and comparing the result to the hash value generated of the data within the certificate.

Public-key infrastructure (PKI) is defined as the infrastructure needed to create, manage, store, distribute, and revoke digital certificates.

It is assumed in the rest of this document that the cryptographic terms and concepts presented above provide perfect secrecy to avoid subjects which are out of scope of this document.

2.3 Access Control

In the context of security systems, access control is the ability to limit and control the access to systems and applications via communication channels, to prevent unauthorized use of a resource. Access control is an overarching term in information security which can be decomposed into three major concepts. Access control is defined as associating an entity with an identity (*identification*), proving the association between entity and identity (*authentication*) and restrict the entity's access or actions based on credentials(*authorization*). This model is related to the concepts that have been presented in section 2.1 as

it allows us to distinguish legitimate users from malicious attackers, and prevent attackers from completing unwanted actions.

In the previous section we have used the term entity it is a very vague term that can mean almost anything. To be able to describe the roles and significance of different entities engaged in access control we relate to Andersons definition in [7]. Anderson separates an entity into two concepts, a subject which is a physical person e.g. a human which can take on any role such as operator, organization, principal or victim. The second concept, a principal is an entity that participates in a security system. A principal can be a subject, a person, a role, equipment or even a communication channel. This is a necessary and important distinction, because it allows us to describe relations between a subject and a principal. A subject is usually authenticated through a principal such as a PC or another form of proxy. Determining the level of trustworthiness of a principal is crucial, especially in a case such as this where new devices such as smartphones are being considered as principals in a security system. A trustworthy principal will not change from its specified behavior and will not aid an attacker in compromising the security system. Whereas an untrustworthy principal, might deviate from its expected behavior allow an attacker to compromise the system and generate unwanted consequences.

The rest of this section will be dedicated to clarifying the three concepts derived from access control, e.g. *Identification*, *authorization* and *authentication*. Identification, is the binding of a principal to an identity in other words, it is when two different names correspond to the same principal. It is important to highlight that normally identification is coupled with authentication to in combination make a statement of an identity and to prove this identity [4], [7]. Even though identification and authentication are used in combination they are two different actions e.g. generating an identity claim and proving it, hence the separation of these concepts in two sections.

Authorization is defined by determining which actions a principal is allowed to perform in the system (*privileges*). Before a principal can be authorized it is necessary to authenticate the principals to assure that the communication is authentic, that is, each of the principals are the ones they claim to be.

2.3.1 Authentication

Authentication is usually divided into two services, peer entity authentication and data origin authentication. Peer entity authentication enables two peers to authenticate each other by providing an association between a principal and an identity. Data origin authentication consists of proving the origin of a piece of information from a specific entity. In peer entity authentication there are two participants: The prover, and the verifier. The prover needs to present “proof” of the association between the principal and identity and the verifier is responsible for verifying the correctness of the proof.

When dealing with authentication systems, there are four essential issues that must be considered: Effectiveness, usability, cost and impersonation attacks. As we have discussed previously in this section it is very difficult to achieve perfect and absolute security and the same applies to authentication, due to technical and non-technical factors. The second, if the authentication system is to be used by humans it is extremely important, to consider how usable it is. Since the authentication system will be a part of a

security system it is also constrained by budgets, as a consequence even though very effective authentication systems do exist they are not always used. Lastly, impersonation attacks in which a (rogue) prover attempts to demonstrate a false identity claim, must be considered. In general, masquerades can be achieved by replaying or relaying valid authentication sequences, during authentication [4].

During peer entity authentication, the prover has to provide information to the verifier which usually takes the form of credentials or items of value to really prove the claim of who the prover is. Kizza states in [5] "*The items of value or credential are based on several unique factors that show something you know, something have, or something you are"*. Furthermore, Kizza adds a fourth less significant and more indirect factor *somewhere you are*. The first authentication factor consists of using a secret which a human subject mentally possesses, or in the case of a device such as a smartcard a key stored in secure memory. This could be a password or a key, which is only known by the prover and the verifier. The secrets have to be hard to guess to avoid guessing attacks such as dictionary attacks and this is why people are encouraged to use difficult to guess passwords. The second factor relies on something the principal has, examples of such tokens, are hardware/software code generators or the more widely used smart cards. This type of authentication is slightly safer than something you know because it is harder to lose a token than a secret. The third factor, something you are, is only applicable to people e.g. subjects and relates to the biometric characteristics of the subject such as voice, fingerprints or iris patterns. The last factor, somewhere you are, usually based on the whereabouts of the subject, such as specific workstations or terminals. It is less secure than any of the previous, and can only be used in combination with other factors but does offer more in terms of usability for the subject.

Regarding to the *data origin authentication*, it can be achieved by using two cryptographic techniques. The first one is based on calculating a MAC over the information using a symmetric key X shared between the entity ascertaining the authenticity of the information and the entity verifying it. The second way consists of using digital signatures in such a way that the data is signed using private key PR of the entity ascertaining its authenticity. Since a MAC is based on symmetric keys, it is significantly faster to calculate, however, it requires the parties to have a shared key. The digital signature combat the key distribution weakness of MAC's by using asymmetric keys, meaning it does not require two parties to have a pre-existing shared key, in order to calculate and verify a signature. Asymmetric keys are usually implemented using public-key cryptography, allowing one party to calculate a signature for a message using a private key and another party to verify the message authenticity and integrity using the corresponding public key.

Authenticating a principal based on the first authentication factor (*something that you know*), can be achieved using three families of mechanisms: Basic authentication, One-time passwords and Challenge-response. Basic authentication consists of a reusable password which is shared between the prover and the verifier, the prover must reveal the password to the verifier to be authenticated. Basic authentication is the simplest authentication mechanism to implement, but also the weakest in terms of security, because they are usually easy to guess and forget and they need to be revealed to the verifier. This leaves the password vulnerable to eavesdropping and masquerades by any entity with access to the communication channel

over which it is sent. However if the channel is secure enough to maintain confidentiality and integrity, then basic authentication can be a simple and secure authentication mechanism.

One-time password (OTP) authentication can unlike reusable passwords only be used once and then disposed of. They are randomly generated using powerful random number generators. The generators utilize one way functions to generate these passwords. The property of one way functions is that they are, in terms of computational complexity, easy to compute but hard to invert [8]. A secret key is used as input to these functions. Time based OTP's rely on changing the input with relation to time, whereas non-time based OTP's usually rely on chaining hash functions together to generate a hash chain. One-time passwords are great at fighting passive attacks such as eavesdropping, however it is still vulnerable to active attacks such as man-in-the-middle, the degree of vulnerability depends on the type of one-time passwords used in the authentication process. A man-in-the-middle attack is illustrated in Figure 2.2 where an attacker A, masquerades as the verifier V to the prover P to obtain an OTP from P. The OTP can then be used to impersonate P to V.

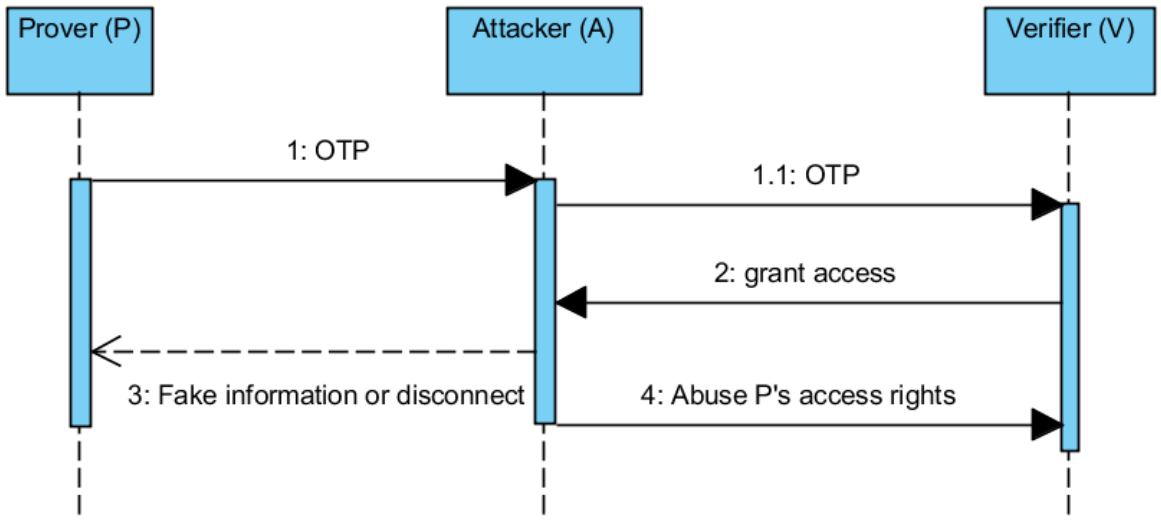

Figure 2.2 An active Man-in-the-middle (MITM) attack against OTP-based authentication scheme.

Time-based OTP's rely on time synchronization between a server and a client to generate passwords which is only usable for a short period of time. To compromise a time-based OTP an attacker has to mount an online real-time MITM attack. The strength of OTP's can be further increased if combined with challenge-response authentication, which will be presented after this.

Challenge-response authentication mechanism consists of the verifier presenting the prover with an unpredictable challenge every time the prover attempts to authenticate. For every challenge there is an associated response that allows the prover to be authenticated if he is able to compute it and send it to the verifier. As a consequence of the unpredictability of the challenge, this family of mechanisms ensures that the entity being authenticated is active when authentication takes place. Hence, passive attacks are not

possible, and even though active attacks remain feasible, they need to be carried out in real time, which means that the time interval between the moment when the attacker captures the response and when it is used needs to be very short for the prover to accept it.

The challenge-response protocol utilizes cryptographic techniques in order to bind challenges to their corresponding responses. There are three main types of challenge-response protocols. The first one is based on *symmetric encryption* in which a shared key *sk* is used by the prover to generate a response *r* by encrypting an unpredictable challenge *c* sent by the verifier. The verifier in turn uses the shared key to decrypt the response in order to check that *r* matches *c*.

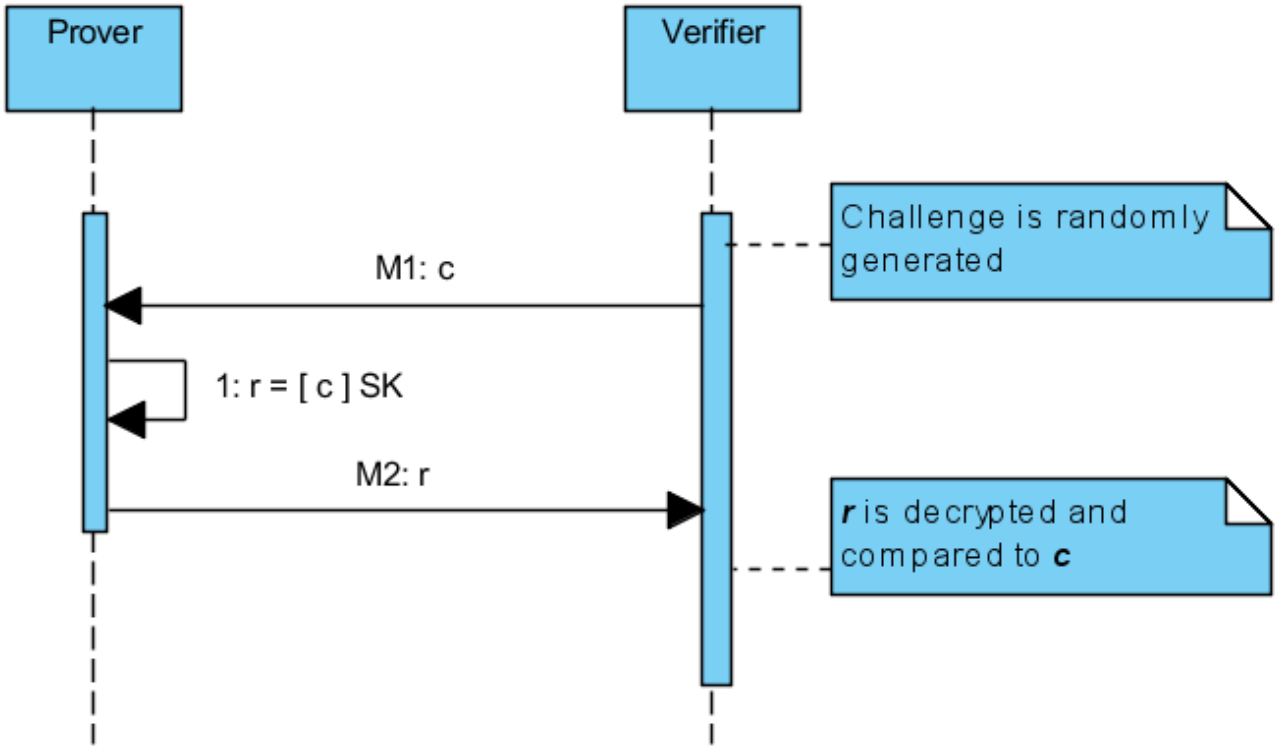

Figure 2.3: Challenge-response protocol using symmetric encryption.

The second type of challenge-response protocol uses asymmetric encryption in which the verifier creates a challenge *c* by encrypting a random value *n* under the prover's public key *PU*. Then, the prover is authenticated by being able to produce *n* as the response, which the prover obtains by decrypting c using the prover's private key *PR*.

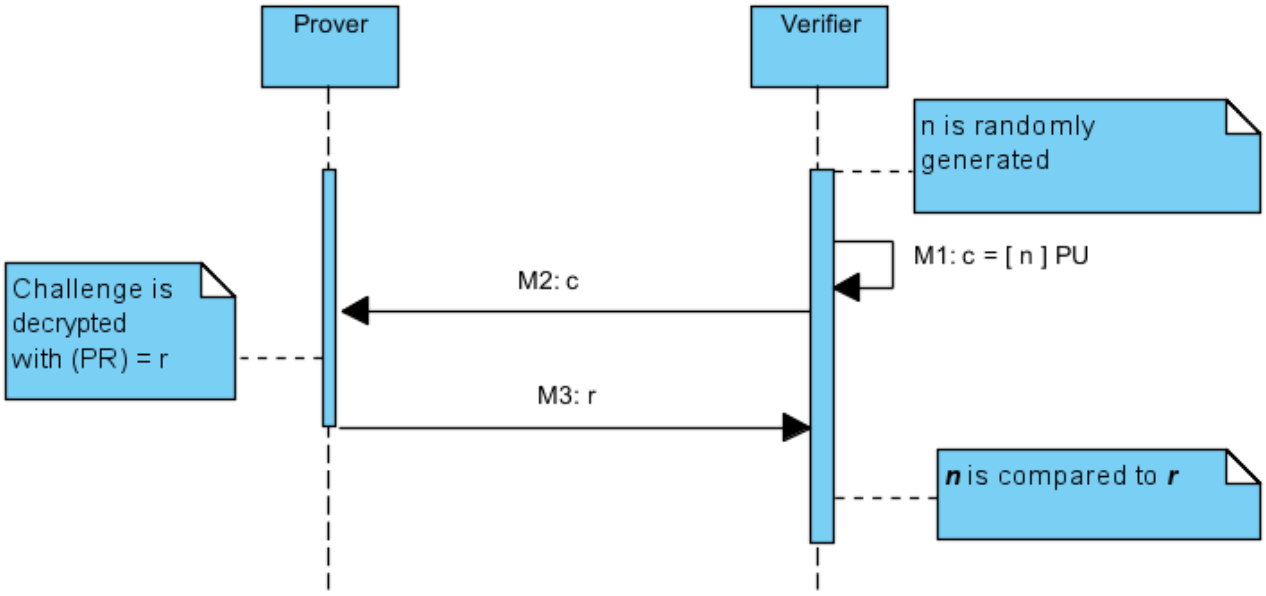

Figure 2.4: Challenge-response protocol using Asymmetric encryption.

It is important to note that similar protocols can be achieved using MACs with symmetric keys or signatures using asymmetric keys.

3 Overview of Mobile Student ID

3.1 Scope

To better understand the context in which the identification needs to work, this section will describe in which scenarios the MS-ID (mobile student ID) will function.

In these scenarios the MS-ID should provide equal or better services than an ordinary student identification card for it to be viable in any form. For this purpose we identify four major services the MS-ID must assist with.

Identification: The MS-ID must be able confirm the identity of a student.

This scenario will allow the student to be identified and approved for participation in an exam by authenticators.

Access control: The MS-ID must be able to distinguish between different levels of access for student facilities.

This scenario will allow a student to access different school facilities such as laboratories and workstations.

Identification during transactions: The MS-ID must be able to confirm if a student is eligible for student rebates.

This scenario will allow a student to take advantage of student rebates provided by stores.

Student registration: The MS-ID must be able to establish a student identity and bind it to a smartphone.

This scenario will allow students to register their identity with their smartphone in a safe way. It is a necessity for the previous scenarios to function.

3.2 Actors

In the scenarios presented above there are three main principals: The user (or student), the school authority and third parties authenticators.

The school authorities are represented by access control terminals and employees, which require identification from the student. The school authority also provides website and server to facilitate the MS-ID.

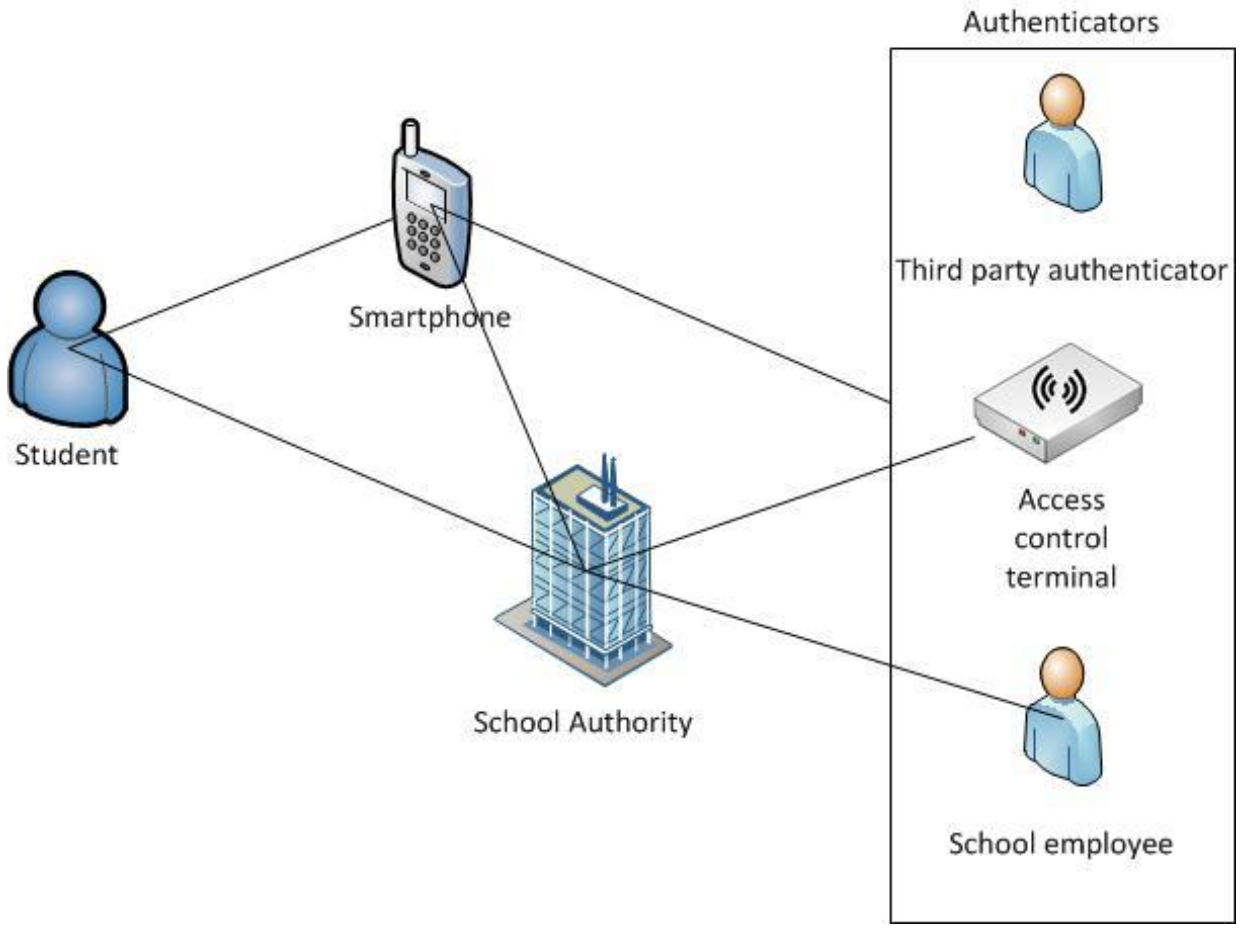

Figure 3.1 Overview of the main principals of the MS-ID scenario.

3.3 Goals

The school authorities want to provide the students with access to schools facilities and resources in a secure and cost-efficient manner, and thereby preventing abuse of these facilities and resources. As a result, both functionality and security are important considerations. The aspect of usability is also highly regarded since the facilities will be used on a daily basis, if it is a hassle it will be neglected. The following goals will be considered in an evaluation of the identification system.

Goal 3.1 (Server authentication) The schools server used for access control must be authenticated by the student. Achieving this goal is necessary because the student releases sensitive information such as passwords, it is a requirement that the student is made sure that she is talking to the schools server and not an adversary posing as the school server.

Goal 3.2 (Student authentication) The student must be authenticated by the school server in order to allow legitimate users to access facilities and to prevent intruders from acting on their behalf. All actors

attempting to access the facilities must be successfully authenticated by the school authority. This also applies during authentication with third party authenticators.

Goal 3.3 (Data authentication) An attacker should not be able to perform actions if she is not a legitimate student. This goal explicitly acknowledges that user authentication is necessary but not sufficient to ensure that actions are only executed by legitimate students. This follows from the fact that after a student has been authenticated, an attacker may take over the session in certain scenarios. Naturally, then it is not only necessary for the authenticator to authenticate the student, but also the transactions that she requests.

Goal 3.4 (Privacy) The information exchanged between a student and the school servers should not be revealed to any third party. Confidentiality is essential in maintaining user trust and confidence in the workings of the system, if there is any doubt that sensitive data is being leaked it will severely hurt the adoption and usage of the system.

Goal 3.5 (Usability) The security system should be usable. Usability is a crucial factor in the effectiveness of a security system and peoples willingness to adopt the system. Since this is a system that will potentially be used several times a day. One of the most crucial factors in usability is how intuitive the user interactions with the system is.

Goal 3.6 (Cost) The cost of the security system should be reasonable. Security systems are constrained by economic factors. This will make us strive to use existing infrastructure and security mechanism. By striving to use existing infrastructure the proposed solution will be more economically viable. Solutions requiring completely new infrastructure will usually require a large initial investment which will increase the cost.

3.4 Threat model

To better understand the security needs of the mobile student ID card it is necessary to describe which attacks a system of this kind will have to endure. To present a meaningful overview we build a threat model based on the threat model process described in [13]. This section will start by presenting an overview of the principals with a description and trust levels of each principal. The following section will describe the possible attackers that might compromise the system. The third section will present the system assets and their importance. The fourth section will describe the attacks the system will have to counter. Finally, the last section will present means of general mitigation of attacks presented in section 3.4.4.

3.4.1 Principals

The attackers or intruders, attack a security system by compromising one or more principals. Figure 3.1 presents an overview of the primary principals and the communication channels among them.

The principals will be described in this section to clarify their purpose and level of trust in the system. The levels of trust which will be used are low, medium and high.

Low trust means that the principal is at high risk of abusing or compromising the assigned assets. Therefore no valuable assets will be transferred to a principal with this trust level. Principals with medium trust level have access to assets that are necessary for them to prove their identity, to in turn gain access to resources provided by SA or 3rd party vendors. High trust is assigned to principals who are authorized to create and assign new credentials.

School authorities (SA) maintain the infrastructure required to operate the MS-ID (Servers, access control terminals etc). The school authorities are at the pinnacle of trust within the system, as they are the maintainers of the system. The SA creates and assigns all new MS-IDs.

The students are the systems primary users and are registered with the SA. The students use their smartphones as the MS-ID platform. The students have a moderate trust level within the system. The students contact information is known to the SA, and as such less likely to compromise the system. The students however can have ill intentions towards the system, such as abusing the MS-ID for malicious purposes. The students can abuse the MS-ID by sharing it with friends to gain rebates or access to otherwise restricted areas.

The smartphone represents the student in communication with other devices and has obviously no intentions of its own. The SA can have embedded keys or information inaccessible for the student within the phones secure memory, as such the trust level of the smartphone is the same or higher than the students (moderate-high).

The authenticator is a categorization of principals with the purpose of authenticating the student. The authenticator role represents multiple principals which all share the same purpose. The authenticators trust levels range from low to high due to the fact that multiple entities can be authenticators depending on the scenarios described in section 3.1.

The school employee will be authenticating the student in the *Identification scenario*, the school employee is hired by the SA and as such is implicitly trusted to handle sensitive information and has a high level of trust.

In the *access control scenario* the student will be authenticated by an access control terminal deployed by the SA. As with the student smartphone the access control terminal has no intentions of its own and therefore inherits trust levels from the principals that control the device. Because the SA controls the access control terminals the trust level is high.

The last scenario involving an authenticator is *identification during transaction*. The student is authenticated by a 3rd party vendor, which has no affiliations with the SA. The vendors are assigned a low trust level, since they are not registered by the SA, it is conceivable that some registration process can be put in place to allow some level of trust to be established between the SA and the vendors.

3.4.2 Attackers

To be able to describe the attackers in this case, we describe attackers in the sense of their location. For purposes of the problem at hand two types of attackers will be considered: Local and remote.

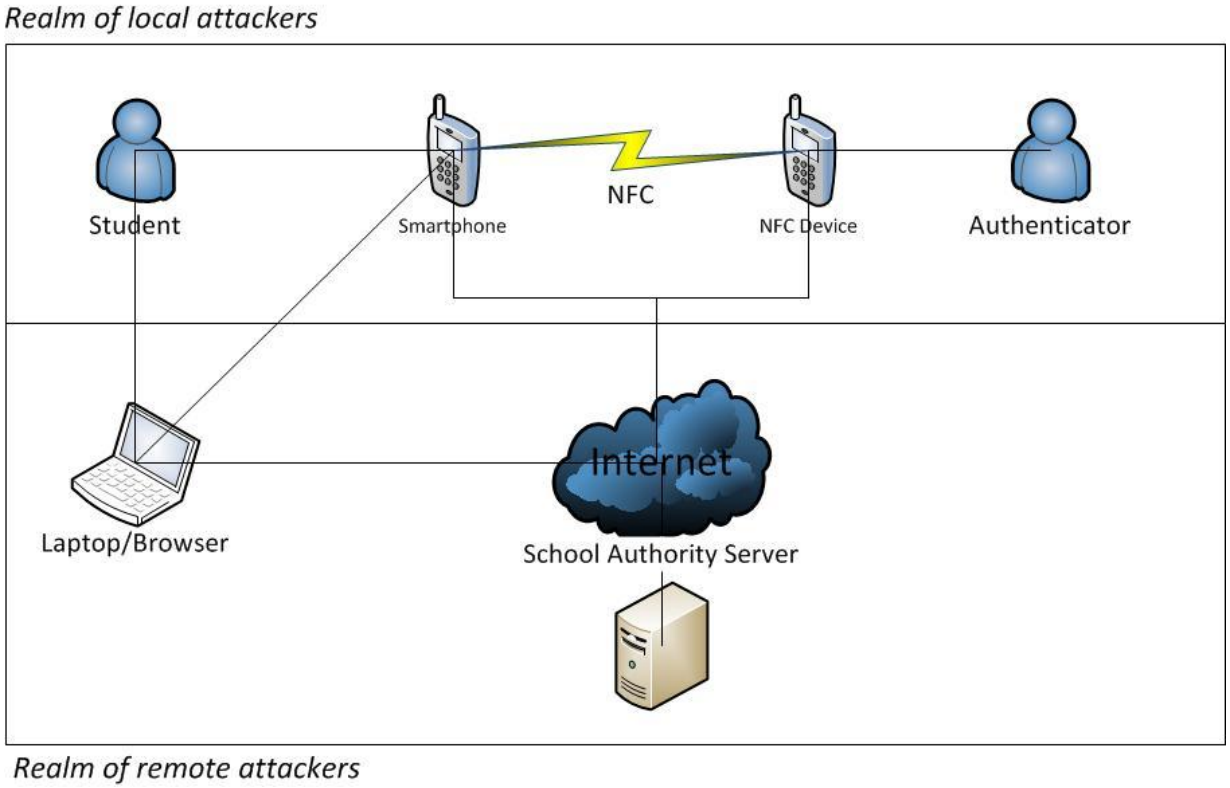

Figure 3.2 The different areas which the two types of attackers pose a threat to.

The local attacker will exploit his possibility of gaining physical access to principals and channels used by system. This will allow him to attempt to compromise the system by stealing or hack a students smartphone. The access control scenario is especially vulnerable to local attackers due to the needed access control equipment, which is susceptible to physical tampering.

The Remote attackers on the other hand operate outside of the physical surroundings of the student, and usually attack using the internet or other remote communication technology. It is important to keep in mind that there are attacks such as Man in the middle (MITM) described in section 2.3.1 which can be initiated locally and remotely. An example of a remotely executed relay attack using two NFC enabled smartphones can be seen here [14].

Normally in security systems the user has no interest in relaying his credentials to others, because the user does not benefit from it and it puts himself at risk. In this case however there is an exception the user can abuse the MS-ID by sharing his credentials with friends and relatives, so they can benefit from special offers to students only. Another example of abuse is cheating by an individual attending an exam on behalf of someone else, by sharing student ID credentials. This is not the case with other security systems such as credit cards. This puts the MS-ID system in a special situation where the student becomes or helps the attacker. The current student ID tackles this issue by adding a picture on to the plastic ID card along with the student details. The validity of the ID is either printed on the card or has to be provided separately by the student.

3.4.3 Assets

In any application which maintains or handles identities, the protection of the users and the systems assets are essential to the success of the application. This section will clarify which assets are important to protect in this context and provide a description of them.

Student MS-ID

The Student MS-ID asset facilitates the authentication of the owner of the student MS-ID in question. This asset contains of the digital content of the MS-ID and the knowledge of the related password. The full list of assets contained by the Student MS-ID will be presented in the design and implementation chapters of this document.

Student credentials

The student credentials are the student number and password used by the student to gain access to the school websites. Through these websites it is possible for the student to manage practically all relations the student has to the school. The options offered on the school website include signing up for courses, changing student identification image, reading sensible data about the student and even the option of discontinuing the education at the school.

Student privacy information

The current student identification card has the students central person register (CPR) number printed on the front of it. This number is used in many situations in the community as the only item to authenticate a person and is therefore a very sensitive piece of information.

Security tokens

These tokens are certificates, signatures, passwords and keys used both in communication and authentication. They can be stored on principals in the system in order for that principal to engage in authentication or secure communication. The tokens can be stored on the principals in advance, be

generated by the principal or be exchanged, to enable authentication by knowledge of that token. For two principals to engage in secure communication on a public channel, a security token can be transferred between the two principals on a secure channel. The security token can then provide confidentiality and authenticity on the public channel. All tokens stored in the system which are not deliberately made public must be securely stored in the principal in possession of that token.

Access to school facilities

The system facilitates access to the school facilities and has ability to allow access to authenticated and authorized students. If the system is compromised attackers may gain access, enabling theft or gathering of sensitive information. Attacking the system in order to compromise access control may target the access control terminal or the SA server.

Some of these assets are interconnected in such a way that, if one is compromised it may automatically compromise other assets. An example of this is if an encryption token or a MS-ID is compromised it may lead to access to school facilities being compromised.

3.4.4 Threats

This section will explain the threats posed by attackers to the assets described in the previous section. The threats described below will refer to the principals which they threaten.

Eavesdropping

The threat of eavesdropping is passive in the sense that an attack does not change data that is flowing through the channel. This makes eavesdropping very hard to detect which can result in channels being compromised for a long time without channel participants or administrators knowing about it. Wireless communication channels are relatively easy victims to eavesdropping, as any attacker within range can listen in on the communication if the data is not encrypted. The wireless technology WIFI, which is used excessively to connect electronic devices to the Internet, can enforce encryption on the network. The latest protocols WEP, WPA, and WPA2 have unfortunately been cracked [52], which means that network security over WIFI must rely on encryption in higher level protocols to ensure protection against eavesdropping. Communication over electronic channels are not the only principals vulnerable to eavesdropping, an attacker might also be able to observe a user's password while the user inputs it. Closed networks that transmit data through cables are often harder to eavesdrop because an intruder must have gained physical access to the cable or have gained control over a principal in the communication.

Traffic analysis

Another passive threat is traffic analysis where an attacker gathers information from analyzing, not the content of the traffic but, indirect information regarding the transmission such as message length, size, pattern and frequency. This attack only occurs when an attacker is unable to determine the contents of message transmissions.

Active threats involve actual modification and changes on data passing between two communicating parties or actual impersonation.

Masquerade

This attack requires that an attacker impersonates or pretends to be another entity. The masquerade threat is usually combined with other forms of active attacks such as replay, or the more sophisticated man in the middle attack. Phishing and Spoofing are both attacks that use masquerading to gain information in an illegitimate way. Attacks of this kind can both be targeting devices on a network or be targeting people. To accomplish an attack on people an attacker will be masquerading as a trusted entity, this can e.g. be a bank or government. By masking his true identity the intruder can lure the victim into revealing sensitive information or install malicious software. This attack is often done through e-mail or social sites but can also take place in application markets where an application with a name and icon resembling that of a trusted entity can easily be mistaken for legitimate application.

Replay

This attack involves the capture of an authentication sequence that has taken place, thus enabling an unauthorized user to gain access, by retransmitting the captured sequence. Replay can also be exploited when the validity of a message transmitted between two parties is not bound to a specific session. An attacker may then send the same message at a later time to gain access or influence the system in other ways.

Denial of service

The purpose of this attack is to prevent or limit normal use of a service or resource. There are different approaches to deny a service. It can be done by suppressing all messages to the target destination, disabling the network or by overloading the network.

Man in the middle

This attack uses Masquerade to place the attacker between two communicating parties without them knowing. In a successful MITM attack the attacker intercepts all messages which he then transmits to the intended receiver. Depending on the protocol being attacked the attacker may be able to expose and modify the messages transmitted between the victims. The specialized case of the MITM attack where the attacker does not expose or alter the messages but only benefits from sending the messages to another recipient than originally intended, is called relay attacks. Wireless technology which can be initiated without user confirmation, such as RFID and NFC tags are especially vulnerable to relay attacks because they can be read wirelessly by an attacker without the owner's knowledge. If the tag is used for access control the attacker can relay the communication between the access control terminal and the tag and gain access.

Theft

Theft is always a threat to security systems but especially system that uses mobile equipment which cannot be locked away when in use. Mobile phones have always been a target for criminals but within the last decade the high-end mobile phones have become almost as expensive as laptops making them an even more valuable target.

3.4.5 General mitigation

This section presents a brief overview of the general methods and concepts involved in mitigating the threats presented in the previous section. Technologies enabling the methods and concepts are listed to give the reader an impression of the solutions available.

Obscurity

Systems often incorporate security by obscuring how the system works, also referred to as "Security through obscurity". This is in general considered an unreliable method to secure a system and its assets. Because the secret of how the system works is too unstable to keep hidden, attackers have the possibility of analyzing the system behavior and reverse engineer it. Another critical property of security through obscurity is that if the system is compromised it is complicated to counter the threat.

Encryption

Encryption serves to hide the information of a message from anyone but the intended receiver by changing the original message into a cipher text. The term is explained in depth in section 2.2. Some of the most used encryption methods are for symmetric encryption Advanced Encryption Standard (AES) and its predecessor Data Encryption Standard (DES), for asymmetric encryption the defacto standards are RSA and elliptic curve (ECC)[4].

Authentication

Authentication provides insurance that the entity which is authenticated does have the claimed identity. The term is explained in depth in section 2.3.1. Authentication through knowledge is often proven by passwords or pin-codes which are not subject to particular standards. In authentication through possession, methods and concepts used depend highly on the principals used but usually involve an identification card with any number of identification technologies embedded. Identification cards often implement technologies such as embedded smart card, RFID chip, barcode, magnetic stripe or simply a photo. The layout and appearance of the identification card can also be a contributing factor in an authentication. Authentication through possession can also involve one-time-password (OTP) printed on a card or displayed on an OTP generator. Certificates described in section 2.2 are widely used in authentication especially through the X.509 standard. Authentication can also be granted on the premise of biometric data such as fingerprint or facial photos.

Storage

Secure storage is an important part to thwart the threats mentioned in the previous section. Secrets that are not suitable for humans to remember or result in inconvenience if needed entering, can be stored in secure storage. Specific implementations of secure storage are smart cards or secure elements presented in section 4.4. Smart cards are widely used in access control systems while secure elements in smartphones have not yet shown widespread utilization in third party applications.

4 Technology and Standards

This chapter will contain relevant technologies and standards used for mobile authentication, focusing on those used by the proposed solution presented in chapter 6.

4.1 Smartphones

Smartphones distinct themselves from ordinary feature mobile phones by not just accommodating applications running on platforms such as Java ME. Smartphones provide a complete operating system which enables the execution of more advanced applications [23]. The Smartphone also provide user controlled software which is able to communicate with external systems [42].

Mobile phones have evolved from being a device of voice conversation to a powerful mobile computer which offers users data connectivity but also a wide variety of technologies such as cameras, GPS and near field communication. Starting out as the result of mating between the PDA (personal digital assistant) and the older feature phone, the smartphone has become an important part of many people's everyday life. In a report done by the American research institute ComScope they estimate that 43% of the Danish population owned a smartphone at the beginning of 2012. If smartphones follow the same adoption pattern as other electronic devices it is conceivable that a much higher percentage of the young population own a smartphone.

The smartphone market consists of several hardware and software vendors, in Denmark the big ones are Apple with their iPhone product and IOS operation system and several hardware vendors with the same operation system Android. In a report published by the information technology research and advisory company Gartner the worldwide sales of mobile devices is shown according to operation system. This diversity in operating systems increases the cost of deploying an application which must be available for a large percentage of the smartphone owners. Despite this, the smartphone application development has exploded the last years, now offering users a variety of services, varying from live football results to banking transactions.

Worldwide Mobile Device Sales to End Users by Operating System in 3Q12 (Thousands of Units)

Operating System	3Q12 Units	3Q12 Market Share (%)	3Q11 Units	3Q11 Market Share (%)
Android	122,480.0	72.4	60,490.4	52.5
iOS	23,550.3	13.9	17,295.3	15.0
Research In Motion	8,946.8	5.3	12,701.1	11.0
Bada	5,054.7	3.0	2,478.5	2.2
Symbian	4,404.9	2.6	19,500.1	16.9
Microsoft	4,058.2	2.4	1,701.9	1.5
Others	683.7	0.4	1,018.1	0.9
Total	**169,178.6**	**100.0**	**115,185.4**	**100.0**

Source: Gartner (November 2012)

Figure 4.1 World wide sales of mobile devices with the specified operating system in the third quarter of 2012 [12]

Assets contained by the smartphone become more and more valuable as smartphones are being used to access banking services and contain credit card services. These assets make the smartphone a high value target for criminals and provide the incentive to recognize the need for high security. The physical dimensions, user interaction and the interfaces provided by the smartphones present several threats. These threats might take the same form as threats known to compromise personal computers but may also be unique to the smartphone platform.

Attackers might benefit from some users negligence or unawareness towards threats against smartphones. This attitude towards smartphone security might originate from the conception that feature phones are not as vulnerable when it comes to security. Feature mobile phones may have revealed the users personal contacts and their communication history while also giving the attacker the possibility of exploiting the phone for calls when compromised. However, the smartphone can if compromised reveal considerable more information about the user and may be exploited in ways that the feature phone could not. As the smartphone inherits the functionality of the feature phone e.g. telephone communication the security assumptions and expectations may very well also have been inherited by some users. This will have imprinted security expectations in the minds of those users that the smartphone cannot honor. Therefore smartphone systems and applications must put security against user negligence in special regard.

In the same way as personal computers, smartphones can be attacked by phishing, malware, Break-in, wireless network attacks or attacks aimed directly for the availability of the device like denial of service attacks as mentioned in [23]. Because of the mobility and size of smartphones they are easily lost or forgotten, compromising the device if sufficient security measures have not been enabled. Giving an attacker the possibility of having physical interaction with a lost smartphone may not only compromise the data contained at the time but also enable him to collect and divulge data from the device in the future. This is not a threat that is necessarily limited to smartphones but made more likely because the user pattern of the smartphone is different than that of a personal computer.

Countermeasures have been taken to strengthen the security of smartphones by installing anti-virus applications which supervise the system and allows the user to lock the smartphone remotely when lost. The drawback of improving security with security applications is that the user must see the need and possibility of adopting such measures before taking the trouble of installing these.

The smartphone as a platform offers many technologies with simply APIs such as sensors and network controllers, which enables developers to create applications which incorporate these technologies. This allows a developer to deploy an application which requires otherwise expensive hardware rather cheaply by utilizing the fact that they already exist in the smartphone. Below this section there is a list of the most relevant technologies supported by most smartphones for the needs of this document:

4.1.1 Networking technologies

Smartphones offer a wide range of communication technologies these all have their individual characteristics that makes them suitable for certain situations. GSM is a technology inherited from the mobile telephone it started out supporting voice communication and a very low bandwidth data connection. This technology has later been improved with the enhancements GPRS and EDGE which provide packet data capabilities and higher data rates. The third generation cellular system includes support for GPRS and EDGE but improves the data rates so that most networks offer up to 14.0Mbit/s downlink. While this technology is the leading cellular system today offered by all mobile network operators (MNO) and supported by all smartphones, the fourth generation system (4G) is on the steps. 4G network is supported by new smartphones but first expected to be provided by MNOs in Denmark in the late 2012. This new generation cellular system will offer peak data rates of up to 300Mbit/s downlink and 75Mbit/s uplink. Wi-Fi is another network technology which is widespread offering high data rates and ranges of about 20 meters indoors. This makes Wi-Fi provide full signal which makes Wi-Fi provide a stable connection when the smartphone is indoors, where the third generation cellular system is rather weak. The different technologies all have different coverage and/or services which make them complement each other so that the smartphone can provide the user with a somewhat stable network connection at all time.

Bluetooth

Bluetooth is a mid-range communication technology which enables sending messages between the smartphone and surrounding devices. Existing in many versions and supporting different modes of communication makes Bluetooth a very agile protocol supporting both high data rates and low energy consumption. The technology was originally made to replace cables from the PC to the surrounding equipment but has found uses in communication among embedded devices. Because of the importance of this technology to the making of this document it is described further in section 4.3.

NFC

NFC is a very short range communication technology which offers embedded devices to transmit and receive data over distances of about 10 cm without the need for any configuration. This technology is of great importance to the making of this document and has therefore been dedicated its own section for a more in depth explanation, section 4.2.

4.1.2 Other hardware

Camera

High resolution cameras are a must-have in all smartphones today many of which have both a camera on the front and on the back of the phone. These can naturally be used to take pictures and capture video but maybe more useful for developers the cameras can also be used to capture text information through character recognition software on the phone. The newest smartphones also have embedded facial recognition software which can distinguish faces in captured photos. The most widespread functionality of the camera apart from the traditional uses is the ability to capture digital content, from 2D barcodes. This feature is used by more than half of the smartphone users in Denmark [10] and enables the user to easily transfer relatively small amounts of data to her smartphone. Most often the data includes a link which is presented on the smartphone and thereby saves the user the hassle of inputting this information through the keypad.

Positioning

GPS changed the game of outdoor location awareness, earlier GSM was used to get a course grained location by measuring the nearest antenna but after the introduction of the Global Positioning System a very high accuracy of down to 1m is possible. GPS is not designed to work indoors where it preforms poorly or not at all. Indoors the smartphone can use the knowledge of nearby Wi-Fi networks to establish a rather accurate position. This of course depends on the number of Wi-Fi networks in the area and the age of these networks. This is because the method depends on the last information gathered in the area by smartphones. The vendors built a mechanism into the smartphones which provides them with anonymous data about your location and the Wi-Fi networks around you [11]. The mechanism extends the location awareness into buildings, while at the same time guaranteeing no level of availability or accuracy.

Gyroscope, Accelerometer and Magnetic sensors

Sensors are fitted in most smartphones today, enabling the phones to read their orientation and force impact. Many of the newer phones include all three sensors which increases the accuracy of the measurements.

4.2 Near Field Communication

This section presents an overview of the NFC technology, in section 4.2.1 the NFC protocol will be presented, hereafter in section 4.2.2 the utilization of NFC in current smartphones will be presented and in section 4.2.3 the security of NFC is explained. The main sources used to form this chapter are [33, 34, 41, 45, 46].

Near Field Communication (NFC) is a short range wireless communication technology which is promoted by the NFC Forum which was formed in 1983. The purpose of the NFC forum was to “enable the use of touch-based interactions in consumer electronics, mobile devices, PCs, smart objects and for payment purposes”

[17]. The physical layer of NFC NFCIP-1 has since been standardized in 2004 in ISO 18092 which is compatible the previous standard ISO 14443. Later the NFCIP-2 has been standardized in ISO 21481 which defines the selection mechanism between different technologies at 13.56 MHz. On top of the NFCIP-1 the NFC Forum has published technical specifications on the higher level communication between NFC devices to benefit interoperability. With the introduction of NFC enabled smartphones, developers have been given a platform with many capabilities to run their applications on.

NFC offers its users an intuitive approach to exchange of information. When the user wants information from some NFC enabled source she only needs to bring her NFC enabled device in contact with that source and the content is transferred to her device. The same procedure is applied when the user wants to push information to another NFC device. This seamless and intuitive data exchange is only possible because NFC does not require any configuration such as other wireless communication technologies such as Wi-Fi or Bluetooth.

RFID which is considered as NFC predecessor considers participants in the communication as either a RFID reader or as transponder, which is a storage entity also referred to as a tag. This technology is used in many industrial applications especially to identify products from one another. Because the NFC technology is compatible with the RFID standard ISO 14443 many existing systems can be utilized by NFC enabled devices. Entities in NFC communication are referred to as peers this is because they can behave as both passive storage entities and as active reader/writers depending on which mode they are communicating in. NFC incorporates 3 modes of communication Reader/Writer mode, Card Emulation mode and Peer-to-Peer mode, a description follows below.

Reader/Writer mode: In this mode the NFC device takes the role as a RFID reader. The device transmits a continuous signal which enables transponders in close proximity to communicate with the device by load modulation. This mode enables NFC devices to communicate with passive RFID tags which power their chip by inducing a current from the received signal.

Card Emulation mode: This mode enables the NFC devices to emulate a RFID transponder and thereby allowing the device to communicate with a RFID Reader. This can be utilized to authenticate the device in many existing systems which incorporate the ISO 14443 standard. Communication between two NFC enabled devices is also provided by this mode in cooperation with the Reader/Writer mode. When communicating with another NFC device one part may take the role as a RFID Reader while the other takes the role as transponder. This allows two NFC devices to communicate while at the same time offering a skewed energy consumption policy between the two participants. The energy consumption of the transponder role is far less than that of the RFID Reader role because the transponder only needs to generate a load modulation on top of the existing signals from the RFID Reader.

Peer-to-Peer: In the Peer-to-Peer communication mode the two NFC devices can either take the role as NFC initiator or as NFC target. Both parties take turn sending information to the opposite part by turning on their NFC signal to transmit and turning it off to receive. As oppose to the behavior of the NFC target in the other communications mode the NFC target in Peer-to-Peer mode does not transfer information to the initiator by load modulation. The target activates its own transmitter while the initiator switches into receiver mode. The roles as NFC initiator and NFC target are assigned at the beginning of the protocol; the NFC device which activates its transmitter is the NFC initiator while the device receiving the

signal is assigned the role of NFC target. These roles govern the sequence of the messages exchanged by the two devices. The NFC initiator must begin the communication while the NFC target may only communicate to the NFC initiator by replying to received messages. Information transmitted between devices in this mode is contained in NFC Data Exchange Format (NDEF). Peer-to-Peer mode is also referred to as Active mode because both parties use their self-generated magnetic field to transmit as oppose to Passive mode where one party utilizes a load modulation.

4.2.1 NFC Protocol

NFC has published technical specifications concerning the data exchange format, data types, data exchange protocol, link protocol and NFC tag operations. Here follows a description of the most important specifications provided by NFC Forum.

NFC Data Exchange Format (NDEF) [47] is a specification of the format in which data exchanged by NFC Forum devices. This format is built upon messages which encapsulate one or more records that contain a payload described by a type, a length and optionally an identifier. Supported forms for the record type field are NFC Forum well-known types, NFC Forum external types, absolute URIs and MIME media-type constructs. NDEF also specifies a mechanism to build unique NDEF record type names.

Simple NDEF Exchange Protocol (SNEP) [49] is an application level communication protocol which specifies how two NFC devices should send and receive NDEF messages. SNEP is a request/response protocol, the client sends request messages and the server answers with response messages. In the NFC Forum architecture, SNEP is located on top of the Logical Link Control Protocol (LLCP) in the protocol stack.

Logical link control protocol (LLCP) [50] is responsible for the upper half of the Data Link layer in the well-known Open System Interconnection (OSI) model. The Media Access Control (MAC) is responsible for the lower half of the Data Link layer accessed by the LLCP by a set of mappings specifying the binding requirements. One of the main features of LLCP is Link Activation, Management and Deactivation which specifies how two NFC Forum devices recognize compatible LLCP implementations, establish a link, manage and deactivates it. LLCP also provides the Asynchronous Balanced Communication which offers a communication protocol separate from Normal Response Mode. Asynchronous Balanced Mode (ABM) liberates peers from being bound to a master/slave relation where the Initiator only is allowed to send data as a response to a request from the Target. In ABM both peers may send information at any time. LLCP facilitates both connectionless and connection-oriented transport.

4.2.2 Utilization

Unfortunately not all NFC operating modes are made available for applications in their specified state on current smartphones. At the time of writing, the latest version of the Android^{tm} operating system does not offer the full features of Peer-to-Peer mode but instead the Android Beam interface. The Android Beam interface [64] is a reduced version of Peer-to-Peer mode which lets the two devices exchange one message per connection request. The user input which is read after the two devices detect each other determines the direction of the message. If more messages are needed exchanged, a new connection must be established by bringing the devices out of reach and then back together.

The authors of article [19], present a secure credit transfer application for android based platform with NFC. The authors circumvent the limitations of the android API by using NFC to setup a bluetooth connection.

Further limitations are seen with regard to developers access to NFC Card Emulation mode in smartphones. This feature of NFC is on current smartphones only available from a embedded secure execution environment called the secure element (SE) which is further described in section 4.4. This means that the NFC card emulation mode can only be used if the application has access to the SE which requires an agreement with the owner of the SE.

4.2.3 Security

NFC benefits highly from its low range when it comes to security against attacks such as eavesdropping. The mode in which the peers communicate is also a vital factor in the plausibility of an eavesdropping attack. Two peers communicating in Active mode are vulnerable to eavesdrop at a larger range then if Passive mode was used. In the article [46] a rough estimate is given regarding the possible range in which an eavesdrop attack can occur, the article states a range of up to 10 meters when transmitting in active mode and 1 meter when in passive mode. NFC does not implement any encryption scheme which makes transmitted data visible to eavesdroppers without the need for further processing. If a channel needs to be protected against eavesdropping the necessary countermeasures need to be implemented in higher level protocols.

A MITM attack against NFC is not practically feasible because one cannot send data to the attacked parties individually without the other party also hearing the received data. When an attacker has intercepted and blocked a message from A to B and then starts sending a new packet to B, A will receive the message and recognized the problem in protocol. The blocking of the first message from A to B could also be detected by A which, if A is listening while sending, it would make A stop the protocol.

Data modification is a threat to NFC communication, the attacker will in the simplest form of data modification have the intent to disturb the communication in such a way that the receiver is not able to recognize the data send by the opposite part. Disturbing the NFC communication in such a way is realizable by transmitting high amplitude noise in the operation frequency of NFC. In the other case of the Data modification attack it is the intent of the intruder to alter the transmitted data without destroying its validity. The exploits that can be made available by data modification attacks highly depend on the coding scheme used between the two devices. NFC uses the two coding schemes modified Miller and Manchester which are used depending on the baud rate. Modified Miller is use by devices in active mode at a baud rate of 106 kBaud or lower at higher baud rates the Manchester coding scheme is used as well as in all cases for passive devices. Because of the difference in the way the two schemes modulate bits the Modified Miller is only sensitive to attacks which try to alter a bit of value 1 to a bit of value 0 prerequisite the preceding bit is of value 1. The Manchester coding scheme is sensitive to attacks which alter bits in despite their original value and without requirements to the preceding bit. Precautions which could detect an attack by data modification could be made by the transmitter by listening for alterations while transmitting.

4.3 Bluetooth

For a more comprehensive description of the Bluetooth[tm] technology the authors refer to the main sources of this section [61, 62, 63]. The Bluetooth technology is a short-range wireless communication technology which is designed to replace cable between electronic devices. Since its formal introduction by the Bluetooth Special Interest Group (SIG) in 1998 the technology has gone through many revisions and enhancements. Bluetooth strives to offer a robust, low power consumption and low cost wireless technology. Transmitting in the 2.4 GHz ISM band Bluetooth implements frequency hopping to reduce interference and fading. Bluetooth supports different operations which offer max data rates from 721.2 kbps to 24 Mbps.

The latest version of Bluetooth (Bluetooth Version 4.0) offers two forms of wireless technology systems, one which can achieve high data rates the Basic Rate/Enhanced Data Rate (BR/EDR) and one which provides better features for devices that require lower energy consumption the Low Energy (LE). BR/EDR is compatible with Bluetooth versions back to 1.2 while LE is only forward compatible from version 4.0. An implementation of the Bluetooth technology may support only BR/EDR, only LE operations or both in the primary controller of the Bluetooth core system. In addition the system may support any number of secondary controllers in the case that the primary controller supports BR/EDR. A secondary controller provide an alternative communication channel such as 802.11. Communication may then be moved between the BR/EDR Controller and the secondary controller as requirements to the channel changes.

Bluetooth provides in addition to a reliable communication channel also a set of well-defined profiles that devices can implement in order to make interoperability between products from different manufactures easier. For Bluetooth connection to commence there must exist a master/slave relationship between the participants. A network of connected Bluetooth devices is called a piconet and must include one master device and up to seven slave devices. Many piconets can be interconnected to form a scatternet which can distinguish device number limitations in piconets and range constraints. The master device in a piconet acts as the moderator in the communication between a slave and the master and among the slaves.

For two devices to communicate through Bluetooth they must first be paired, originally this could only be done by entering the same PIN in the connecting devices. Devices that did not have input capabilities were forced to have a fixed PIN which made them more vulnerable to attacks. The procedure was also found very cumbersome by users which limited the usability of the technology. As a countermeasure Bluetooth version 2.1 and the versions since have been specified to support Secure Simple Paring (SSP) [65]. The SSP protocols primary goal is to simplify the pairing procedure the secondary goal is to improve the security of the technology. Communication which relied on the earlier pairing procedures often resulted in connections that were vulnerable to eavesdropping and Man-In-The-Middle attacks. SSP offers four pairing models described here:

The Numeric Comparison model is an option in the pairing procedure if the devices in question are capable of displaying a six digit number and have support for user confirmation. In input method one device starts connecting to the other, then the same six digit number is shown on both devices and the users confirm that the numbers are the same. If the numbers are not compared by the users the connection could be subject to a MITM attack.

The Just Works model (JW) works without any authentication it requires no input or output capabilities to the user. The JW method is convenient where devices such as Bluetooth headsets without displays, should be connected to phones. If the two devices equipped with displays the users may be prompted to acknowledge the connection. Against passive eavesdropping the JW model is secure because cryptographic information is exchanged which ensures confidentiality. The model however is vulnerable against MITM attacks.

The Out of Band model builds upon the exchange of connection information through a separate channel other than Bluetooth. The devices which are desired paired must both support the OOB mechanism as well as the technology used in the transfer. NFC is a candidate to this separate channel because it is resistant to MITM attacks, which is a requirement. On the separate channel the device address together with cryptographic data used for authentication is exchanged. If the pairing is done using NFC, the users must first bring their devices into contact where after they are prompted on their respective devices to confirm the connection. This model allows for a connection which is secure against eavesdropping as well as MITM.

The Passkey Entry model does resemble the former legacy pairing method. The model is applicable when one device offers the capability of inputting a six digit number while the other device has the capability of displaying such a number. In contrast to the PIN entry model used by Bluetooth Core Specification 2.0 and earlier versions, the Passkey Entry does thwart attackers trying to eavesdrop the communication by acquiring the PIN. The model also resists MITM attacks.

4.4 Secure element

Reveilhac and Pasquet define a secure element in [15] as the following: "*The secure element (SE) is a combination of hardware, software, interfaces and protocols embedded in a mobile handset, which enable secure storage and provides a secure area for the execution of applications and protection of assets such as payment data and keys*". Secure elements are particularly relevant for security applications because they offer tamper resistant storage of cryptographic keys, while simultaneously allowing them to be used for cryptographic operations. Tamper resistance means that the SE provides mechanisms which make it difficult, but not impossible for an attacker to compromise the data and security keys on the SE. Analysis of SE security and examples of possible attacks on NFC enabled devices with embedded secure elements can be found in [14].

Without a secure element, applications have limited options of secure data storage for sensitive information such as cryptographic keys. The easiest data storage is to simply write the data in along with the application code, this technique however is very vulnerable to disassembling, which allows attacker to retrieve the sensitive data in application code. A far worse consequence of the above technique is cloning or reuse of the application on another device, since the data is in the application code, the application can simply be copied and used on another device[43]. Another option is to rely on remote storage and a secure channel to transfer and retrieve the sensitive data; this however can be a complex and expensive task which limits the usability of the application by requiring online connectivity.

Secure elements share the same hardware, software platforms and security standards as regular smartcards. As stated in [14] “Typical secure elements – like NXP’s SmartMX – are standard smartcard ICs as used for contact and contactless smartcards. The only difference is the interface they provide”. There are three major interfaces which secure elements utilize. The standard contact smartcard supports only ISO 7816 which is the common standard for contact cards[32], contactless cards support ISO 14443[33], whereas secure elements utilize Single Wire Protocol [41](SWP) for connection to the NFC controller while supporting ISO 7816 during communication with the application processor. The SWP protocol is standardized by the European Telecommunications Standards Institute (ETSI) which defines the physical and data link layer between the SE and the NFC controller.

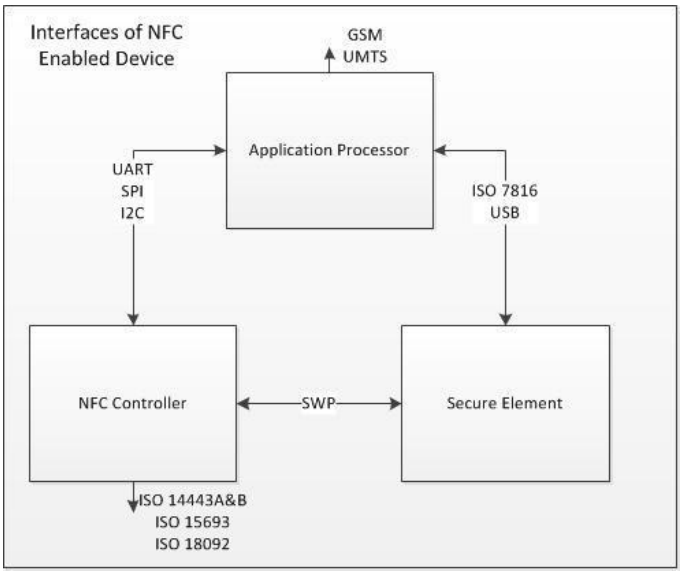

Figure 4.2 Interfaces of NFC enabled device (adapted from [15] & [17])

A secure element is a requirement for an NFC device to perform card emulation. The purpose of a dedicated connection between the NFC controller and the SE is to facilitate card emulation (mentioned in section 4.2), without having to pass data through a non-secure application processor. This illustrates the main difference between a normal smart card and a secure element.

There are many different types of secure elements ranging from removable to non-removable. For a comprehensive and detailed walkthrough of different secure element types we refer to [15] and [16]. In the following section we will focus on the most relevant types of SE’s. The three most common types of SE’s are embedded hardware, secure memory cards (SMC) and Universal integrated circuit cards (UICC).

The *embedded hardware* is usually a standard smartcard which is built into the phone during the manufacturing process. This means the embedded SE has the same level of security as a normal smartcard, and supports the smartcard standards such as Javacard, ISO 7816 and Global Platform. However the embedded hardware is fully controlled by the manufacturer, which limits its ease of deployment.

Secure Memory Card (SMC) is a combination of a memory card (SD card) and a smartcard, which supports the same standards as the embedded hardware SE. The SMC offers large capacity which can host numerous

applications, and unlike the embedded hardware it is removable and can be inserted into any device with a memory card slot.

Universal integrated Circuit Card (UICC) is a generic platform for smart card applications standardized by the ETSI Technical Committee Smart Card Platform (ETSI TC SCP). It can like its SMC counterpart store multiple applications, from different application providers. The main difference between UICC and SMC is the fact that UICC can host GSM/UMTS applications allowing the mobile network operators (MNOs) to provide over-the-air (OTA) management of the secure element. The UICC is in most cases already deployed on most phones.

4.4.1 Smartcard architecture

To better understand how a smartcard operates and what possibilities are available regarding smart cards in security applications this section will provide an overview of a common smartcard architecture.

GlobalPlatform is an organization that has been established in 1999 by leading companies in the industry, to promote a global infrastructure for smartcards. The GlobalPlatform standards goal is to grow the use of multi application smart cards and provide freedom of choice regarding cards, terminals and back-end systems. The GlobalPlatform card specification [26] provides a common security and card management architecture which provides a hardware-neutral and vendor-neutral smartcard infrastructure. The specification defines the behavior of a GlobalPlatform Card. The GlobalPlatform card architecture (see figure 4.3) is comprised of a number of logical and physical components that provide application interoperability and security, in an issuer controlled environment.

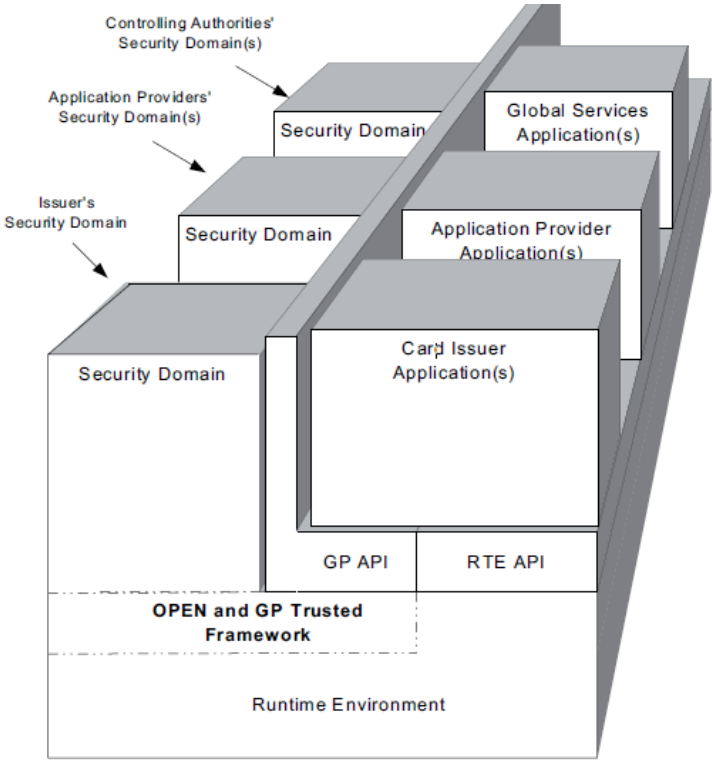

Figure 4.3 GlobalPlatform Card Architecture [26]

At the bottom of the architecture we find the Runtime Environment, which is responsible for providing a hardware-neutral API for applications as well as secure storage and execution space for applications. The runtime environment ensures that each application and its data can remain separate and secure from other applications on the card. The Trusted Framework provides inter-application communication services. The GlobalPlatform Enviroment (OPEN) provides an API to applications such as command dispatch, application selection and card content management. The three security domains reflect the three types of authorities recognized by a card: Issuer Security Domain, Application Provider Security Domains and Controlling Authority Security Domains. The Issuer Security Domain (ISD) is used to perform all issuer related card content management such as holding the issuer's keys and performing cryptographic operations when content changes occur. The Application Provider Security Domain (ASPD) is a secured environment which allows application providers to download, install and maintain applications. The Controlling Authority Security Domain (CASD) enforces the security policy of all application code loaded onto the card.

Standard smart cards are usually configured and loaded with applications prior to distribution to the consumers. To support post-distribution configuration the GlobalPlatform card specification introduces Delegated Management. Delegated management allows card issuers to delegate some card content management operations (such as loading, installation etc) to application providers. Delegated management protects card issuers from unauthorized changes and gives application providers the flexibility of managing their own applications. Delegated management works by using tokens and receipts, both utilize public key signatures and in generel terms enable the ISD to only allow authorized changes, and document any changes made to the cards content.

Utilizing a secure element supporting GlobalPlatform would allow a card issuer to easily cede some control of the secure element to an application provider, while still maintaining the integrity of the secure element. The way this is done is by using delegated management mechanism as described above. The mechanism allows a card issuer to "preauthorize" actions they wish an application provider to be able to accomplish, i.e. install an application, retrieving sensitive information etc. The Card issuer can identify the correct application and provide access to it, by examining and validating an applications unique identifier. The identifier also protects the applications integrity and authenticity thereby having a dual purpose. The identifier is usually protected by public key cryptography and is assigned to the application at compile time, also known as a code signature.

4.5 Secure Socket Layer/Transport Layer Security

For more information on SSL/TLS we refer to the main sources of this chapter [53, 54, 55]. SSL/TLS is one of the most widespread security protocols in use today it provides confidentiality, integrity and authenticity. It was created by the Internet Engineering Task Force (IETF) from 1997 to 1999 as an attempt to harmonize the Microsoft and Netscape approaches. Netscape had developed the Secure Socket Layer protocol (SSL) and Microsoft the Secure Transport Layer Protocol (STLP). TLS was built on the same underlying concepts as SSL, implementing support for more encryption standards. Because of TLS's role as SSL's successor and the fact that they are supported by most browsers they are generally referred to as one protocol SSL/TLS.

The SSL/TLS protocol is generally used for protecting data on the web transferred using HTTP. This combination of sending HTTP messages through a SSL/TLS connection is named HTTPS and is widely used to present secure content to a user through a web browser. In the protocol stack SSL/TLS is located between the Application layer and the Transport layer supporting both connection-oriented protocols such as TCP and datagram-oriented protocols such as.

SSL/TLS is used in many applications where security is essential especially within the bank sector and commercial websites where sensitive data is exchanged between these and the user. Though SSL/TLS offers a secure channel to communicate between server and client, pure HTTP is still being used because the support of SSL/TLS is costly performance wise.

The SSL/TLS protocol is divided into two phases, a handshake phase and a data transfer phase. The handshake phase has three objectives, protocol agreement, establish cryptographic keys and authenticate the participants. The client and the server first agree upon a protocol. In the initial stages of the communication the client sends the server a list of supported cipher suites where after, the server, chooses a cipher suite and responds with the server's certificate. This certificate then serves to authenticate the server as well as providing the client with the server's public key. The cryptographic keys are then calculated on both sides separately based on the chosen cipher suite and a Pre_Master_Key. This key is created by the client and delivered securely to the server by encrypting it with the server's public key. To ensure the reliability of the handshake MAC's are exchanged between server and client at the end. Optionally SSL/TLS allows authentication of the client. This can only be initiated by the server and is realized by the client responding to the request with his certificate. To ensure that the client is actually the owner of the transmitted certificate the client also transmits a Certification Verify message which is signed with the private key associated with the certificate.

The data transfer is enabled by the handshake phase which provides symmetric keys that are used to ensure integrity, confidentiality and authenticity. Data exchanged in the SSL/TLS protocol is fragmented in records which contain parts of the data, a MAC and a record header. The MAC is computed over the data and is then appended to the data. This is then encrypted before being appended to the record header, completing the record. The encryption of the data and the MAC combined resolves the three security objectives described in the CIA as mentioned in section 2.1. The record header is only there for practical reasons specifying the length, type and SSL/TLS version of the record.

SSL/TLS also provides an option for use of pre shared keys TLS-PSK which omits the use of certificates. Depending on the chosen cipher suite this eliminates the need for public key operations which is useful for systems with limited computational resources.

5 State of the art

This chapter will present an overview of the state of the art with regards to mobile authentication, identity and security. Each section will describe some of the most prevalent mechanisms used within these fields. Some of these mechanisms are currently being used, while some others are innovative and not yet used in practice.

Section 5.1 presents research related to authentication on mobile platforms. Section 5.2 will present an overview of applications within electronic identity. Section 5.3 presents considerations and solutions to security issues on mobile devices, more specifically smartphones.

5.1 Mobile Authentication

There are several different novel approaches for authentication on mobile platforms, this section will focus on authentication approaches which relate to the proposed solution in this document.

5.1.1 Certificate based authentication

Certificate based authentication relies on public key cryptography to generate certificates which in turn can be used to authenticate users, the advantage of public key cryptography for authentication is the asymmetric nature of the keys used. Asymmetric keys allow two parties which do not trust each other in advance to authenticate each other on the basis of a third party trusted by both parties called a CA, more information on certificates can be found in section 2.2. An Authentication scheme proposal based on certificates for a mobile environment is described in [18] by Mantoro and Milisic, their proposal is based on smart card technology and public key cryptography. The proposal is implemented by utilizing the tamper resistant memory provided by smart cards, which exists in mobile phones as a SIM card. The tamper resistant memory provides an excellent location for storage of the sensitive data required to perform public key cryptography. The usage scenario for the above mentioned approach is communication between an internet application (hosted on a PC) and the users mobile phone, the communication takes place via NFC, through an NFC reader connected to the PC. The phones SIM card transfers the public key to the application on the PC, which then proceeds to authenticate the user, by using a challenge-response authentication process. The private key never leaves the phone, and if the phone can provide the correct response it is authenticated.

The PKI suggested by Mantoro and Milisic requires the Mobile network operator (MNO) to be a trusted third party for communication between the application provider and the user. A PKI with the MNO in the role of CA, in which the MNO has to create and maintain all the users' certificates is ideal for the MNO's since they will be able to "commercialize" all applications which will need to use their PKI. Using an MNO's PKI also means the secure element will most likely be an UICC, this brings with it some big advantages. UICC allows secure remote management of sensitive data on a secure element. It essentially enables an MNO to remotely assign and provide certificates over the air (OTA) using GSM/UMTS. A unique aspect of the UICC is that it allows us to utilize more than one connection for sensitive data transfer. Instead of transferring keys over NFC it is possible to transfer the phone number via NFC and retrieve the certificates via the MNO. Splitting the delivery over multiple connections makes it more difficult to eavesdrop.

The most significant aspect of the research in the article mentioned above, for this document is the authentication mechanism. By basing the authentication mechanism on public keys combined with challenge response protocol enables the system to authenticate a user without directly involving a third party. As long as the certificates have been placed on the devices respective secure elements, the authentication will work “offline” without communication with the CA, for as long as the certificates are valid.

The proposed solution presented in the article mentioned above, does however have a major vulnerability in the fact that it only requires one authentication-factor. The only authentication-factor used is “something the user has” which is the smartphone. The smartphone provides the users credentials and is the only proof of identity for the user. This means if the smartphone gets stolen, there is nothing preventing a thief from impersonating the user.

5.1.2 Biometric based authentication

Biometric authentication takes advantage of the fact that every person has unique biometric data such as fingerprints, facial structure or iris patterns. These biometric samples are converted to a biometric template which is a digital reference of distinct characteristics extracted from a biometric sample, which can then be used during the authentication process. An implementation of a biometric authentication system for the mobile environment is presented in [21] which utilizes the unique fingerprint patterns to distinguish authorized users from intruders. The fingerprint images are captured and verified using a smartphone. This approach provides a two-factor authentication (as explained in section 2.3.1.) using “something you are” e.g. the biometric data and “something you have” which is the smartphone. The authentication begins when the user captures an image of his finger, which is then verified by the phone using a stored copy of the user’s biometric template. The fingerprint is compared to the biometric template and if the resulting score is higher than the system acceptance threshold, it is accepted. The user then touches an NFC reader with the phone, which allows the phone to send an authentication token via NFC to a PC, if the user is authentic; the user gets authorized and can gain access.

Biometric authentication is in theory more secure than traditional password based authentication, since it cannot be lost or forgotten and is not vulnerable to guessing attacks. Furthermore, biometric authentication has the potential to be more user-friendly than traditional authentication mechanisms, since it does not require a user to remember a pin code and is convenient as people naturally carry it around for other purposes. The usability and security of biometric authentication depends heavily on good error rate performance.

False acceptance rate is the chance of allowing an impostor through, it is paramount that this rate be very low for security reasons. False rejection rate is the chance that an authentic user will be rejected, and required to reenter the biometric data. For convenience and usability this rate must be low. For system comparison a value called Equal Error rate (EER) is used, which is the point where equal amount of false acceptance and false rejections occur.

The EER performance of the prototype presented by the authors varies greatly depending on the phone used and which environment it is tested in. The article presents results reaching 0% EER using a Nokia N95 phone, the phone however was attached to a fixed position and an ideal camera setup was used. Another test was run using another phone in a more “real world scenario” setting, with the phone not being

attached to a fixture but manually operated by an individual. The EER achieved in this scenario was about 8%.

Biometric authentication is not limited to fingerprints, there are several different possible sources of biometric data. Facial recognition is also a fairly common biometric data source, which has seen wide use on notebooks due to integrated web cameras.

The authors of [22] show how it is possible to implement a mobile authentication system based on facial recognition. The authors have implemented this system on a mobile personal device which is slightly larger than a mobile phone, but compared to new smartphone they have almost the same technical hardware specifications. The system relies on capturing images of the user's face and comparing them to biometric templates of the user. The result is a score which is matched with a system acceptance threshold, very similar to the fingerprint comparison mentioned earlier. If the score is higher than the threshold the image is accepted.

The authors float the possibility of storing the user's biometric template on a central database server due to limited computational resources on the mobile device. Storing the template data on a server however makes the template data vulnerable to attack, and opens up the possibility for eavesdropping when data transfer occurs. Compromise of biometric data is very serious, since biometric data cannot be renewed or reissued, it is permanent. The authors chose to store the templates locally to avoid these complications and make the system not depend on network access.

The authors achieved an EER of 2% with rather low hardware requirements. An important factor in achieving high EER is the enrollment phase, in which the user's biometric templates are registered. More specifically the authors point at even illumination during this phase is critical to obtain better EER.

It is clear from the articles presented above that biometric authentication has huge potential as an authentication-factor, and is seen as a desirable replacement for traditional password based authentication. However the measured error rate performance in both articles suggests that biometric authentication is not yet accurate enough for use in authentication. UK Banks set a target for biometrics of a fraud rate (false acceptance) of 1% and an insult rate (false rejection) of 0.01% [7 ch. 13], it is clear that biometric authentication cannot currently reach this margin on mobile platforms.

It is important to note that biometric authentication is not a complete solution and still requires the support of an infrastructure similar to that which was described in approach one, e.g. the smartphone needs to utilize some form of secure storage to store the biometric data and encryption keys to pass the data around securely. However the results in the article point towards a very promising future for biometric authentication on mobile platforms with ever increasing camera resolutions and better camera software. It is not hard to imagine that biometric authentication will reach an acceptable error rate sometime in the near future.

5.1.3 One-time password based authentication

One-time passwords (OTP) have typically been deployed on dedicated hardware, as hardware tokens to provide additional layers of security to existing static password based authentication. The recent developments of one-time password authentication however has increased it's usability in mobile environments.

A one-time password authentication scheme for mobile environment is presented in [24] which proposes a two-factor mobile authentication solution, based on one-time password (OTP) as described in section 2.3.1. The authors propose an improved one-time password approach by using two different one-way hash functions, one for updating the seed and one for OTP generation. The scheme consists of two phases, registration phase and authentication phase. During the registration phase a device is provided with two hash functions and an initial seed which is unique for each mobile device. The seed consists of an International Mobile Equipment Identity (IMEI), International Mobile Subscriber Identity (IMSI) and registration date. The three values combined provide the initial seed which will be used as input to the hash functions. During the authentication phase the one-time password is used as an additional layer of security after the authentication of the static user credentials. The client provides a server with its current OTP status, to assure the client and server have the same seed value. The server then challenges the client with a set of two random indexes for the hash functions. The client and server calculate a new OTP using the indexes, if both are identical the client is authenticated.

The novel aspect of the approach presented in the article is the fact that by utilizing two nested hash functions it is able to provide forward and infinite OTP generation. Using forward hashing techniques allows for infinite OTP generation which increases the usability, unlike reverse or backwards hashing, which relies on generating a fixed amount of OTP's which are then used up. The implementation requires several computational steps by using two hash functions, but is still computationally more efficient than applying public key cryptography to produce a signature hash chain. The approach does not use a time-based algorithm and therefore does not have to guarantee a main server synchronized internal clock.

These characteristics make it very suitable for security applications on mobile devices. However it is important to note that even though this solution does not use a time-based algorithm it still requires connectivity with a main server during authentication. This means it is not possible to do offline authentication using the proposed solution. The solution which is proposed in section 6 of this document provides the means for offline authentication, which is a major advantage in mobile environments where online connectivity cannot be guaranteed.

5.1.4 Basic authentication

There are a lot of possibilities regarding authentication, however the most widely used is basic authentication in the form of a simple username and password, as such being able to support such an authentication system can be critical. The authors of article [20] have devised an authentication scheme based on public key cryptography and challenge response authentication which can support basic authentication. The article focuses on improving the security of standard password authentication without replacing the legacy password authentication systems. The authors achieve this by creating a challenge-response (section 2.3.1) scheme where the user's password is a part of the response code. The authentication server can recover the users' password from the response code and authenticate the user. Merging a password into the response code, keeps the password protected from most attacks such as eavesdropping and replay. This approach significantly improves the security of traditional authentication systems, and enables these systems to repel most modern attacks. Interfacing to traditional authentication systems is mostly relevant when discussing access control, the reason for this is that access control infrastructure is a massive investment, and is expensive to replace. Access control systems have a long lifespan, thus the ability to interface to these systems is significantly less expensive.

The most important aspect of this research in relation to the proposed solution in this document is an authentication scheme capable of integrating traditional password authentication systems with newer more complex authentication systems based on public key cryptography.

5.2 Electronic Identity Applications

The amount of services provided electronically such as e-government, e-tickets and e-payment is ever increasing and as such the need for an electronic form of identification is ever more present.

This section will present an overview of some relevant technologies regarding electronic identity and what their possibilities and limitations are.

5.2.1 Electronic Identity Card

The electronic identity card (eID) is used in many nations such as Germany, Spain and Turkey. Turkey has recently launched a pilot project to unify all electronic ID cards [30]. The electronic identity card is a physical card with smart card capabilities, used to authenticate users electronically. The majority of eID cards have a contact based interface (ISO 7816-3) similar to secure elements mentioned in section 4.3, however some utilize a contactless interface (ISO 14443). The smart card contains the sensitive information such as user identity information, biometric data and digital authentication certificates. The user identity is protected using PKI, and biometric templates for authentication, both of which were presented in detail in sections 5.1.1 and 5.1.2 of this document. The biometric templates used in the Turkish national eID are fingerprints and finger-veins. To protect the privacy of the eID holder the biometric templates are not released until mutual authentication is performed. For authentication the eID card can be used with or without biometric data depending on the authentication policy of the given service, however for high level security all three authentication factors are used, biometric template, pin-code and the card itself. The card is compatible with most smart card readers, however it requires a specialized device known as a Card Access Device (CAD) to use the biometric authentication. The three levels of authentication achieves very high levels of security and is difficult to compromise, since the authentication process itself is done online with an authentication server. The confidentiality, integrity and authenticity of the communication between the CAD and the authentication server is achieved using public key cryptography.

The eID has great potential and is definitely very secure which is naturally a requirement for a device which is supposed to be used with e-Government applications. However the high levels of security is not necessary for all types of applications and might appear extreme for smaller applications such as student ID or library cards. For smaller applications the biometric authentication and the special equipment required could significantly reduce usability and increase the cost of the required infrastructure. The fact that most eID use a contact based interface is from a security perspective a safer alternative to contactless interface, as it is far more difficult to eavesdrop a contact interface as it requires physical tampering with the card readers.

The bad aspect of a contact based interface is that it hampers the possibility of peer-to-peer communication, i.e. that users can authenticate each other without a terminal connected to an authentication server. It is conceivable that the eID can at some point be extended to work on a smartphone. However, if the eID is contact-based it makes that transition significantly more unlikely, because the smartphones will not be compatible with the CAD or other terminals used with contact-based eID.

5.2.2 Contactless Cards

Identification is the hallmark of RFID and gave birth to the term "Internet of things", the idea that any object can be equipped with tags and be identified electronically. RFID has since seen widespread deployment in institutions and workspace in the form of RFID enabled ID cards for identification and access control purposes. Aarhus University has such an access control system based on RFID cards and RFID terminals[31]. The cards used are of the type MiFare classic and each student is issued a card, with an image of the student and the social security number for identification purposes. For access control the contactless integrated circuit card (ISO 14443) is used, combined with a pin code to provide two-factor authentication. The cards also contain a magnetic stripe to support the legacy access control systems in some of the universities older buildings. The RFID chip in the cards are not used for identification purposes other than in tandem with access control. For identification in other scenarios the image or the social security number on the ID card is used. The social security number is used to link a students identity to a citizen database for use with other government services such as libraries and to validate enrollment confirmations.

RFID cards can also be used for payments, in areas where small amounts of money are involved such as cantines or printing services, which is ideal in a school environment, however the levels of security regarding RFID payments is not high enough to support large amounts of money(in general). The RFID payments works by mapping the serialnumber on the RFID card with a consumers account, which gets debited each time a payment with the card occurs. The payments require no authentication other than the card itself, which is why it is not safe enough for larger amounts of money. Some applications however increase the security by incorporating a challenge-response protocol [39].

Note - Some NFC phones can be used with RFID systems if they support card emulation and contain a secure element. It is important to remember that card emulation, will emulate the type of cards used. As such the level of security using card emulation is limited by the type of card emulated. The integrated chip on the student card of Aarhus University is a Mifare Classic 1k RFID chip, which offers one kilo byte of storage and a serialnumber as mentioned earlier. These data are protected by a proprietary security protocol which long since has been hacked to the point where as W. H. Tan states in [51], *"MIFARE Classic cards can now be cracked and cloned in a matter of seconds discreetly."*. This renders the security offered by the mifare classic chip minimal because it enables an attacker to clone a student card without physical access to the card. Taking into account the assets relying on the security of the chip, the risk is not critical.

5.2.3 Near Field Communication

NFC (described in section 4.2) as a contactless communication technology has huge potential to revolutionize the way users interact with computers, terminals and each other. NFC allows users to accomplish tasks with one device, which would otherwise require specific cards or hardware, such as identification, access control and transactions.

The success of physical cards for payments, identification and other purposes have led to an increase in the amount of cards an individual has to carry around. This has pushed the development of NFC as a technology capable of performing the function of multiple physical cards with one device. The reason NFC is capable of this is the fact that it utilizes a secure element (SE). Secure elements which follow common standards such as GlobalPlatform have the option of allowing multiple application providers to utilize the excess space

available on the SE as described in section 4.4.1. The SE allows an NFC enabled device to become a platform holding multiple identities and payment accounts, while simultaneously capable of allowing access to private areas, such as home or work. There are already NFC payment applications such as Google Wallet, which works with most major credit cards and can be used any place that supports contactless payments. Google wallet is a great example of how NFC technology can incorporate the function of several cards into one platform[38]. It is however important to note that the Google Wallet application only works in the United States and only with specific mobile network operators (MNO) who have a partnership with google. Google Wallet's dependence on specific MNO's suggests that it utilizes the SIM card as a secure element and relies on GSM features for authentication. In other fields such as access control and identification several companies and governments are currently working on innovative applications using NFC. A few examples of successful demonstrations and trials of NFC based applications are the NFC access control demonstration in a Swedish hotel[36] and the NFC based National ID pilot project in the United Arab Emirates [37]. The demonstrations and pilot projects illustrates that there definitely is a market for such services.

Demonstrations and pilot projects aside there is a significant amount of research in the area of NFC which offer several proposals for implementations of NFC based applications.

The authors of [40] propose an NFC mobile payment application based on the authentication and identification capabilities of the SIM card in mobile phones, which is very similar to Google Wallet, the application is not limited to payments it can also be used for identification. The proposal suggests using the existing GSM features for authentication and identification with SIM cards, the proposed solution can take advantage of MNO's existing services thereby reducing cost of development, lower the initial investment cost and simplify integration into the current mobile network infrastructure. However the MNO's will look to profit from the use of their network and services and will require payment from the application providers. The GSM security is based on an Authentication Centre which uses symmetric key cryptography and it's knowledge of International Subscriber Mobile Identities to perform a challenge-response protocol with the SIM Card for authentication. The authentication capabilities of GSM is used to authenticate all three parties in a transaction, the customer, the shop and the backend-system, before any transaction is initiated. The triple authentication means that it is a requirement that shop owners who wish to participate in a system such as this are registered with the corresponding MNO and must have a contactless point of sale system which is connected to the MNO. Furthermore, using the GSM authentication means that all parties engaging in authentication must establish a relationship to the same MNO. Regarding the level of security, GSM security provides cipher keys of a maximum length of 64 bits. 64 bits is a relatively small key length which do not provide enough security for larger transactions. Considering the limitations of the proposal, it still presents opportunities to benefit from the current legacy systems with NFC for small payments and identification purposes.

There are other similar proposals based on SIM cards which utilize more modern networks such as 3G for payment and identification purposes. The authors of [29] present a proposal of such a system, which is based on 3G instead of GSM, and uses a PKI infrastructure supported by the government. The PKI used is known as Citizen Digital Certificate (CDC) which is a physical smart card which contains a government issued certificate. The scheme is divided in two phases, a registration phase which is only done once and a transaction phase. The CDC is used during the registration of a users NFC enabled mobile phone and

provides trust between the user and the MNO by having the government act as a trusted party in the role of CA. The CDC is used to generate endorsed credentials for the secure element on the mobile phone which the MNO can trust since they are signed by the private key located on the CDC and by extension legally recognised by the government. Once the credentials are created the user can engage in transaction services offered by MNO's which recognise the government CDC. 3G networks introduce maximum key lengths which are twice as long as GSM (2x64 bit), which is a significant advantage for an application in which security is paramount.

The core of the proposal is the endorsed credentials which is intriguing since it removes the MNO from a position of maintaining the identity of its users. By removing the MNO from this position it is possible to support multiple MNO services and it is not necessary for shops to be equipped with proprietary MNO point of sale systems, which was one of the limitations of the previous proposal. Using PKI however is less computationally efficient compared to symmetric key cryptography, which is a concern in a mobile environment, but it does remove the difficulties of key distribution faced by symmetric keys.

A solution similar to the two proposals discussed above, is presented in [28], the major difference is the fact that instead of utilizing the SIM card as a secure element it takes advantage of external smartcards as the secure element for transaction purposes. The communication protocol is based on asymmetric, mutual authentication using certificates which is very similar to the proposed solution this document will present in chapter 6. The external smartcard allows the solution to be entirely independent of MNOs, which makes the proposal easier to deploy. It is worth mentioning that it is capable of using contactless smartcards and performing secure peer-to-peer transactions between two smartcards.

5.3 Mobile device security

Several state of the art techniques to provide security through authentication have been presented in section 5.1.1. However for mobile devices there are further concerns besides devising a secure protocol. Storage of cryptographic keys and other sensitive data is a major concern, as no matter how secure a protocol is, if the keys and other data are not accessible or securely stored the application has a major vulnerability.

5.3.1 Mobile platform

In a mobile operating system such as Android, each running application is assigned a portion of the internal memory which only the application can access. This gives application providers execution space to perform cryptographic operations, in a safe and secure manner. The security provided by the OS cannot be guaranteed by the OS developers nor the smartphone manufacturers as users are capable of compromising the OS platform by intentionally modifying it, also known as jailbreaking or rooting depending on the platform[23]. Users can have multiple incentives to intentionally modify the platform on their smartphone since it disables multiple security features in the OS, not just the security of the memory provided to applications. The most common reason users intentionally modify their platform is to gain access to third party applications or applications otherwise inaccessible on the original platform. However there is a risk that these applications can contain malware and with an intentionally modified platform they can wreak havoc on the smartphone. There are variety of possible threats malware poses on the smartphone such as availability attacks which executes random instructions just to slow down the smartphone or by stealing and

possibly damaging private information located on the smartphone. The malware threat is not limited to the device which is infected, most malware continues to spread to other devices using the many communication options available on smartphones such as Bluetooth and Wi-Fi[42]. The bottom line is that at the time of writing this document, storage on a smartphone is secure enough for most purposes as long as the platform is not modified, however since this can't be guaranteed applications providers will have to look elsewhere if the information they require to be store is critical.

The challenges described above has led to research into possible alternatives for secure storage for applications in mobile environments leading to storage options in which the applications providers have more control.

5.3.2 Remote key storage

When a local environment is untrusted, it is natural to look for secure storage and secure execution space outside the mobile phones environment. The most widely used form of secure storage in mobile authentication is a secure element which in most systems is the SIM card, several examples of payment and identification systems using a secure element have been presented in section 5.2.3. However there are alternatives to secure elements and ways to enhance the security a secure element provides. Remote servers have been used for years to facilitate secure communications and store vital information. The remote storage offers a separation of hardware from the mobile device and separation of responsibility from the mobile device user. A solution based on remote key storage using servers is presented in [43]. The authors present a key management mechanism for NFC which is capable of facilitating a mobile payment application. The mechanism establishes a secure connection with a remote server by using a secure element within an NFC handset. The secure element is used to store and generate cryptographic public keys for communication with a server, once a secure connection has been established using these temporary keys, the server validates the secure element, and responds with the key stored on the server, the temporary keys can then be used and discarded. The next time the key stored on the server is needed the process is repeated, this way no sensitive data is stored long-term on the handset. The key in this case is used to access a contactless Mifare card, to authorize a payment. The mechanism however can be used with almost any other type of security key, to support many other applications. It is also possible to take the mechanism a step further and instead of storing keys, it could be possible to store identity credentials. This way if the handset is stolen, the user can report the theft to the server administrators and remove the handsets access to the servers, where otherwise the thief would have ample time to attempt to tamper with the secure element. More importantly the application is more resistant to active attacks as the data have to be obtained real-time through the server. The communication with the server is protected by challenge-response authentication process. However, the increased security comes at the price of server maintenance for the application, and exposes the sensitive data to an additional channel i.e. between the server and handset.

It is conceivable that the server functionality described above, could be achieved through the use of cloud computing, which is capable of expanding its computing capabilities with user demand and shrink when the demand is low. Furthermore, cloud computing removes the large initial investment which would otherwise be necessary to deploy dedicated servers, thereby making is more economically viable to use remote storage for the purpose described above. However, examining the security repercussions of using cloud

computing to store sensitive data is out of scope for this document and more information on this subject can be found here [44].

6 MS-ID Authentication with NFC-enabled mobile devices

This chapter presents the proposed authentication solution using NFC enabled mobile devices. Section 6.1 presents the goals of the MS-ID. Section 6.2 describes the proposed design and presents the three primary protocols RMS, AMS, AMA. Section 6.3 presents the parts of the proposed solution which are implemented in the prototype. Section 6.4 assesses the extent to which this implementation satisfies the goals previously presented.

6.1 Goals

The goals listed in section 3.3 Mobile student ID Goals:

- Server authentication, Student authentication, Data authentication
- Privacy
- Usability
- Cost

These goals should be met at least to the same degree as with the solutions described in Section State of the Art: Authentication.

Additionally the resulting prototype must be implemented for smartphones and as such must aim to utilize OS independent libraries to increase portability. Finally, it is desirable for students to be able to download and install the application onto their smartphone themselves. The same applies for authenticators.The prototype must have a high level of usability, and as such must not depend on network coverage.

6.2 Design and Architecture

The diagram below presents an overview of the system with the principals grouped in three domains Student, Third party and School Authority. The principals are grouped according to the authority responsible for them. To keep the diagram simple the principals represented by channels are not included.

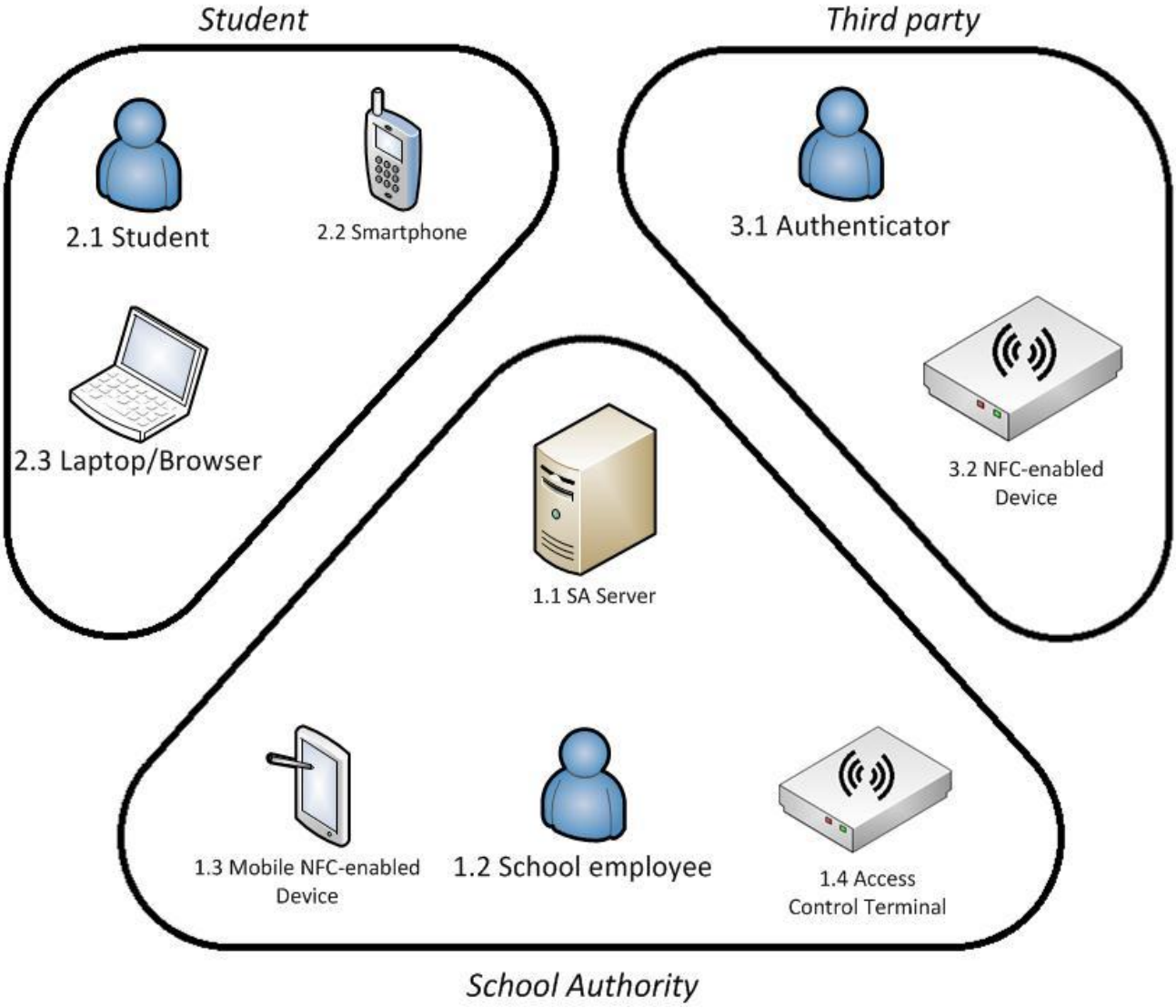

Figure 6.1: Domain overview.

6.2.1 MS-ID Structure

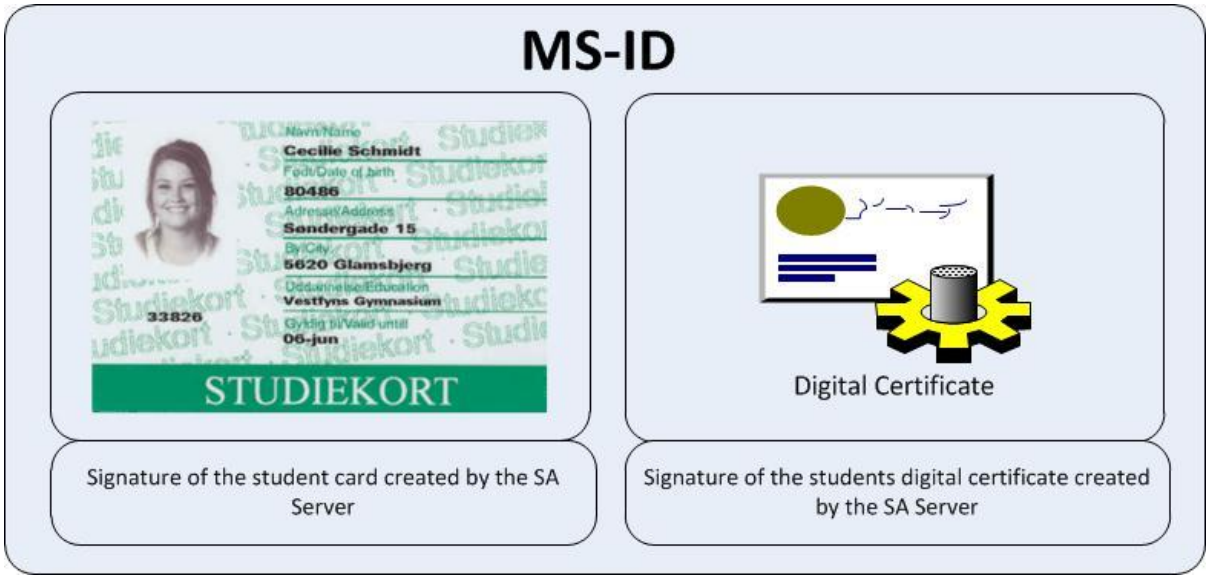

Figure 6.2 Overview of the MS-ID structure

The MS-ID provides the student credentials for authentication purposes, it consists of two artifacts: A Certificate and a signed ID Image. Both artifacts are protected by a digital signature which ensures the integrity and authenticity of the artifacts as described in section 2.2.

The certificate provides details of the Students University, student ID number and enrollment validity time. It is signed by the SA server using its own SA certificate. The MS-ID certificate enables the MS-ID application to authenticate the student without revealing sensitive data which can be exploited by the verifier. This is done through a challenge response protocol which verifies that the MS-ID application is the owner of the MS-ID certificate.

The ID image is an image which contains a photo of the student along with similar information as the certificate. It is also signed by the SA server using the SA certificate. The ID image is transferred to the verifier when used in authentication and it can therefore not be ensured that the ID image remains confidential.

6.2.2 Principals

This section describes the principals shown in figure 6.1 and their relations.

6.2.2.1 Student Domain

The student is a subject who is a legitimate student at a university who is utilizing MS-ID. The purpose of the student is to obtain access to school resources and facilities as well as external benefits such as rebates from third parties. It is assumed that the student is equipped with a personal smart phone. It is also assumed that the student has authorized access to the school education management website.

The smartphone is the property of the student, it is the platform of operation for the MS-ID. The smartphone must contain a bluetooth interface and either a camera or have NFC capabilities. It runs the MS-ID application which is capable of interacting with other NFC enabled devices. The MS-ID is stored on

the phone allowing for offline authentication. For the MS-ID to be on the smart phone it must first registered by the SA Server through the internet.

A Laptop/Browser provides a secure connection to the existing school website, which offers services to the student. The transfer of the MS-ID is initialized by the student when logged into his account on the school server. Requirements for the Laptop/Browser are that it is able to connect to the school website and support the security mechanisms that the website provides.

Authenticator NFC device

The third parties NFC device can be any NFC enabled device such as a mobile phone or a terminal, as long as it provides a display for verification for the MS-ID. It is also assumed that this device has at some point been connected to the internet, and installed the MS-ID verifier app. The MS-ID verifier app contains the SA server's certificate which will be used to verify the MS-ID provided by the student, this provides verification of the first authentication factor.

6.2.2.2 School Authority Domain

School authority server is also referred to as the SA server its main obligations are to generate and distribute the MS-ID and to manage access control. It corresponds to the computing equipment at the school side that services the browsers' requests using SSL/TLS as described in section 4.5. The system needs to provide a web interface by which the students can login to acquire the MS-ID app and perform the registration process used to retrieve an MS-ID. Secondly the system must be able to create and maintain certificates as a certificate authority(CA) i.e. provide public certificates upon requests. A strict requirement following the role of certificate authority is that the SA must be able to maintain the confidentiality of the private key used to sign certificates. The last requirement on the SA system is managing access control terminals, to only allow legitimate students into the school facilities. It is assumed to be very unlikely to be compromised. The mechanisms devised to ensure this condition fall outside the scope of this document.

School Employee is as the name states a person who is employed by the school. At exams the school employee may authenticate students through the Mobile NFC enabled device described in the paragraph below.

The Mobile NFC enabled device is used by the school employees to authenticate students at exams. For the device to provide a list over students participating in the exam it must first have been asked to download the appointed participants list from the school servers. The device must provide a mechanism for school employees to authenticate themselves.

The Access Control Terminals facilitate access to school facilities by allowing for authentication at the entry to the restricted facilities. The main requirements to the access control terminals are therefore that the terminals must be able to authenticate students with MS-ID. This then requires that the terminals are equipped with NFC interfaces. The terminals must also present a keypad, allowing for input of a pin-code used in the authentication. To gain knowledge regarding authorization the terminal must be able to connect to the schools access control servers.

6.2.2.3 Third party Domain

The Authenticator is a clerk at a shop offering student discounts and therefore has the incentive to be able to authenticate student. The clerk facilitates the interests of the shop owner to only give discounts to students. This is achieved by the clerk who authenticates students through the NFC-enabled device and validation of a picture of the student. The only requirements to the clerk are that she must be trained to operate the NFC-enabled device and identify people from a picture.

The NFC-enabled device is an electronic device with an NFC interface, it offers the capability of validating the MS-ID authenticity and to present it. The device can be represented by a smart phone, another handheld device or a stationary device. It differentiates from the Mobile NFC enabled device of the School Authority by not necessarily offering mobility. At a time prior to authentication the device must have had the MS-ID Verifier application installed. The device must offer display capabilities allowing for a picture of the MS-ID owner to be presented.

6.2.3 Relations

This section describes the interfaces between the principals mentioned in the previous section. The section is divided into the four scenarios Registration, Identification, Access Control and Identification during transaction mentioned in section 3.1.

6.2.3.1 Registration

The registration scenario includes all the principals in the Student domain and the SA Server principal. The initiator of this scenario is the student who wishes to utilize the MS-ID and therefore engages to acquire it. This scenario is a prerequisite for the other scenarios describing the features MS-ID.

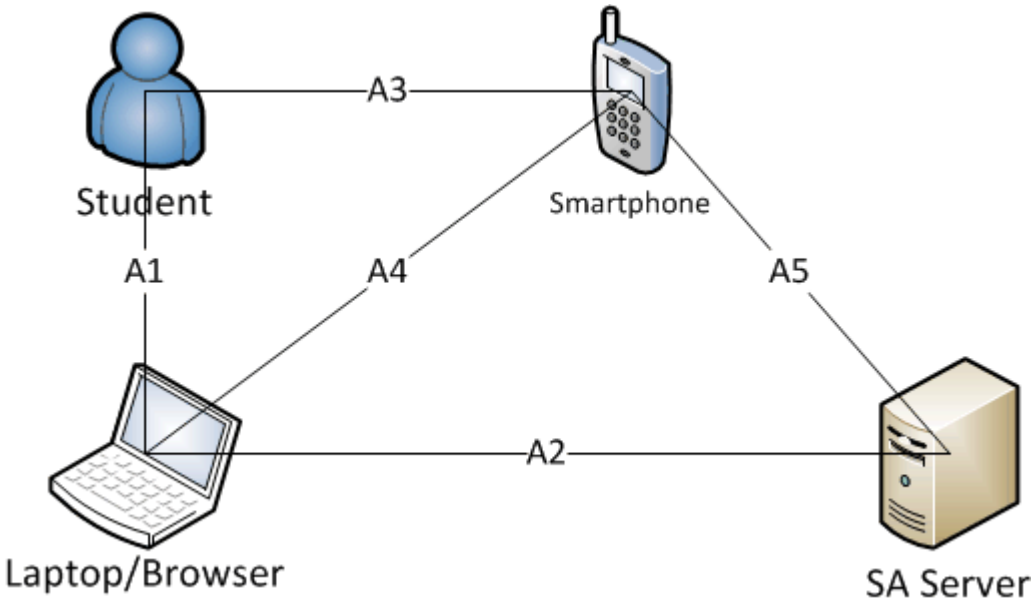

Figure 6.3 Principals in the Registration scenario.

A1: The communication between the student and a Laptop/Browser is initiated by the student to gain access to the school website. The screen on the laptop serves to display the school website and the input mechanisms i.e. mouse and keyboard provide capabilities to navigate and send information to the website.

A2: The communication between the Laptop/Browser and the SA Server is initiated by the Laptop/Browser and information is passed over the Internet using the HTTPS protocol. The SA Server interface is used to authenticate the student through his credentials i.e. student number and password. Information about the installation and use of the MS-ID are sent using the Laptop/Browser interface. The SA server also sends an authentication token used to authenticate the smartphone in connection A5.

A3: The communication between the Student and the smartphone is initiated by the user. Through this connection the student can control his smartphone and information and instructions can be given both ways.

A4: The communication between the Laptop/Browser and the Smartphone is initiated by the Laptop/Browser. It is used to ease the workload on the student by letting data be transferred directly to the smartphone. The data transferred is the security token mentioned in paragraph A2. This information can also be transferred by the student through A3. The A4 connection relies on 2D barcodes displayed on the Laptop/Browser screen and on the camera of the smartphone to transfer data as described in 4.1.2.

A5: The communication between the Smartphone and the SA server is initiated by the Smartphone and used to transfer the MS-ID to the Smartphone. The connection uses the TLS protocol to ensure security, where the smartphone uses the security token received through A4 to authenticate. The SA server is authenticated through the SA Server certificate.

6.2.3.2 Identification

The Identification scenario takes place at an exam and involves the principals Student, Smartphone, Mobile NFC-enabled Device, School Employee and SA Server. The scenarios goal is to authenticate the student to the school employee monitoring the exam.

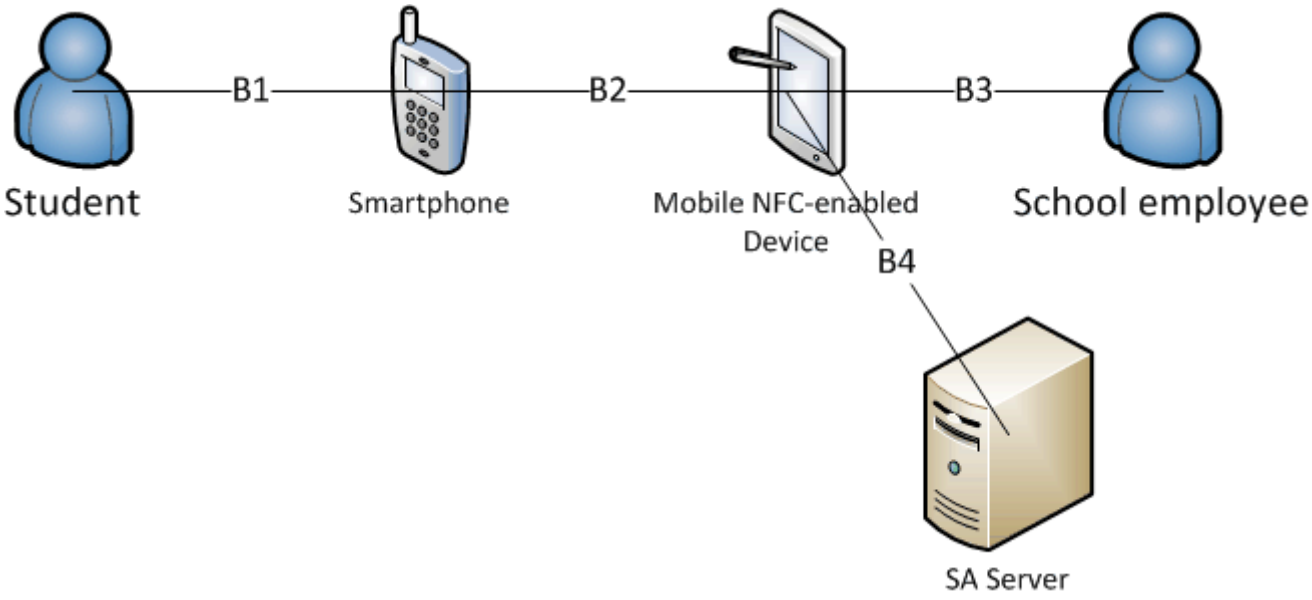

Figure 6.4 Principals in the Identification scenario

B1: The relation between the Student and the Smartphone serves only the purpose of bringing the smartphone into range of the Mobile NFC-enabled Device so the authentication can take place.

B2: The communication between the Smartphone and the Mobile NFC-enabled Device relies on NFC. Through it runs the AMS protocol used to validate the students identity, this protocol will be presented later in the document.

B3: The communication between the Mobile NFC-enabled Device and the School employee offers the school employee visual overview of the participants and status of the attendance to the exam. It also offers feedback from the authentication process allowing the employee to verify the student identity by picture. To verify the identity of the school employee the connection also facilitates transfer of credentials and password through a keyboard.

B4: The communication between the Mobile NFC-enabled Device and the SA server is initiated by the Mobile NFC-enabled Device. The connection is used to transfer lists of exam participants from the SA server to the Mobile NFC-enabled device. The connection uses the TLS protocol.

6.2.3.3 Access control

The Access control scenario is engaged by the student who wants to gain access to school facilities which are prohibited for unauthorized persons. The participating principals are Student, Smartphone and Access Control Terminal.

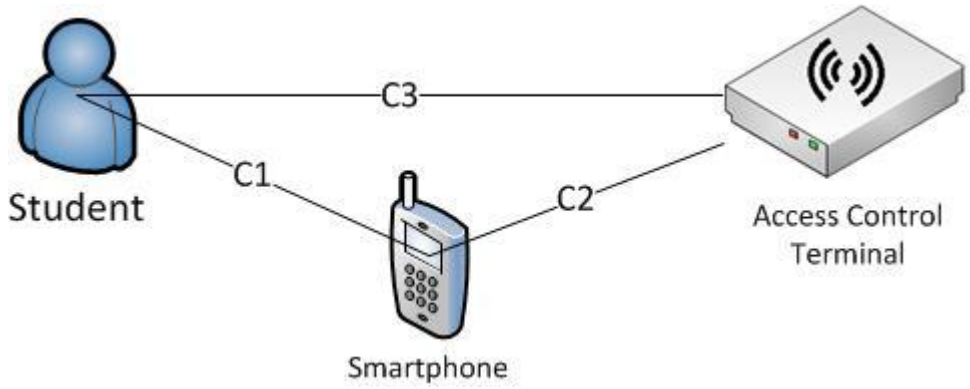

Figure 6.5 Principals in the Access control scenario

C1: The relation between the Student and the Smartphone serves the purpose of bringing the smartphone into range of the Access Control Terminal.

C2: The communication between the Smartphone and the Access Control Terminal is initiated by the Access Control Terminal which sends the Smartphone a message starting the MS-ID application. The connection is then used to authenticate the two parties.

C3: The communication between the Student and the Access Control Terminal is used to prove that the person who wants to gain access has possession of the second authentication factor i.e. the pin code. This is verified by the person entering it on the available keypad on the Access Control Terminal.

6.2.3.4 Identification during transaction

The Identification during transaction scenario takes place at a place of business offering discounts to students. The goal of the scenario is for the clerk(Authenticator) to authenticate and confirm the students identity as a legitimate student. The principals involved in this scenario are the Student, Smartphone, Authenticator and NFC-enabled device.

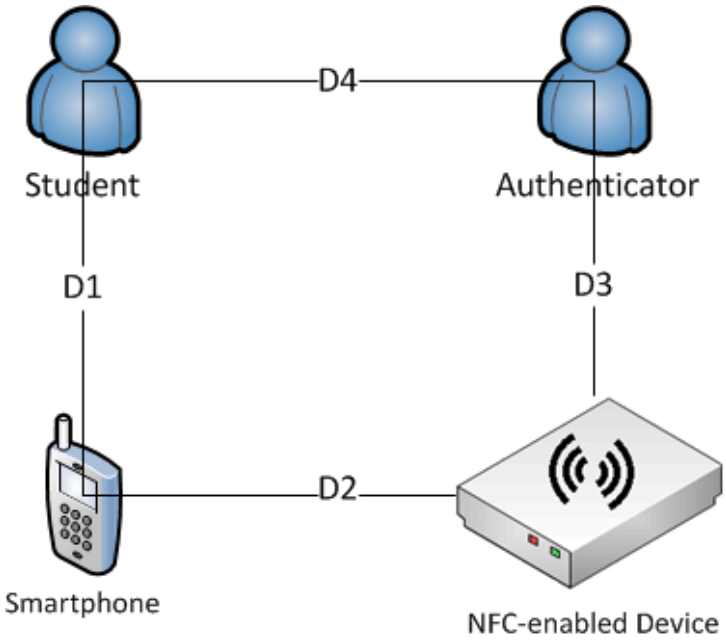

Figure 6.6 Principals in the identification during transaction scenario

D1: The relation between the Student and the Smartphone is initiated by the Student which brings the smartphone into range of the NFC-enabled Device.

D2: The communication between the Smartphone and the NFC-enabled Device is used to authenticate the students identity through NFC. This is done by the smartphone proving possession of the MS-ID i.e. the ID image and the certificate. The ID image is transmitted to the NFC-enabled Device.

D3: The relation between the Authenticator and the NFC-enabled Device is used to display the students ID image received through D2.

D4: The relation between the Authenticator and the student must offer the authentication the possibility of comparing the students facial characteristics with the picture displayed on the NFC-enabled Device.

6.2.4 Protocols

This section will clarify the inner workings of the protocols which support the scenarios described in section 3.1 scope. The protocol messages are displayed using sequence diagrams and a detailed explanation of each step of the protocol accompanies the diagram. A rationale section is included for each protocol, which enlightens the reader of the basis of which decisions are taken regarding the development of the protocols.

6.2.4.1 Registration of MS-ID (RMS)

Registration of MS-ID is the process of transferring, the mobile student identification (MS-ID) to a mobile device. It involves communication with the school authority (SA) server, with the end goal being a signed MS-ID transferred to the mobile device.

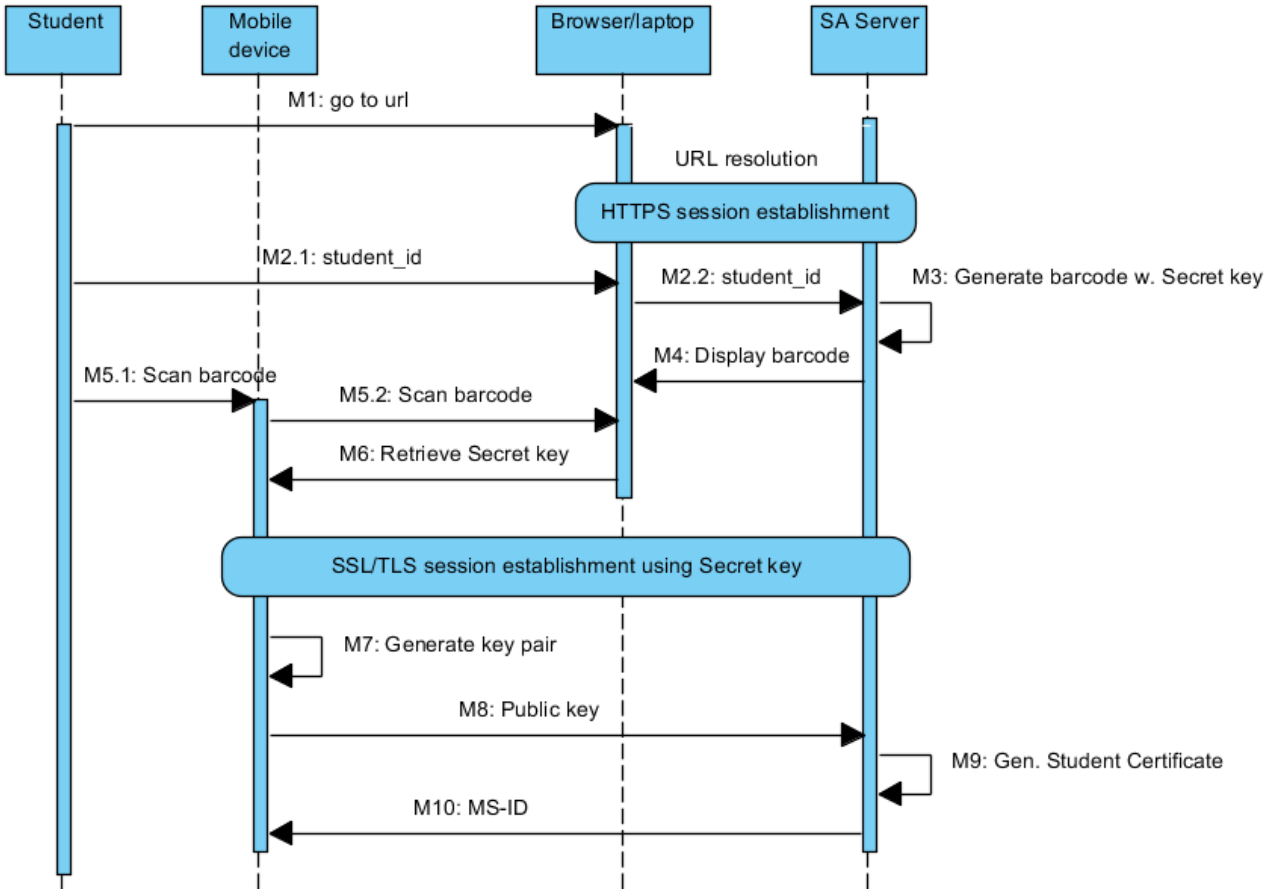

Figure 6.7 Student registration protocol

1 The student enters the URL of the SA site in the laptop (M1).
2 The laptop resolves the URL and opens up the SA's site. A HTTPS session is established between the SA and the browser.
3 The server sends a form to the browser for login.
4 The student types in her student-ID in the laptop, i.e. her student number and password (M2).
5 The server generates a secret key and embeds it into a 2D barcode (M3).
6 The 2D barcode is then displayed on the student's laptop (M4).
7 The student can now scan the 2D barcode using the mobile device, and retrieve the secret key (M5, M6).

8 The mobile device uses the secret key to establish an SSL/TLS connection to the SA server.
9 The mobile device generates a key pair (public key and private key) (M7).
10 The mobile device transfers the public key via the secure SSL/TLS connection to the server (M8).
11 The server generates a new certificate based on the public key received (M8) and previously known student details. The server signs this certificate using its own certificate (SA certificate) to allow authenticators to verify the authenticity. Furthermore, the SA server generates an ID image using an image of the student and data mapping the ID image to the students certificate, which is then signed using the SA certificate. The certificate along with the ID image is the "MS-ID" (M9).
12 The SA server responds with the MS-ID and terminates the SSL/TLS Session(M10).
13 The HTTPS Session is terminated upon logging off the SA website.

6.2.4.1.1 Rationale for Registration

In this section the properties of the RMS protocol will be presented along with alternative design opportunities.

MS-ID Credentials; The registration scenario utilizes the secure connection already existing between the student and the school i.e. the school website. This connection is secured by the TLS protocol explained in section 4.4. The provider of the certificate used to verify the schools identity is the GlobalSign Organization Validation CA - G2. The student is authenticated during the login process on the school website, using student number and password which authorizes him to manage his relation to the school over a TLS connection. This demonstrates the importance of student number and password as credentials and is why the system is not designed to authenticate the smartphone directly using the students credentials. Allowing the student to enter his credentials into the MS-ID application could allow for malware infecting the students smartphone to intercept these. Further phishing applications masquerading as an application originating from the school could, easily resemble the original application and lure the students to hand over their credentials. Both cases can enable the attacker to compromise much more than just the MS-ID. Instead the system generates an additional set of credentials which only authorizes the MS-ID. The original set of credentials is thus separated from the effects of a compromise of the MS-ID.

Transfer Credentials; The user scans the 2D barcode from the school website to enable a secure connection between the smartphone and the SA server. Such a connection could also be achieved by displaying a human readable password which could be entered into the smartphone by the user. However, barcode scanning is considered more convenient and it is assumed that most smartphones support this technology. The barcode does have some disadvantages such as, being easier and faster to read than a human-readable password and as such can be eavesdropped faster by a malicious person seeing the barcode displayed on the students PC. The system is designed to minimize the risk of someone exploiting the information in the barcode to receive the students MS-ID by making it usable only one time. Therefore an attacker would have to scan the barcode before the student to be successful.

SA Server Connection; The connection between the students smartphone and the SA server is of the type TLS PSK described in section 4.4. This is utilized because such a connection offers authentication of both parties on the basis of the shared knowledge of the PSK. TLS also offers mutual authentication through certificates, these are however more difficult to exchange through human interfaces or barcodes. The initial communication between the SA server and the Smartphone represented by A2 and A4 in section 6.2.3.1 has

the limitation of only being one way and allowing for a finite amount of data to be transferred. More accurately, the data transferred must be able to fit in the 2D barcode displayed on the school website. It would be possible to partition the data into several 2D barcodes but this would impact usability negatively because user would have to scan more than one barcode. The amount of data needed to be transferred to the smartphone to authenticate in TLS would include both a signed certificate and the corresponding private key, bringing the barcode size up to a point where it might be difficult to scan. The data transferred if using TLS PSK only includes the students PSK identity and the pre shared key which is 182 bytes in total [53].

MS-ID; The MS-ID consists of two artifacts, a certificate and an ID image, which are used in different scenarios. The certificate serves to affirm the identity of the student to a verifier by proving the possession of the private key corresponding to the certificate. This makes the smartphone which has the MS-ID installed an authentication factor of the kind “something you have” though the knowledge of the private key is not physical. Generating and storing the private key in a secure element however, makes it a physical token. This is because the keys stored in the secure element are not accessible from the general execution environment. The secure element upholds higher security than most physical security tokens because of its defenses against tampering and cloning. The ID image is used to ensure authenticity and integrity of the image of the student. In the Identification and Identification during transaction scenarios the ID image would be enough to authenticate the student securely. This is because it ensures that the validity date, picture of the student and all other data contained, is generated by the school. The verifier can trust that the person is a student if the photo in the ID image is of the person attempting to authenticate as the prover. The downside to only using the ID image in identification would be that the authentication process would only be protected from replay attacks by image verification. Allowing intruders to relatively easy access to an ID image and attempt to authenticate as the student. Combining the certificate and the ID image, serves to protect the MS-ID against being copied. The users trust in the integrity and security of the MS-ID would be significantly crippled if it was possible to copy the MS-ID. Copying the MS-ID would only have limited security repercussions because the image would still be of the student owning the original MS-ID.

Secure Element Alternatives; The secure element owned by the Mobile network operators (MNO), i.e. ,the UICC also known as SIM card, offers Over The Air modification capabilities which could generate a very different registration scenario if used. Transferring the MS-ID to the smartphones using the secure connection offered by the MNO’s could ease the registration process for the student. The student might only be obliged to download the MS-ID application and it would automatically be personalized by the MNO. This approach is not chosen to support flexibility in the choice of secure element for future works.

6.2.4.2 Authentication with MS-ID for identification (AMS)

This section will present the AMS protocol for authentication with MS-ID when a student wants to authenticate his or her ID with an authenticator (school authority or third party). The process involves two NFC enabled devices, and a message exchange with the purpose of authenticating the student only. This protocol does not implement mutual authentication, its only one-way authentication. The authentication mechanism implemented here is very similar to asymmetric challenge-response mechanism described in

section 2.3. The difference between the third party authenticator and the school employee authenticating the student is that the school employee has previously been in contact with the SA server and downloaded an exam attendance list. Through this list the application will keep track of which students have authenticated themselves and which have not. The authentication protocol is the same in both scenarios and the protocol is therefore only presented once.

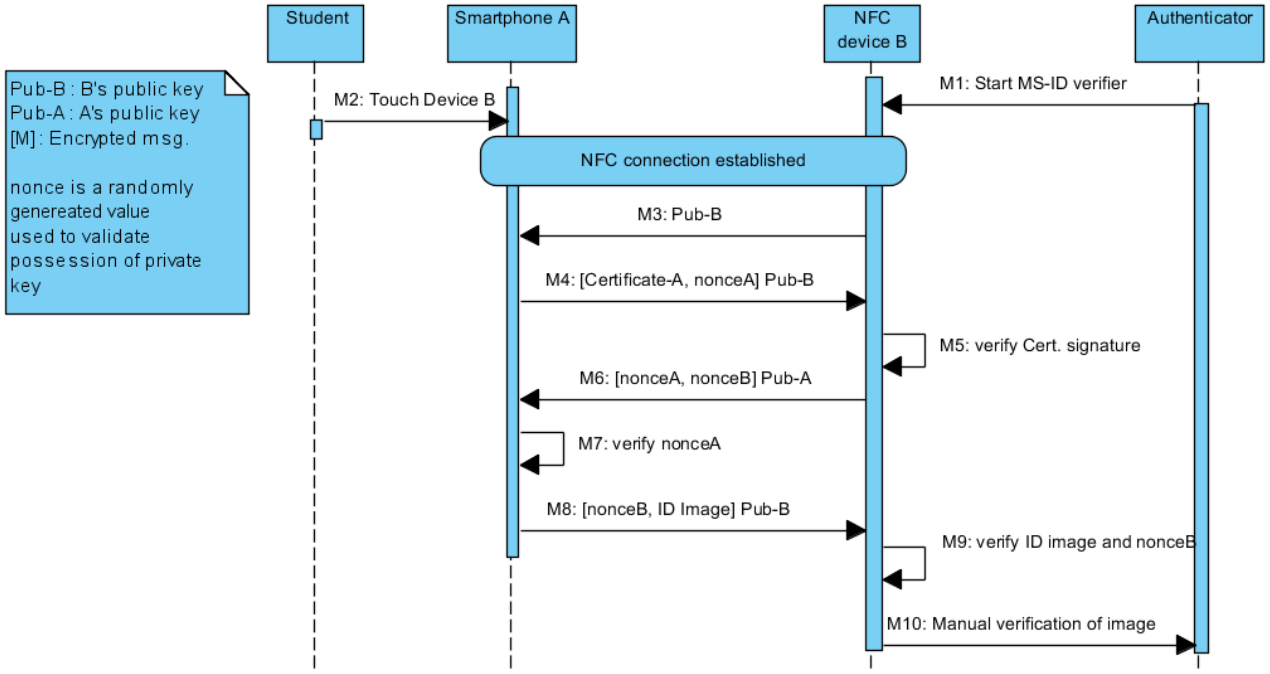

Figure 6.8 MS-ID authentication protocol (identification)

1 The scenario starts with the authenticator starting the MS-ID Verifier app (M1).
2 The student touches the authenticators NFC device with his Smartphone, this establishes an NFC connection (M2).
3 Device B transmits its public key to Device A, to provide confidentiality for the rest of the information transferred (M3).
4 Device A transfers its SA signed MS-ID certificate *Certificate-A* along with a *nonceA*, encrypted with device B's public key *Pub-B*(M4).
5 Device B decrypts the message using its private key and verifies the authenticity of the certificate signature using an SA certificate (M5). (Note: Communication ends at this point if device A has an invalid certificate).
6 Device B generates its own nonce *nonceB* and encrypts it using A's public key along with *nonceA* and transmits it to device A. (M6).
7 Device A decrypts (M6) using its private key and verifies that *nonceA* transmitted in M4 is the same as the one received in M6. If so Device B has proven its possession of the private key corresponding to public key B thereby making sure the connection has not been hijacked (M7).
8 Device A decrypts *nonceB* from M6 to prove its possession of private key A, and transmits *nonceB* along with the ID image. The message is encrypted with public key B. (M8)

9 Device B decrypts message M8 with its private key and compares the random value *nonceB* from M8 with the *nonceB* transmitted in M6, if so device A is authenticated. The ID image authenticity is confirmed, by verifying the signature of the ID image using an SA certificate. (M9)
10 The authenticator views the ID image and verifies that the student matches the image provided thereby performing two-factor authentication(M9).

6.2.4.2.1 Rational for Identification

In this section the properties of the AMS protocol will be presented along with alternative design opportunities.

Offline Authentication; The authentication of the student in the Identification and Identification during transaction scenarios is designed not to require internet connectivity. This is chosen because networks connectivity on mobile devices is not reliable enough in a dynamic mobile environment. Occasionally low bandwidth or connection loss in certain areas would result in low availability and a degraded user experience. Considering that the MS-ID should be used as identification at university examinations the availability of the system is critical. The system is also designed so that there is no connection between the SA Server and the third party authenticator which could be utilized to inform the authenticator about obsolete MS-ID's. Instead the validity of the MS-ID only depends on the validity period included in the MS-ID which we suggest is set on a per semester basis. This is chosen because it simplifies the system and exempts the third party authenticator from any requirements such as an updating service might put on the authenticators. Furthermore, no strict requirements are put on the system to prevent students from authenticating at third parties in the immediate time after leaving the school and becoming ineligible for student rebates.

Authentication Factors; The present student identification card relies on two factor authentication, one being the possession of the card and the other being resembling the person in the photo on the card. AMS utilizes same two factors due to the advantage of having a human authenticator present to verify the student photo during authentication, without a human authenticator present the only other viable factor would be biometric authentication which is presented in section 5.1.2. The reason that biometric authentication is the only alternative is that the scenarios in question must be designed to thwart impersonators in cases in which the actual identity holder deliberately helps the impersonator. Against this threat authentication factors such as "*something you know*" and "*something you have*", are weak because they can easily be shared. Most security system relies on trust to the individual user not to reveal security tokens such as the users password. This trust might be justified in cases where the user does not benefit by helping another impersonate him. In the Identification scenario, the student could benefit by letting another person go to the exam and in the Identification during transaction scenario friends of the student could gain rebates at stores by impersonating him.

Privacy; The student certificate transferred in message M4 is encrypted using the public key of the Authenticator because it is desirable to reveal as little information as possible to potential eavesdroppers. The privacy of the student could be compromised if the certificate was transmitted in cleartext. By encrypting possible sensitive data sent to the authenticator these data are only exposed to the authenticator.

Third Party; The system is designed in such a way that the third party authenticator is not authenticated. This leads to the problem that an attacker might impersonate or even be the authenticator. If the student tries to authenticate himself before a malicious minded authenticator the system can not prevent the authenticator from gaining privacy information such as the students name and photo. In case the authenticator is not malicious, the system has to protect the communication from eavesdropping and from an attacker trying to hijack the communication. Because the authenticator is not authenticated there is a possibility that an attack may impersonate the authenticator and take over the communication. To prevent hijacking like the above to occur, the system performs a key exchange at the first available instant i.e. M3, in which the NFC device sends the public key to the Smartphone. This public key protects the rest of the communication from an attacker trying to impersonate the authenticator through a masquerade attack, as described in section 3.4.4, and thereby obtaining privacy information regarding the student. The public key also protects the student from exposing any privacy data to potential eavesdroppers.

Man In The Middle Attack; The AMS protocol is vulnerable to MITM attacks, described in section 3.4.4, where the attacker impersonates both parties and thereby gains access to all data exchanged without interrupting the protocol. An attacker would overwrite the message M3 with his self-generated public key through a data modification attack. By doing this the data contained in message M4 would be encrypted with the attackers public key and he would have full access to the data within. He could then take the data and encrypt it with the public key from the original M3 and send it to the NFC device. The next message which is M6 would be relayed by the attack directly to the Smartphone. The attacker will then have access to the data in the last message M8, which he will encrypt using the original public key and send the message to the NFC device. Thereby revealing the identity of the student without interrupting the protocol. This threat could be countered by embedding a hash value of the public key in M6 and thereby authenticating the public key. It is also possible for an attacker to use a relay attack this allows the attacker to authenticate as the student somewhere else by relaying the messages from the attackers smartphone to the students smartphone and vice versa. This attack however is impractical because the attacker would have to authenticate at exactly the same time as the student and intercept the NFC communication. Moreover, the attacker has to fool the authenticator into believing he is the person on the photo that appears on the student ID Image, which is very difficult unless the attacker looks very similar to the owner of the ID. However, both of these attacks require sophisticated and expensive equipment to perform.

Replay Attack; Replaying the messages eavesdropped by listening in on the communication between the smartphone and the NFC device might enable an attack in successfully authenticating as a student. This is the case if the public key Pub-B owned by the NFC device has not been changed since the messages eavesdropped by the attacker were sent. In addition, the random challenge generated by the NFC device nonceB must also be the same as in the messages eavesdropped. In an ideal scenario with a perfect random number generator, which generates numbers of 128 bits there is one in 2^{128} probability of two numbers in a row being equal, which makes attacks by replay very unlikely.

Third Party Authentication; The third party is not authenticated because it would require the system to implement an authentication procedure for third parties. This procedure would be a hassle for the third parties to go through and might not be worth it when dealing with student discounts. Another disadvantage by requiring third parties to be authenticated is that the costs of managing the system would increase due to the added registration and distribution costs. The downside of not authenticating third

parties is that students private data can be revealed to people that the system cannot guarantee are valid business owners or clerks. However this does not outweigh the advantages of not troubling the third parties with an authentication procedure.

6.2.4.3 Authentication with MS-ID for access control (AMA)

This section will present the AMA protocol for authentication with MS-ID, when a student wants to access restricted institutional facilities (access control scenario). The process involves two NFC enabled devices and a message exchange with the purpose of authenticating the access control terminal and the student. This protocol implements mutual authentication by authenticating both participants with certificates signed by the SA server. The authentication mechanism implemented here is very similar to asymmetric challenge-response mechanism described in section 2.3.

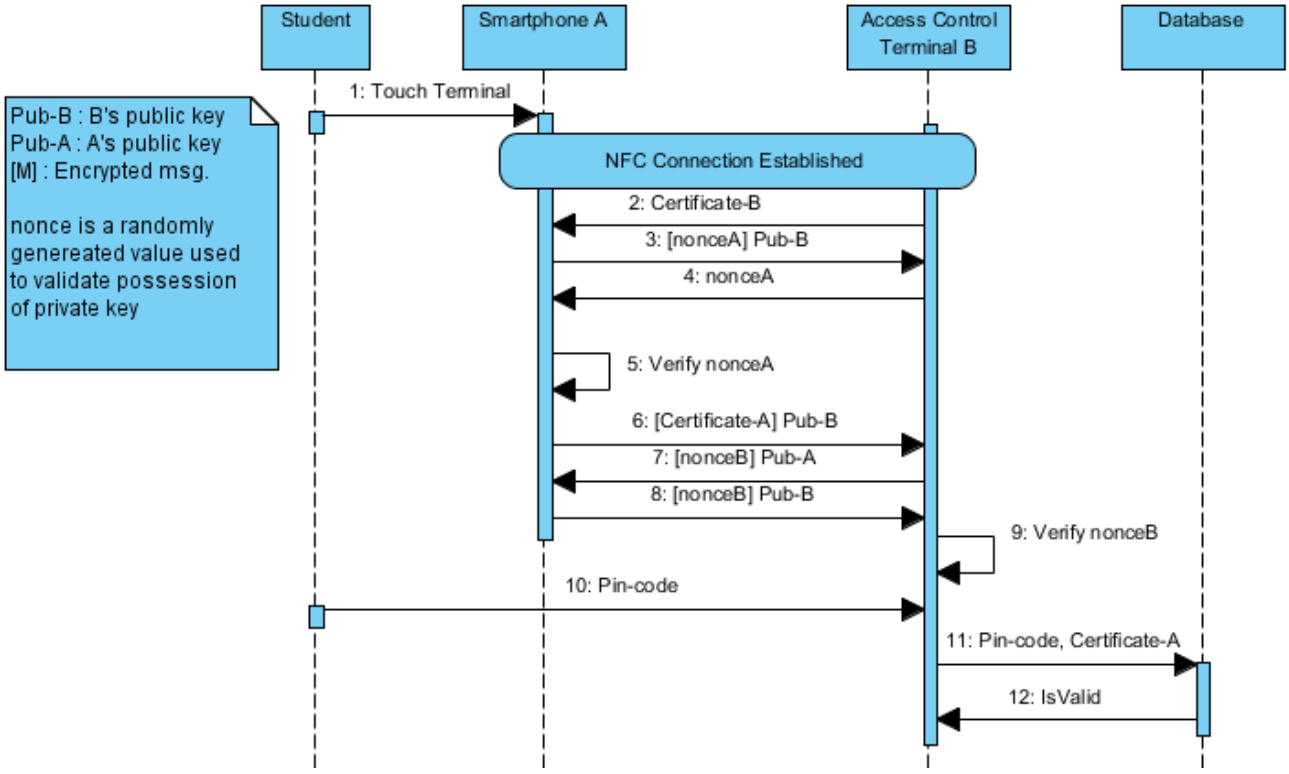

Figure 6.9 MS-ID authentication protocol (Access Control)

1 The scenarios starts with the student bringing his smartphone (A) into proximity of the Access control terminal (B), this establishes an NFC connection (M1).
2 The Access control terminal (B), provides the smartphone (A) with an SA signed certificate (M2).
3 The smartphone verifies the authenticity of certificate from M2, then generates the nonce *nonceA* and encrypts it with B's public key provided in M2 (M3).

4 The access control terminal proves the possession of the private key, belonging to the certificate transferred in M2, by decrypting *nonceA* and transmitting it in clear text to A thereby authenticating B (M4).
5 The smartphone verifies if the random value of *nonceA* received in M4 is the same as was sent in M3, if so the access control terminal is confirmed as being authentic (M5).
6 In order to authenticate the student the smartphone now sends the students MS-ID Certificate encrypted with B's public key (M6).

7 To authenticate the smartphone the access control terminal verifies the authenticity of certificate from M6 using an SA certificate, and then the nonce *nonceB* and encrypts it with A's public key obtained in M6 and transmits it to the smartphone (M7).
8 The smartphone decrypts the random value *nonceB* with its private key. *nonceB* gets encrypted with public key B. *nonceB* is then transmitted back to the access control terminal, thereby authenticating the smartphone(M8).
9 The access control terminal verifies that the random value of *nonceB* transmitted in M7 is exactly the same that was received in M8, if so the smartphone has been confirmed as authentic (M9).

10 Both participants are authenticated and have exchanged public keys the second authentication factor takes place, the student is requested to enter his pin-code on a keypad connected to the access control terminal(M10).

11 The access control terminal compares the input pin-code with the database equivalent, if correct the student is allowed access.

6.2.4.3.1 Rational for Access control

In this section the properties of the proposed solution to the Access control scenario will be presented. Alternative design opportunities will be presented and discussed.

Mutual Authentication; The Access control scenario is designed to achieve mutual authentication between the two parties the Student and the Access control terminal. Authentication of the Access control terminal is done, because it enables the smartphone to only disclose sensitive identity information to the terminals controlled by the SA server. This renders attacks on confidentiality by man-in-the-middle useless because the sensitive data is encrypted with public keys which are obtained through SA signed certificates.

Relay Attack; Relaying the messages between a students smartphone and a remote access terminal cannot be prevented. However, a relay attack will only enable the attacker to authenticate one of the two authentication factors necessary to gain access, i.e., the certificate. The pin-code, however, will not be revealed through a simple relay attack which would only imitate the students smartphone through wireless communication. More advanced relay attacks where a fake access control terminal is replaced with a legitimate terminal, allows the attacker to learn the students pin-codes and successfully relay the full protocol to a remote access control terminal. Because of the relatively low importance of the assets protected by the system compared to the complexity of such an attack, steps have not been taken to relieve the system of this threat. However, the most tangible solution to minimizing this risk is to include location information in the authentication protocol. This enables the access control terminal to determine if the student is within reach of the terminal or if the protocol is being relayed. Another solution may be setting

timing constraints on the communication between the terminal and the smartphone, thereby not allowing time for an attacker to relay the messages successfully. Both these solutions are described in depth in [57].

Authentication Factors; The two factors of authentication i.e. the smartphone and the pin-code are chosen because the smartphone would represent "*something you have*" whereas the pin-code represents "*something you know*". A possible additional factor which represents "*something you are*" could increase security. However, as there is no human authenticator to verify the identity of the student through a photo or other characteristics the increase in security might become expensive. Devices reading biometric data installed on every entrance to the school facilities might become a costly investment. Increasing security more than necessary for most purposes. However, as presented in section 5.1.2 biometric data can be recorded through the smartphone, opening the possibility for relatively cheap support for biometric authentication. Taking into account the reliability of the presented biometric authentication with smartphones, we have determined that the technology is not yet mature enough due to rather mediocre equal error rates.

Separate Certificate Alternative; Even though the certificate owned by the student is used in both the Identification scenarios and the access control scenario, the system might benefit from including a unique certificate used in the access control scenario. By generating a separate access control certificate the validity period of this certificate could be extended without affecting the security of the Identification and Identification during transaction scenarios. The separation would not impact security in the same way in the Access control scenario as in the two other scenarios because the terminals have a secure connection to the SA server. Through this connection the SA server can inform the terminals about which certificates are valid and which are not, unlike in the other two scenario where the validity of the certificate only depends on the validity date. In the case that a student has lost his smartphone containing the access control certificate, the SA server can remove the validity of the certificate in the terminals even though the certificate validity date has not expired. This is not the case in the Identification during transaction scenario because the system is not designed to inform the authenticators of lost certificates. The advantages of having an extended validity period in the access control scenario are that the student would not be required to renew his MS-ID every semester if he only uses it for access control. Also relieving students from situations where they are locked out of facilities because they forgot to renew their MS-ID. However, the system has not been designed to include a unique certificate for access control because it may very well increase the cost of storing the MS-ID on the secure element because of the increased memory footprint.

Peer-To-Peer Mode;The access control terminals deployed by Aarhus University are configured to read MIFARE Classic chips embedded in the existing student identification card. The proposed solution does not utilize the existing terminals by emulating a mifare classic tag because this technology has been proven insecure [51]. Furthermore, using peer-to-peer communication allows for more advanced communications, as it is not necessary to assume one party is passive. There are several restrictions when using card emulation mode, such as being limited to 1-4 KB memory space to read and write from. The other problem is the fact that emulating a card means emulating a specific physical interface which can possibly limit device compatibility and provide some interoperability issues, since few devices support all the interfaces at the same time.

Protocol Separation; The protocol has deliberately been designed so that relay of communication between scenarios is not possible. An attacker would not be able to relay the information from the identification scenario to an access control terminal successfully.

Privacy Data; Even though the protocol has been designed so that privacy information of the student is not exposed to an attacker, the message M6 does leak some information about the student. The student certificate in M6 is encrypted because it should not be revealed to eavesdroppers. However, this does not protect the student fully because the message M6 does not change from each session, and an eavesdropper is therefore able to recognize that the same student has authenticated twice at the same terminal, by comparing the encrypted messages. This might not be a sensitive piece of information by itself, but if the identity of the students can be coupled with this message, an attacker would be able to gain information about specific students whereabouts. To mitigate this risk the access control terminal could use several different certificates with their own key-pairs so that the message M6 for a single student would be different for each certificate used. Another method to mitigate this risk would be to add a piece of randomly generate data to the message M6, this way the message would not be the same in every session. These precautions have not been implemented because the severity of a possible compromise is not great considering the assets exposed. Hence, the likelihood of such an attack is very low taking into account the effort needed to carry it out.

Privacy Continued; To strengthen privacy protection in the protocol the last message M8 is encrypted unlike the basic challenge-response protocol which sends the response in cleartext. The reason why encrypting the nonce, *nonceB,* in M8 strengthens the privacy is that an attacker could identify the student from eavesdropping M7 and M8 if the nonce was not encrypted. A prerequisite for the successful identification of the student is that the attacker is in possession of the students certificate, which he could have obtained through authenticating the student in an “Identification during transaction” scenario. By possessing the public key of the student, the nonce and the encrypted nonce the attacker can verify that a specific student has authenticated by encrypting the random number with the public key and compare it to the encrypted random number. If the data in the eavesdropped message M7 is the same as the data generated by encrypting the eavesdropped nonce from M8 with a given student public key, the attacker can conclude that the student authenticating is the same as the owner of the certificate from where the public key originated. Because of this risk the nonce from M8 is encrypted. However, this is an extreme case in which the attacker puts a lot of effort into obtaining information, which could have been achieved easier by simply following the student. As such it is not a big concern for the MS-ID. The reason it is highlighted is to attempt to discern all vulnerabilities of the MS-ID, big and small.

6.3 Implementation

This section will present the implemented prototype and clarify how and which elements of the proposed designed is implemented at the time of writing this document. To present an overview of the implementation section 6.3.1 displays a deployment diagram and clarifies the deployment of the MS-ID, MS-ID Verifier and SA Server applications. Afterwards, section 6.3.2. clarifies the inner workings of the applications and the prototype deviates from the design.

6.3.1 Overview of implementation

The implementation contains three major applications which are deployed on their respective devices which can be seen in a larger context in figure 6.1. The MS-ID application is deployed on the student smartphone and is responsible for identification and authentication of the student. For this purpose it uses two protocols, the RMS and AMS protocol which are clarified in section 6.2.4.1. and 6.2.4.2. The AMS is fully implemented and works to a degree which will be presented in the evaluation section. With regards to the RMS protocol every separate component which is used in the RMS protocol is implemented, the full process however is not automated and as such must be performed manually.

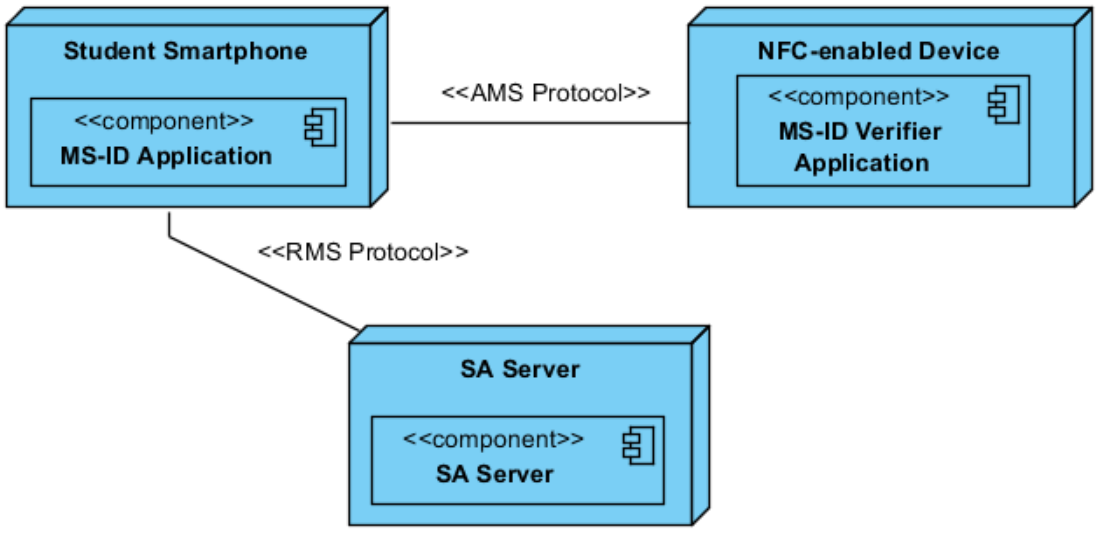

Figure 6.10 MS-ID Deployment diagram

The SA server component is deployed on the SA server and is responsible for registering students and generating MS-IDs. The MS-ID verifier application is deployed on the NFC-enabled device which in the implementation of the prototype is a smartphone, it could however have been a tablet or any other NFC-enabled device, both smartphones utilize Android as a platform. The access control scenario has not be implemented at all, although some components from the other scenarios can be reused in this.

6.3.2 MS-ID Application & MS-ID Verifier application

The MS-ID application & MS-ID Verifier application have been developed in Eclipse indigo using Java 1.5 and the Android Development Tools (ADT) plugin. The applications have been developed for the Android 4.1

(Jellybean) environment. Using Eclipse and ADT plugin in such a way that the functionality offered by the applications can be easily used both in the smartphone as well as on the PC. This approach has proven very useful for testing and debugging the application.

The prototype has been tested using two Samsung Galaxy S3 GT-I9300 NFC phones. Since the applications are developed specifically for Android they are not directly portable to other smartphone operating systems, however the cryptography library used is also available for Apple's IOS, so the amount of code rewriting required to port the application should be minimal.

As required by the proposed design, the applications contain a Certificate Authority (CA) Public key certificate, which is generated by the SA Server and included in the application resources at compile time. Details surrounding the creation and format of the certificate will be explained in the SA Server section 6.3.2.

The integrity and source authenticity of the sensitive data as well as the application itself is ensured by signing the application package file (APK) containing the sensitive data and the application code.

The implemented prototype deviates slightly from the proposed design due to some significant limitations encountered with the NFC API in Android 4.1. In Android 4.1 the NFC API is constrained by a UI overlay called android beam, which only allows one NFC message to be sent pr. user interaction. Android beam cannot be disabled without disabling a device's ability to transmit NFC messages. This means the NFC protocol cannot be implemented as designed, and has led to modifications in the design. The chosen modification has been to include Bluetooth communication into the authentication mechanism. NFC is still used to transfer some cryptographic data and Bluetooth information such as MAC address to setup a Bluetooth connection.

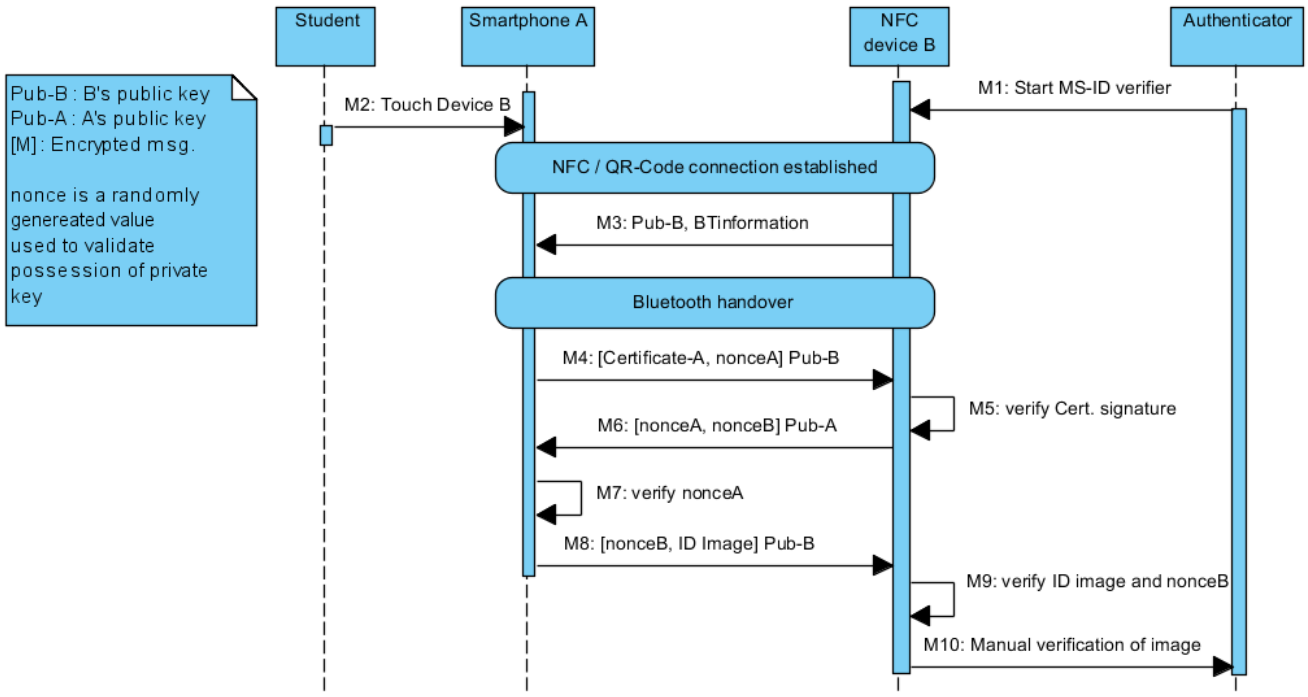

Figure 6.11 MS-ID Authentication (Identification scenario: as implemented)

As figure 6.10 shows, the implemented protocol is very similar to the designed protocol from figure 6.8. However, there are two major differences Bluetooth handover is added and a secure element is not used. The Bluetooth handover is achieved using a Bluetooth socket and the Just Works pairing model presented in section 4.3 which allows for non-paired devices to connect to each other on the basis of a MAC address. The connection is kept confidential by the Bluetooth protocol and by using public key cryptography to encrypt and decrypt messages transferred over Bluetooth. However, the security provided by the Bluetooth protocol when using the Just Works pairing model does not protect against MITM attacks. This is why encryption using the public key cryptography is also used. Both the MAC address and the public key are sent over the initial NFC message to setup the Bluetooth connection.

Bluetooth can be used as a means of compensating for the limitations of the NFC implementation on Android, the possibility of opening up for support of non-NFC devices has been discovered through this mechanism. QR codes can transfer data through the use of the camera included in almost all smartphones. QR codes can bridge the gap in the market between NFC enabled smartphones and those without. The implementation can be seen in figure 6.11. the protocol implements QR code technology as an optional means transferring data used to establish a Bluetooth connection. The procedure using QR codes is very similar to the procedure using NFC in the fact that it transmits the necessary data for setting up a Bluetooth connection, however, it is slightly less usable than NFC in this regard, since the user has to manually aim and focus on the QR code with the camera. NFC and QR codes in this specific instance are somewhat similar in the features they offer however, the QR codes have limited data space and have to generate a new code every time the data changes.

With regards to the secure element, it has not been implemented due to the limitation of the secure element API available in Android. The developers of Android have chosen not to make the API for the secure element publicly available, due to the risk of hardware damage involved if handled improperly by third party developers. Hardware damage can occur within certain secure elements if several failed authentication attempts occur, which would cause the secure element to self-destruct. There are ways around the API limitations by replacing the mobile operating system with a patched or an entirely custom ROM. There are several projects and communities providing such modifications, such as The Open NFC Project[1] which is run in collaboration with security firm Inside Secure, and Cyanogen Mod[2] which is community driven. These approaches provide access to secure elements which have a link to the application processor, such as an embedded secure element or an external secure memory card.

6.3.2.1 Application Structure

The structure of the MS-ID application and the MS-ID Verifier application are almost structurally identical, they are only separated by behavior expressed in the MS-ID core. The application structure can be subdivided in eight main components:

[1] http://www.open-nfc.org/
[2] http://www.cyanogenmod.org/

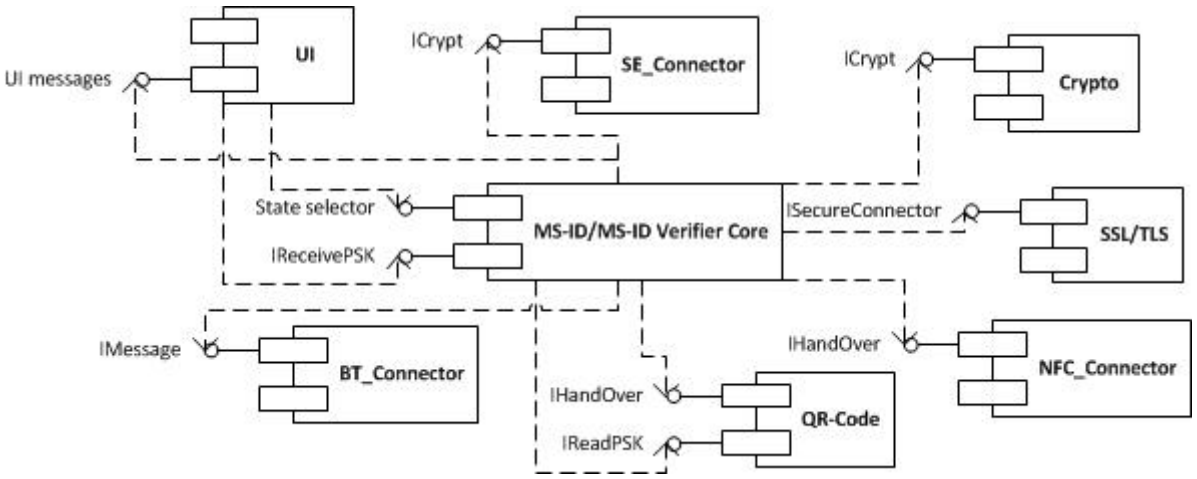

Figure 6.12 MS-ID Application and MS-ID Verifier components

6.3.2.1.1 MS-ID/MS-ID Verifier Core

This component is actually two separate components however, it is depicted as one component in the component diagram to illustrate that there is little structural difference between MS-ID and MS-ID verifier. Differences between the implementation of the MS-ID application and the MS-ID verifier application are mentioned in the components where the differences exist.

The MS-ID core is implemented to support the facilitation of the MS-ID in the scenarios described earlier in this section, Registration, identification, identification during transaction and access control.

The interaction between MS-ID Core and UI takes place via the UIMessages interface and state selector class. The former is an interface that must be implemented by the UI in order to receive notification about progress of ongoing tasks. The latter is the main class of the MS-ID core component which offers the main services which represent the scenarios mentioned in section 6.2, the user selects the scenario through the UI. Furthermore, the user has an option of choosing whether to use NFC or QR codes as the handover technology.

To fulfill these scenarios the MS-ID core utilizes the remaining components in tandem to achieve its goals. The individual components will be explained in detail after this section.

It must be remarked that the MS-ID core stores sensitive information in application memory. This is secure for most purposes as long as the mobile device platform is not modified as explained in section 5.3.1.

The MS-ID Verifier core is implemented in very much the same way as the MS-ID with regard to the interaction with the UI. However, it does not offer the user a choice of scenario as it is only used in Identification during transaction scenarios.

6.3.2.1.2 User Interface (UI)

The UI is presented on a touchscreen capable of registering touch input from the user. The Android built-in features provide the user input to the UI component. Due to the fact that the Access control scenario is not implemented the UI only presents choices for which technology the user wants to use in the Identification during transaction scenario, either NFC or QR codes. The default technology selected is NFC which the user

can then choose to change to QR codes. In figure 6.13 and figure 6.14 the course of the Identification scenario with NFC is displayed for the MS-ID Verifier application and the MS-ID application, respectively. Image number one is the starting screens on the verifier, i.e., the MS-ID application as well as on the prover, i.e., the MS-ID application. When the two smartphones NFC antennas come within reach of each other the second image is displayed. Hereafter, the verifier touches his screen which engages the AMS protocol, which in turn displays image three while executing. Upon the conclusion of the AMS protocol, if the received MS-ID is verified, the authenticator is presented with a digital representation of the students ID card, with information about the ID cards validity as evident from image four.

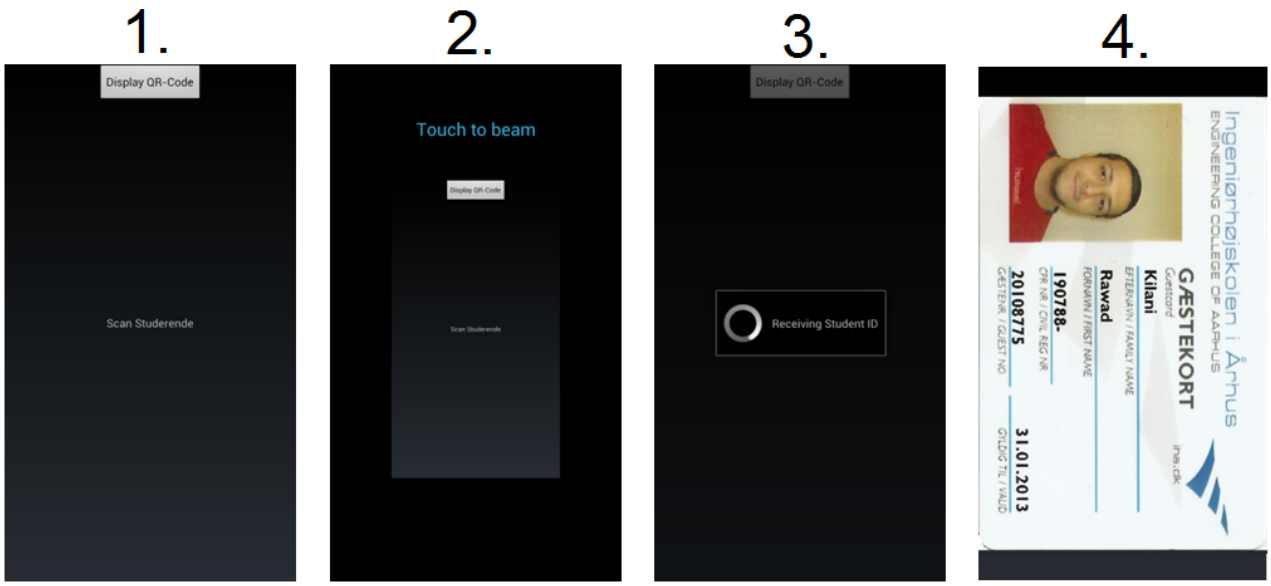

Figure 6.13 MS-ID Verifier Application performing Identification with NFC

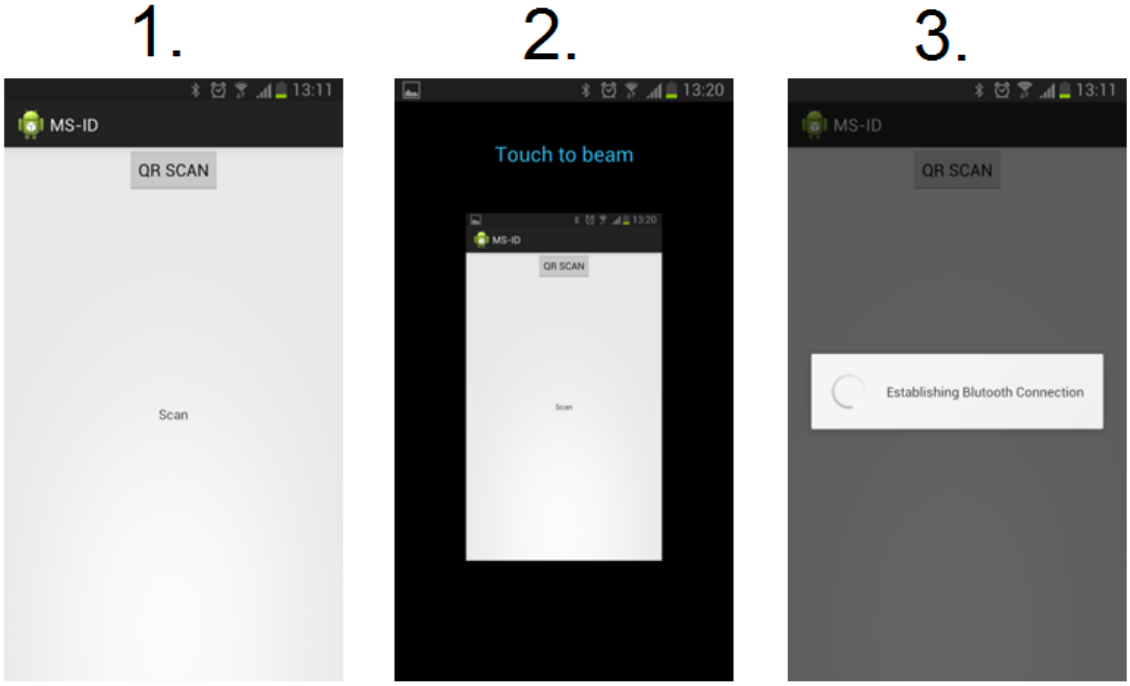

Figure 6.14 MS-ID Application performing Identification with NFC

In figure 6.15 and figure 6.16 the images of the UI are presented in the order they are displayed during Identification scenario with QR codes. Figure 6.15 presents the MS-ID Verifier application and figure 6.16 presents the MS-ID application. The scenario is engaged by both parties by pressing the Display QR Code button and the QR SCAN button respectively. On the screen of the MS-ID Verifier application a QR code is displayed which is scanned by the student using the MS-ID application as seen in image number two. Hereafter the AMS protocol is initiated and image number three is displayed in the MS-ID application. Upon completion of the AMS protocol, if the received MS-ID is verified, the MS-ID Verifier Application displays image number three which contains the digitalized student ID card, with information about the ID cards validity.

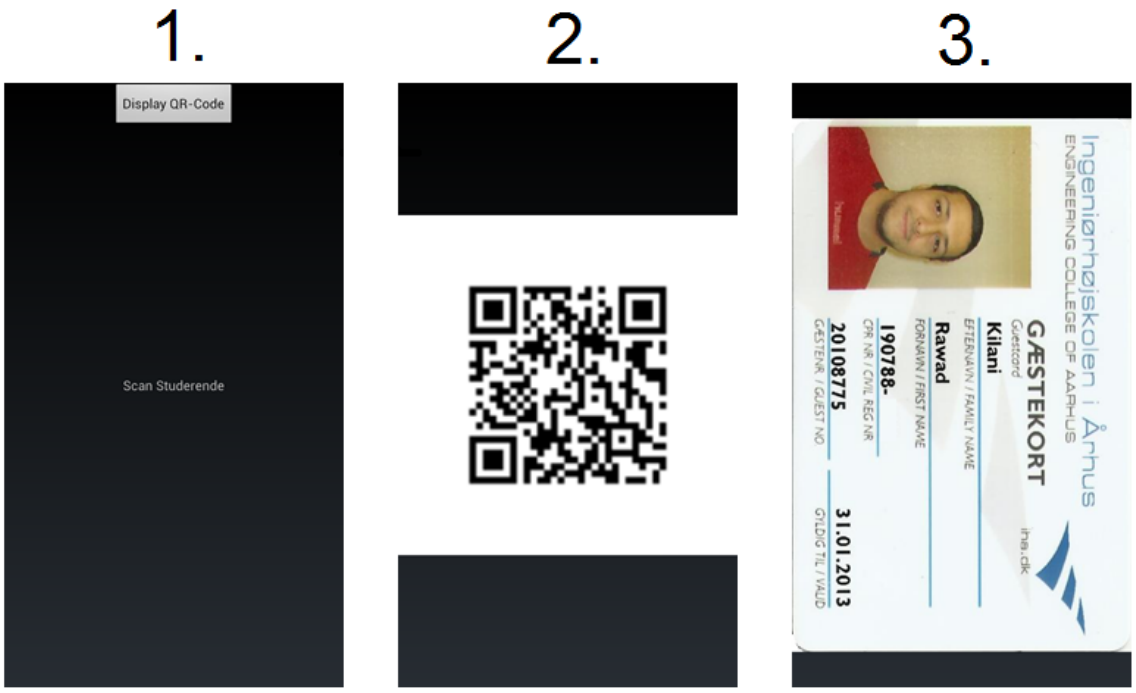

Figure 6.15 MS-ID Verifier Application performing Identification with QR-codes

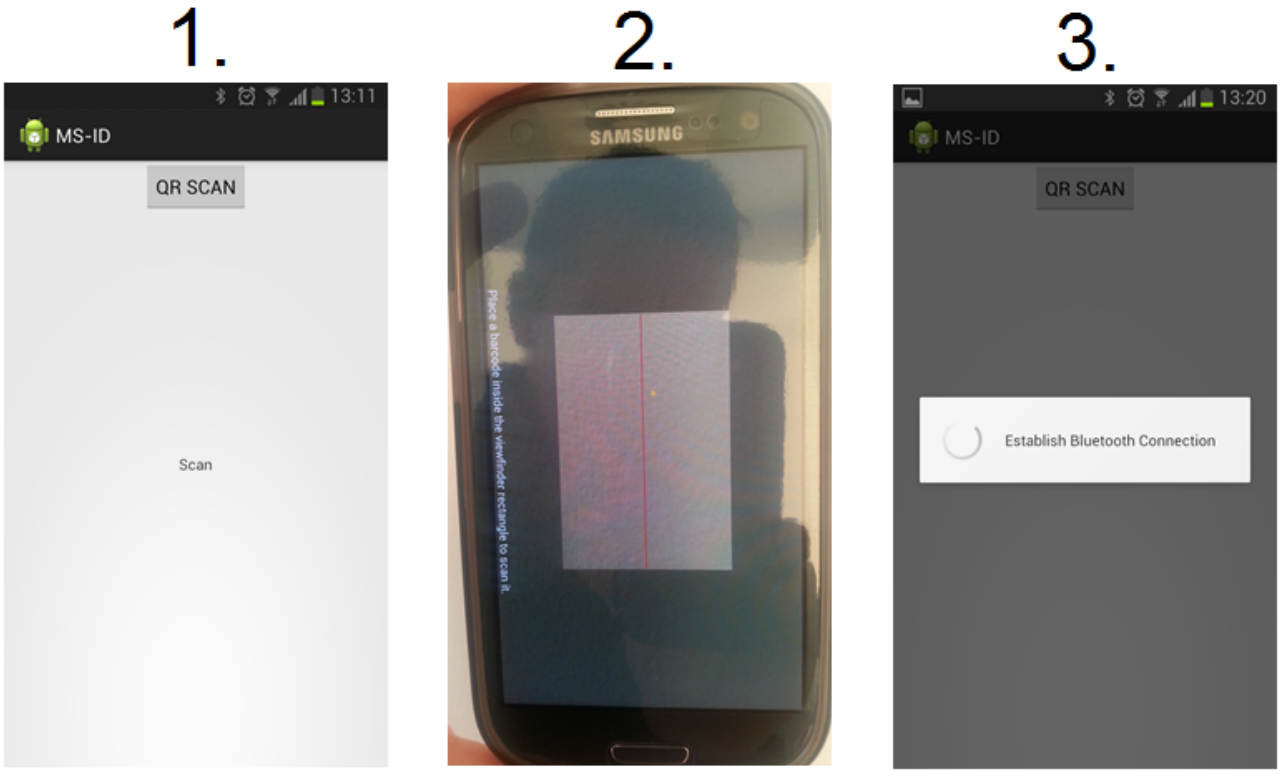

Figure 6.16 MS-ID Application performing Identification with QR-codes

6.3.2.1.3 Crypto

The Crypto component is supported by BouncyCastle v.1.47 library[3], which contains functions which facilitate encryption, decryption and signature verification. Furthermore, the Crypto component uses BouncyCastle to generate RSA key pairs with a key length of 1024. It is estimated that a key length of 1024 is capable of protecting assets of low critical value. The RSA public key cryptography algorithm was chosen due to the nature of RSA keys. The RSA keys are a product of two prime numbers, as such it is possible to optimize RSA to perform signature verification at a significantly shorter timespan than otherwise [25]. However, this comes at an increased time to perform signature generation, but because signature generation is handled by a powerful server, it is not as time critical and does not happen very often compared to signature verification. The encryption, decryption, signature generation and verification is implemented according to the PKCS#1 standard [4]. The specific the cipher suite used is SHA1 with RSA. Key pairs are stored and transmitted in a format specified by PKCS#8 standard [5].

The Crypto component implements the ICrypt interface, which offers encryption/decryption and signature verification services.

[3] Java version of BouncyCastle 1.47: http://www.bouncycastle.org/java.html

[4] RSA Laboratories Public key cryptography standards#1: http://www.rsa.com/rsalabs/node.asp?id=2125

[5] RSA Laboratories Public key cryptography standards#8: http://www.rsa.com/rsalabs/node.asp?id=2130

6.3.2.1.4 SE_Connector

The SE_Connector is a component which is responsible for the communication with a secure element. The prototype does not currently contain a secure element, and as such the component is incomplete and not tested. The SE_Connector implements the same interface as the Crypto component and as such offers the same services, but they would have been performed on the secure element instead of on the application processor. Furthermore, implementing the ICrypto interface means that the application logic will have a minimal impact from the addition of a secure element in the future.

6.3.2.1.5 NFC_Connector

NFC is implemented using the Android NFC API. The API uses NDEF messages as a data format to transfer data over NFC. The prototype uses peer-to-peer mode when communicating using NFC, and takes advantage of a connection-oriented mode for higher quality of service (QoS) during communication using LLCP sockets as described in section 4.2.

The NFC_Connector component implements the IHandOver interface which is identical to the interface implemented by the barcode component. The two components implement the same interface because they have a similar role and purpose from the MS-ID Cores perspective.

6.3.2.1.6 BT_Connector

Bluetooth support in the BT_Connector component is implemented through the Bluetooth API offered by the Android environment. The Bluetooth API does not include support for establishing Bluetooth Low Energy connections.

Both the MS-ID applications are coded to support the two communication roles available in bluetooth i.e. client/server since they implement the same component. Support for both roles is achieved by choosing the device which initiates a connection via NFC or QR code to become the server. In the context of the AMS protocol the MS-ID Verifier would always be the server and the MS-ID application would become the client.

6.3.2.1.7 QR-Code

The QR-Code component is included to explore the possibility of using another technology to handover the connection to bluetooth. To implement the transfer of cryptographic keys and data used to establish the Bluetooth connection between the two applications by barcodes, the Quick Response Codes (QR Codes) specified in ISO 18004, has been chosen. This is because support for this barcode format is widespread, it is read fast and it can contain the desired amount of data. For scanning and generating the QR codes the open-source library ZXing[6] for android has been used. The ZXing library has been ported to other platforms such as Apple's IOS.

6.3.2.1.8 SSL/TLS

This component manages the SSL/TLS connection to the SA Server which is used in the Registration scenario. The connection is of the type TLS PSK which utilizes a pre shared key to establish a secure connection. This component has not been implemented.

[6] ZXing library google code site: http://code.google.com/p/zxing/

6.3.2.2 SA Server

The SA server has like the two previous applications been developed in Eclipse indigo using Java 1.5, to minimize the difference between the applications to reduce possible interoperability issues.

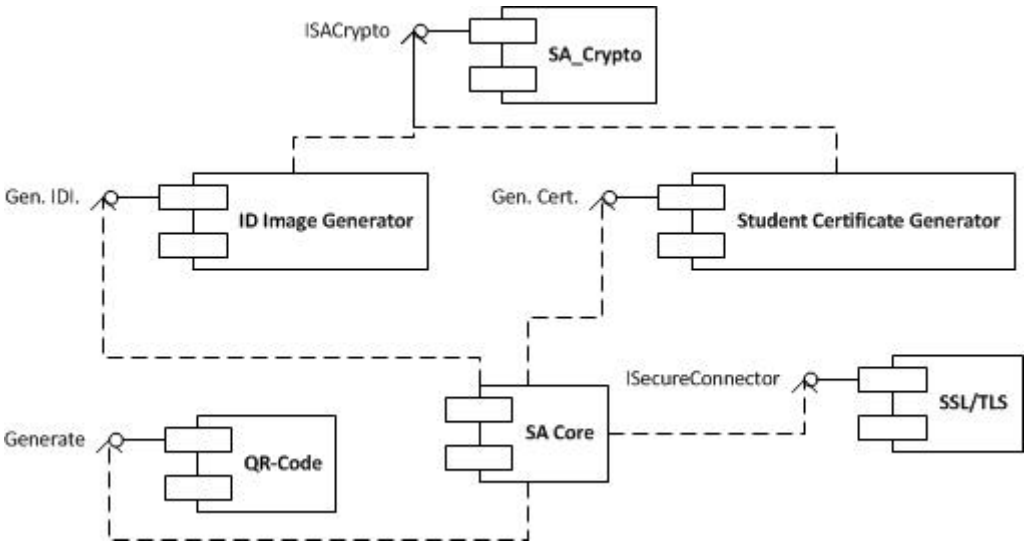

Figure 6.17 SA Server components

6.3.2.2.1 Student Certificate Generator

The certificate is generated in the SA server by using the cryptographic library in the student certificate generator which is BouncyCastle v.1.47. The libraries include an X509v3 certificate builder interface capable of constructing a certificate according to the X509 standards and sign it using a CAs private key. The certificate signature is generated using SHA1 for hashing and RSA for encryption, as specified in PKCS#1. The choice of algorithm is a matter of configuration. Therefore there have not been serious deliberations on the chosen algorithm. However, there has been discovered some weaknesses in the SHA1 hash function as presented in this paper [58]. This leads us to recommend that SHA2 be used instead of SHA1 if this protocol is to be deployed. Below is shown an example of a student certificate:

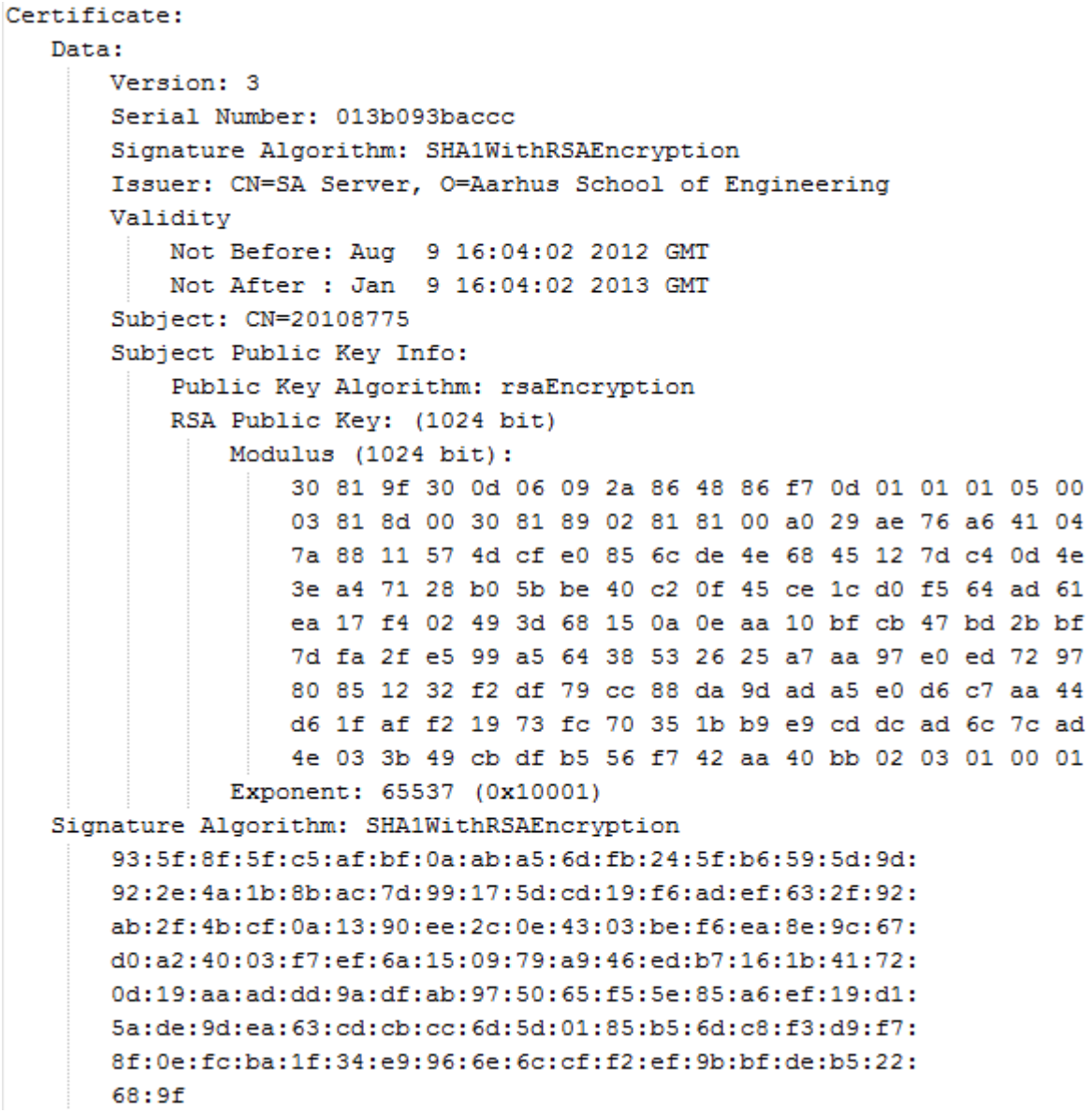

Figure 6.18 X509v3 Certificate Sample

6.3.2.2.2 ID Image Generator

The ID Image generator main purpose is to generate a signature based on the image of the students ID Card to protect its integrity and authenticity. The ID Image generator uses the same cryptographic library as the student certificate generator. To bind the ID Image with the student certificate two additional values are appended to the ID Image, the students ID number and the IDs validity period, this is evident from figure

6.19. The signature is then generated based on the three values the Image, the number and the validity period. The student certificate also includes the student number and validity period and as such can be directly compared and associated with the ID Image.

Figure 6.19 Composition of ID Image

6.3.2.2.3 SSL/TLS

The SSL/TLS component is designed to establish a secure connection to the smartphone during the registration scenario using the RMS protocol. However, the implementation of the SSL/TLS component is not complete and cannot support the RMS protocol. The MS-ID is for test purposes embedded into resources of the MS-ID application.

6.3.2.2.4 QR-Code

The QR-Code component in the SA server is exactly the same component used in 6.3.2.1.7. The only difference is the way it is used by the SA server. The SA Server utilizes the QR-Code component to communicate to the student smartphone, in a secure manner before a secure connection is established between the SA server and the smartphone. By using a QR code which is only presented after the student logs into the SA website, it is indirectly authenticating the smartphone as being in the students possession.

The QR code is used to provide the smartphone with a preshared key to complete a TLS Handshake with the SA server. This establishes a secure connection through which the MS-ID can be transferred.

6.3.2.2.5 SA Core

The SA core is responsible for the behavioral logic which controls the remaining components to perform the task of registering students. This component is incomplete, as such the evaluation of this application will be on the basis of the individual components and not the application in its entirety.

6.4 Evaluation

This section describes the tests performed to assess the performance and usability of the implemented system, and the results achieved from performing these tests.

6.4.1 MS-ID Application & MS-ID Verifier application

This section will present tests and results focusing on the MS-ID, MS-ID verifier applications and the AMS protocol. The smartphones on which the MS-ID applications and MS-ID verifier are deployed are two identical smartphones. The smartphones used are the Samsung Galaxy S3 GT-I9300 which contains a quad-core exynos processor clocked at 1.4 GHz [7].

The tests will measure the time consumptions of the critical tasks performed by the smartphones involved in MS-ID transfer and validation. The tests performed will span tasks involved in the scenarios Identification, Identification during transaction and Access control. The purpose of these tests is to assess the performance and thereby indirectly the usability of the prototype and to quantify the individual tasks influence. Results generated by the tests are presented below all measurements are obtained using the function java.lang.System.nanoTime and shown in milliseconds unless otherwise specified. The individual test cases will be presented in the following subsections, here the results will be explained and discussed.

	Connection	Encrypt	Decrypt	Transfer	Verify Cert.	Verify Image
1.	5393	5	3	500	13	10
2.	4892	9	3	600	11	5
3.	5125	5	2	500	10	11
4.	5334	4	3	400	10	11
5.	4869	4	4	400	10	11
avg.	5123	5,4	3	480	10,8	9,6

Table 6.1 MS-ID Application & MS-ID Verifier application test results.

[7] http://www.samsung.com/dk/consumer/mobile/mobilephones/smartphones/GT-I9300RWDNEE

6.4.1.1 Establish Bluetooth connection

Establishing the Bluetooth connection is done to provide a channel between the verifier and the MS-ID holder through which the student can be authenticated. The test was performed by timing the function, connect in the android.bluetooth.BluetoothSocket. Results of the tests are displayed in the column named Connection in table 6.1, which shows that the average connection time is just over 5 seconds.

This connection is established using Bluetooth 4.0 BR/EDR which does not offer as fast connection establishment as Bluetooth Low Energy (BLE). However, the interface offered by the operating system to control the Bluetooth module does not provide mechanisms to choose BLE, despite the fact that the Bluetooth hardware controller in the phones does support BLE. The opportunity to test the impact of BLE would have on performance, has unfortunately not been possible.

The smartphones used in the Bluetooth connection test were not paired prior to the test.

The Bluetooth connection test has shown to have by far the largest impact on performance of the application.

6.4.1.2 Encryption and Decryption

These mechanisms are used in the scenarios Identification, Identification during transaction and Access control. They are repeated several times in each scenario and can therefore prolong the time it takes to perform the scenario substantially if they take long to execute. Results of the tests are displayed in the columns named Encrypt and Decrypt in table 6.1 and show that the average encryption and decryption time is around 5 ms. This means that the cryptographic operations tested do not have a significant influence on the performance.

The tests are made by encrypting and thereafter decrypting the MS-ID Image which consists of 62.763 bytes. For comparison the student certificate composing the other half of the MS-ID consists of only 480 bytes. It has been chosen to perform this test using only the MS-ID Image because it is the item in the communication which is by far the largest in terms of memory. The MS-ID image is generated by scanning a student ID card and has not been optimized in terms of its memory size in any way, which makes it hard to determine its size in a final product. Because of this uncertainty it has been decided to perform the test on only the MS-ID image and with the good result it has been chosen not to investigate further on this issue.

The results of these tests might not be applicable to solutions where a secure element handles the decryption of messages. This is because the processor in the secure element is very likely not to have nearly the same computational power of processor the test is run upon. However, there is no present opportunity to test if this issue will influence performance.

6.4.1.3 Bluetooth message transfer

Transfer of data through the established Bluetooth connection is done to authenticate the student to the authenticator. Data transfer speeds of wireless communication technologies are often influenced a lot by the environment in which they operate. Therefore conclusive result on the subject of message transfer

speed require extensive test which is deemed out of scope for this project. However, the results do give a good insight into the general transfer times.

The test is performed by displaying a UI message on the smartphone executing the MS-ID Application when the transfer is initiated and a similar UI message on the screen of the smartphone running the MS-ID Verifier Application when the data is received. The interval between the two UI messages is then timed using a stopwatch on a regular wristwatch with time interval 1/10 second. The transferred data was the MS-ID Image which consists of 62.763 bytes. Results of the test is displayed in table 6.1 in the column named Transfer, they show an average transfer time of just under half a second. This result is tolerable and the time spent on data transfer will not have significant negative influence on the performance.

6.4.1.4 Certificate Validation and MS-ID Image Validation

Validating the student certificate and the MS-ID Image is performed by the MS-ID Verifier Application to verify the authenticity of the student. The test is performed by measuring the time it takes to verify the certificates signature and the MS-ID Image. Results of the tests are presented in table 6.1 in the columns named Verify Cert and Verify Image, they show an average time of 10.8 ms pr. verification of the certificate and 9.6 ms pr. verification on the MS-ID Image . This is estimated to have no significant influence on the performance of the system in general.

6.4.2 SA Server

This section will present the tests and measurements for tasks performed by the SA server and the RMS protocol, for the purpose of assessing the performance capabilities of the SA server. The two most critical tasks the SA Server has to perform is the ID Image signature generation and generating a student certificate, as these tasks involve cryptographic operations and have to be executed for each student wishing to use the MS-ID application. The QR Code generation test is included as it is also a part of the process.

The SA server is in these tests represented by a laptop with the following specifications:

Windows 7 Professional operating system, Intel Core 2 Duo CPU - 2 GHz - 4 GB ram.

The RSA key size used is 1024 bits, and the crypto library used is BouncyCastle 1.47. For measurements the java class System.NanoTime is used, to measure the time consumed by the operation. Five measurements are made for each test and the average is calculated.

	ID Image	Stud. Cert	QR Code
1.	42	1065	1388
2.	41	985	1069
3.	40	972	1206
4.	40	963	1066
5.	39	1344	1062
avg	40,4	1066	1158

Table 6.2 SA Server test results.

6.4.2.1 Generation of ID Image signature

The generation of the ID Image signature is one of the fundamental roles of the SA server, and as such has been estimated as being the most time consuming task. The test is especially significant considering the optimization of the RSA algorithm to favor signature verification at expense of signature generation. The ID Image used in the test is an image of a student ID card which has been imported digitally and stored on data storage accessible to the SA server. The read and write times to retrieve ID Image and store the signed ID Image have been included in the measurements.

The result of the test can be seen in Table 6.2. The average time required to sign an ID Image is measured to 40.4 ms. This is a much lower time requirement than first anticipated considering the optimization of the RSA algorithm.

6.4.2.2 Generation of Student Certificate

For the purpose of testing the generation of a student certificate, a certificate which conforms to the X509 standard is created using BouncyCastles certificate builder module. For input to the certificate builder the author's student details and information has been used in the test. The student certificate is signed using a previously generated X509 self-signed certificate. The certificate signature is generated in a similar manner as the previous test performed by using an optimized RSA algorithm.

The result of the test can be seen in Table 6.2. The average time required to create and then sign a certificate is measured to 1066 ms. As the previous test which only included generating a signature took 40.4 ms, it can be assumed that the majority of the time this test required was to generate the certificate.

6.4.2.3 Generation of QR Code containing PSK

To test the time required to generate a QR code containing a PSK, the library ZXing has been used and a 1024 bit long random value generated by BouncyCastle has been used as a PSK. The QR code task is not as critical as the two previous tests, since they are not necessarily linked to a specific student. Since the QR codes are not linked to students as such, the SA server can generate them beforehand and maintain a cache of QR codes to establish secure connections with the student smartphones. However, the SA server has to perform the task nonetheless and as such the measurements are included in the table 6.2. The measurements show that generating the QR code and the PSK is the most time consuming task for the SA Server averaging 1158 ms spent for each QR code.

6.4.3 Usability - Focus group

To assess the usability of the designed solution the prototype has been demonstrated to a focus group. The focus group consisted of 6 people who were presented the Identification during transaction scenario both by use of NFC and by use of QR-code scanning. They found that if the scenario was performed by scanning the QR code the direction of the data flow was confusing because the student ID was transferred to the smartphone being scanned rather than the smartphone doing the scanning. However, performing the scenario with NFC did not raise any confusion. To maintain the AMS protocol the first message has to be transferred from verifier to the prover. Therefore it is a necessity that the smartphone doing the verification is the one that has to be scanned. However, the first message is meant to establish a secure connection between the two parties as explained in section 6.2.4.2.1 *Rationale for Identification*, in paragraph *Third*

Party. This is achievable easily through the scanning of a QR-Code regardless which party scans whom. This is because 2D barcodes are much harder to eavesdrop compared to NFC which renders transfer of security tokens more flexible. By altering the protocol a little so that a public key pair is transferred through the QR code scanning, the usability problem could be solved without major changes to the protocol and without security repercussions.

6.4.4 Transfer of ID Image using Near Field Communication

As the system has been designed with NFC in mind, a small application separate from the prototype was developed to investigate the performance of a solution based solely on NFC. To measure the performance, an ID Image with the size 62KB is transferred through NFC. The test assumptions are exactly the same as presented in section 6.4.1. I.e. the signature is generated using 1024 bit RSA keys and SHA1 hash algorithm. The ID Image is transferred between two smartphones GT-I9300.

The test is less extensive and more informal than the previous since it is not a part of the actual prototype and only a developed for the purpose of investigating a solution focusing on NFC. To measure the performance a stopwatch was used to time the transfer.

The result was on average approximately 25 seconds, this suggests that a solution based solely on NFC is not viable with the current size of the MS-ID assets. However, this can most likely be decreased by optimizing the ID Image, which will be discussed in section 7.2.

7 Discussion

7.1 Novelty

Including mobile devices such as PDAs in an authentication mechanism can be traced back to previous work as presented in [27], published over 13 years ago. More recently the mechanisms presented in Section 5.1 *State of the Art: Mobile Authentication*, also involve using mobile devices such as smartphones.

The authentication mechanism presented in this chapter differs from these mechanisms in the fact that the NFC enabled smartphone assumes an active role in the authentication process. Such a role is novel considering that it is far more common for NFC phones to be used passively in card emulation or reader/writer mode rather than peer-to-peer mode. Using NFC enabled smartphones in peer-to-peer mode allows for more advanced communication. The most critical advantage of peer-to-peer mode is that a principal does not need to assume that other party is passive, compared to the other two modes. Card emulation restricts communication options in two ways, firstly it emulates a passive target with limited space, which in the case of a MiFare card is between one and four kilo bytes memory, the space immediately becomes a problem to manage between multiple applications which need to use the card emulation interface. The second problem is the fact that emulating a card means emulating a specific physical interface such as ISO 14443-A, ISO 14443-B and others. Few devices support all the physical interfaces at the same time, this greatly limits device compatibility and can give interoperability issues.

Another point to promote is the fact that the authentication process is capable of working off-line i.e. without any external network coverage, assuming the devices have been properly registered. However, this is also the case with the current student ID Cards which operates with RFID technology and has information details printed on the face of the card. There is still a significant difference in the fact that the MS-ID prototype provides additional levels of security as presented in chapter 6, whereas the standard ID card is only protected by outdated protocols and visual verification by authenticators.

The protocols designed for the MS-ID (RMS,AMS,AMA) are novel, however, they utilize well established and widely used elements of cryptography such as public key cryptography. The protocol presented in [28] is somewhat similar to the AMS protocol presented in this document and it is also used in a mobile environment. The major difference however is the context the protocol is used in and the data transferred by the protocol. We consider this an advantage rather than a disadvantage, this follows from the fact that by reusing existing and widely studied protocols the expertise and testing that these protocols have already undergone is leveraged.

7.2 Performance

The MS-ID will be used in scenarios where the interaction between student and authenticator is time constrained as such the performance level of the prototype tested in section 6.4 *MS-ID Authentication with NFC-enabled mobile devices: Evaluation* is of utmost importance. Performance-wise the most important attribute besides security is execution time, as high execution time would greatly reduce usability as the user has no real interaction with the UI besides selecting the scenario. As such the time it takes to perform an authentication is the most critical part of the MS-ID experience aside from the product being reliable.

The combined tests of table 6.1 show that the time to perform the major tasks of the AMS protocol between two modern smartphones with quad core processors takes on average 5631.8 ms with NFC and Bluetooth. The measurement is achieved by adding the results of the tasks tested in table 6.1. An Average of 5.63 seconds to perform an authentication is a long time but not intolerable. An important note is that after the initial NFC message the devices no longer need to be in close proximity of each other to complete the remainder of the protocol. However, looking at what takes longest time it is definitely establishing a connection over Bluetooth which alone requires approximately 5 seconds. To increase the performance of the prototype, the time it takes to establish a connection must be reduced, which means an alternative technology must be utilized. There is promise in the aspiring new Bluetooth 4.0 LE which claims to be able establish a connection significantly faster at the cost of bandwidth which is not a major concern for the MS-ID prototype.

Regarding the SA server performance, it requires 2.2 seconds to serve a single student based on addition of the time required to perform each individual task from section 6.4.2. 2.2 seconds is a fairly short amount of time for a registration process, however, the user interaction with the SA server is predicted to be bursty of nature due to all new students requiring the MS-ID at the same time, at the beginning of each semester. It might therefore be advantageous to look into cloud computing for the purpose of the SA server. A cloud server would be able to adapt to the burst of requests for registering an MS-ID and shrink during times of low activity, this would make the SA server more economically viable. However, examining the security repercussions of using cloud computing to store sensitive data is out of scope for this document and more information on this subject can be found here [44].

As the system has been designed with NFC in mind, the test in section 6.4.4 was performed to show how long it takes to transfer an ID Image over NFC without the help of an additional communication technology as a handover. The result was approximately 25 seconds this suggests that a solution based solely on NFC is not viable with the current size of the MS-ID assets. However, there is significant room for optimization of the ID Image, to reduce its size and if necessary the quality of the image can be reduced. The unusually high transfer time is possibly caused by the android beam interface in Android 4.1 this is however, not confirmed.

7.3 Security

The Goal 3.1 (Server Authentication) described in section 3.3 has been enabled by the systems public key infrastructure and achieved through the SSL/TLS protocol which is widely used and accepted. Authentication of the third party authenticator has not been a goal in the designed solution due to the nature of the scenarios in question, which is explained in section 6.2.5.1 under Third Party Authentication.

One of the main goals of the system is to authenticate the student, this is done in every scenario and in many different ways. The proposed design uses the existing authentication procedure where the student uses his credentials on the school website to authenticate the student before transferring the MS-ID. To authenticate the students smartphone a pre shared key is used in the TLS PSK protocol. In the three remaining scenarios the student is authenticated by the certificate in a challenge response protocol and by either pin code or photo. This implements two factor authentication which is the minimum requirement and common practice for security systems as evident from the presented solutions in section 5.2 Electronic Identity Applications

The proposed solution is vulnerable to MITM attacks in the identification scenario and the identification during transaction scenario if the protocol utilizes NFC which affects the Goal 3.4 (Privacy). This follow from the fact that the adversary is able to modify data transmitted by NFC and mount an MITM attack on the Bluetooth communication, as described in section 6.2.5.1 *Rationale for Identification: Man In The Middle Attack*. This is feasible due to the fact that data passed between the participants is not authenticated as stated in Goal 3.3 (Data Authentication). However, it must be noted that the attack must be mounted real-time and within close proximity of the location of the authentication. Goal 3.4 (Privacy) is not completely achieved in the Access control scenario where, as explained in section 6.2.6.1 *Rationale for Access control: Privacy data*, an adversary can extract data that can violate the privacy of the student.

Many security considerations and decisions made in this project have been influenced by the fact that this project's purpose is to research the subject of mobile authentication, rather than only considering the security measurements adequate for designing a mobile student ID. This may have resulted in unnecessarily high security measures in some cases. The secure element is once such security measure that might exceed the security requirements of the student ID. Nonetheless, it might be crucial in systems that offer payment or other services that require high security. The system is also designed to offer high levels of privacy in most situations and separate credentials for MS-ID registration which may not be a crucial requirement for the MS-ID system.

7.4 Usability

The MS-ID reduces the amount of ID cards carried by students by removing the student ID card. This results in a more manageable authentication system. Current ID cards are usually printed on plastic cards and distributed through mail. This procedure is rather costly and generates in the student ID cards case, a side effect which is, the students must print out enrollment statements to receive student discounts. The fact that users have to carry the identity card is not considered to have a big influence on usability. However, this might only be the fact because wallets are still an important part in our interaction with items such as currency, receipts and ID cards. Replacements for many of the things we carry in our wallets today are already on the market or are being developed. The Google Wallet and eKvittering[8] are some of the established products on the market. In future identity systems, ID cards may reduce usability substantially more than today simply because people may not carry wallets, as the economy gradually transitions away from physical currency and becomes a digital economy.

Concerning the usability of the designed solution the Identification during transaction scenario implemented in the prototype has been assessed by a focus group described in section 6.4.3. They found that the dataflow was unintuitive when using QR-codes but not when using NFC, in the authentication process. Working with the focus group has revealed that the identification scenario using NFC was not considered unintuitive due to the fact that most participants had very little experience with NFC prior to participating in the focus group. As such there were very few expectations regarding the results of the user interactions required by NFC, in comparison to the QR-Code scanning where people expected to receive information on the device performing the scan. However, if the focus group had the same experience with NFC as they had with QR-Codes, results may have been that the data flow when using NFC was as unintuitive as the data flow when using QR-codes.

The results from the focus group hint that improvements in the UI, more specifically UI instructions could alleviate some of the confusion created by the data flow when using 2D barcode scanning.

The fact that NFC is capable of supporting payment applications as evident from the systems as presented in section 5.2.3 shows that NFC can potentially increase the value of the MS-ID for the students by implementing micro payment for canteens or printing services.

[8] http://www.ekvittering.dk/

7.5 Cost-efficiency

The MS-ID has some potential to reduce the cost of Educational institutions identification and access control needs. To clarify how this might be possible, this section will analyze Aarhus University's current identification and access control system. To make the assessment of the cost-efficiency as accurate as possible the numbers used are based on private communications with school authority employees responsible for the current ID card system [31]. The immediate benefits of the MS-ID is a reduction in the cost creating and distributing student ID cards, as it is assumed students who own a smartphone will prefer to use the MS-ID and as such do not have to receive a student ID card in the normal fashion. For the MS-ID to be economically feasible a large portion of the schools student base must own a smartphone. Studies performed in this area display a significant increase in the amount of smartphones in the households in Denmark from 2011 to 2012, which increased by 17% totaling 50% [2]. It can safely be assumed that it is the youth which is in possession of the majority of smartphones, as they are more accepting of new technology, which benefits this case.

The student ID cards deployed by Aarhus University costs 12 kr. to produce, due to the fact that they contain both and RFID Chip and magnetic stripe. Furthermore, the student details are printed on the card and the card is laminated to protect the information on the face of the card. After printing, the cards need to be registered in the access control system by school maintenance personnel for each student. After the card has been registered it is shipped by mail to the individual students. After the entire process is complete the school authorities have spent approximately 35 kr. for each student. The cost is kept down by hiring student workers to perform the manual operations of physically printing and laminating the cards.

At Aarhus University there were 34.129 students in 2011 and 6473 new students enrolling for a bachelor's degree [56]. The authors of this document estimate that 70% of the students on Aarhus University carry a smartphone with NFC capability or QR code capability based on statistics performed in [2] which estimates 50% of the population own a smartphone, the 20% increase originates from the fact that students belong to an age group more accepting of new technology. The 70% assumption provides an estimation which reveals that a possible annual savings provided by adopting MS-ID would be approximately 160.000 kr., and issuing new student ID's in cases of loss or damage would have no additional cost. To use the MS-ID the school authorities would have to make some investments in infrastructure to handle the MS-ID operations, however the savings are far greater than the required investment, especially in the long-term.

The investment in printing equipment and infrastructure to maintain the current setup of printing ID cards at Aarhus university have cost the university approximately one million kroners. A similar investment for an MS-ID like system, for identification purposes would be limited to server hardware, software and maintenance of the servers, as the databases holding the student information could be reused. However if the MS-ID system were to be used for access control as well the proposed solution would require NFC readers be deployed in order to communicate properly in peer-to-peer mode with the student smartphones. The NFC readers themselves are not particularly expensive, it is the installation and deployment which requires a large investment. Since NFC readers are compatible with the current student ID card technology (RFID), one could imagine an incremental rollout.

It is important to remark that the MS-ID is not as such bound to smartphone and can possibly function well on feature phones with NFC or QR code support.

7.6 Other Goals

The realizability of the prototype is highly dependent on several factors such as the advancement and market penetration of the NFC technology. The amount of devices using and supporting NFC is critical for deployment of a solution based on NFC, as such the state of the market is important. The market is currently in a stalemate, where manufacturers, vendors and consumers are all waiting for each other to adopt NFC [59]. However, if one examines the problem more closely it appears to be a marketing problem rather than a technical one. As it is mentioned in section 4.4 *Technology and standards: Secure element* , there are multiple options for hosting the secure element: The SIM card, secure memory card and embedded secure element. From a technical perspective, all three options are equivalent, but from the mobile network operators (MNO) perspective it is more convenient for the secure element to be under their control, as they can profit from the applications deployed into it. Further, it is claimed that by using the SIM card as the secure element the NFC applications are independent of the particular handset where they are executed. As a consequence MNOs tend to favor the former option, i.e. having the SIM card acting as the secure element. This position is explicitly stated in [60]: "*The secure element recommended by the GSMA [GSM Association] for the payment application in the mobile phone is the Universal Integrated Circuit Card (UICC), commonly known as the SIM card".*

On the other hand, having an independent secure element is more flexible as no collaboration from the MNOs is required to deploy applications into it, making the business case or these applications more attractive. However, MNOs are from their position able to influence their customers' choice of handset. As such they have a significant say with regards to the placement of the secure element.

The greatest obstacles for MS-ID specifically are not only the amount of NFC enabled handsets available but also the limited NFC infrastructure. As an example, to deploy the proposed MS-ID Access control scenario in a university, there would have to be invested a lot of effort in changing the existing access control terminals to ones that support NFC. However, MS-ID is definitely viable without access control, but it loses some of its potential usability in the fact that students still have to carry around an RFID card for access control.

It is the opinion of the authors, however, that the outlook is not as bleak as it appears: once the secure element issue is solved, it is likely that the market penetration of NFC devices will increase dramatically. At the time of writing however, the authors deem NFC to be immature based on the issues encountered during development of the prototype as well as evident from the market penetration of NFC devices and services. In any case, regardless of the particular solution to the aforementioned problems, NFC will be an interesting platform upon which authentication solutions can be built.

Until NFC becomes more mature, QR codes have proven to be an adequate alternative for the MS-ID prototype to perform mobile authentication as a short term solution.

7.7 Future Work and Extensions

There are several improvements which can enhance the prototype in general and the authentication protocol. Starting from the simplest, the UI can be improved significantly to provide more user feedback and solution based solely on NFC communication would be desirable. More importantly however, the authors would have liked to implement a secure element in the prototype to gain knowledge of the intricacies such a device provides. A secure element would be necessary to make the MS-ID resistant against modifications of the platform. Furthermore, a secure element would take the prototype a step towards a more generally applicable mobile identification platform with multiple identities separated inside the secure element. It would be preferable if the secure element conformed to the Global Platform standards which are described in section 4.4 *Technology and standards: Secure element,* to avoid compatibility issues. The next step in the research of mobile identity would be to look into a NFC enabled protocol which could support the security requirements for mobile payment or for a national ID.

8 Conclusion

Developing this project has been interesting, intensive and time consuming. The fact that many existing technologies were reused has been helpful in order to achieve an effective and somewhat efficient prototype. However, to understand the intricacies of these technologies and use them properly it has been necessary to go through a rather large amount of documentation. Furthermore, the research regarding the state of the art in mobile authentication, identity and security took longer than expected and was more extensive than initially planned. Nevertheless, it is the hope of the authors that this research presents the readers with a meaningful overview of the state of mobile authentication and a useful starting point for other people looking into the subject of mobile authentication.

Covering the entirety of the considered solution, from the design to the implementation has been an enriching experience. Especially considering the authors limited experience with information security prior to the writing of this document. As such it has been necessary to achieve a good understanding of the fundamentals regarding information security and is the reason why the information security overview chapter is rather extensive.

It is known that the implemented prototype does not withstand online active attacks such as MITM and relay attacks. However, enhancements to counter these types of attacks have been proposed. Additionally, one of the main findings of this project is that using NFC enabled smartphones for mobile authentication is technically *feasible.* In spite of this, the lack of market offer of NFC enabled smartphones has been found to be the only relevant factor threatening the feasibility of the considered solution. The reason why NFC smartphones are not yet widespread seems to be the indecision as to whether the secure element connected to the NFC interface should be owned by the MNOs, the manufacturers or the users.

Unfortunately, it is hard to predict how long it will take for the stakeholders to reach a decision regarding the secure element location. In any case, it is the opinion of the authors that a purely NFC based solution would hardly be realizable in the short term. Nevertheless, due to the fact that the considered solution implements authentication mechanism based on QR codes as well as NFC, it is certainly feasible to consider a gradual transition starting with a few smartphones which will grow as NFC market penetration increases.

The MS-ID prototype is working very well based on our evaluation and solves the identification problem Logica has presented. Furthermore, the findings regarding QR code have enabled the MS-ID system to function without NFC and therefore the system is deployable until NFC becomes more mature. With regards to mobile authentication in general the research of this project has shown that there is definitely a potential and a market for such applications, however, if NFC is the answer such applications have been waiting for is difficult to say at this stage of NFCs maturity and level of deployment.

Bibliography

[1] Mulliner, C. (2009). Vulnerability analysis and attacks on NFC-enabled mobile phones.*Availability, Reliability and Security, 2009. ARES '09. International Conference on,*695-700.

[2] Danmarks Statistik “Elektronik i hjemmet - indkomst, forbrug og priser”. [Online]. Available: http://www.dst.dk/pukora/epub/Nyt/2012/NR229.pdf , November 2012

[3] Laukkanen, M. (2007). Towards operating identity-based NFC services. *Pervasive Services, IEEE International Conference on,* 92-95.

[4] Stallings, W. (2010). Cryptography and network security: Principles and practice 5th.

[5] Kizza, J. M. (2005). *Computer network security*. New York: Springer

[6] Konheim, A. (2007). *Computer security and cryptography*

[7] Anderson, R. (2001). *Security engineering : A guide to building dependable distributed systems*. New York: Wiley.

[8] Jonathan, K., & Lindell, Y. (2007). *Introduction to modern cryptography,* Ch. 7.4

[9] Politiken.dk “Danmark slår USA på udbredelsen af smartphones” [Online].

Available:http://politiken.dk/tjek/digitalt/telefoni/ECE1549827/danmark-slaar-usa-paa-udbredelse-af-smartphones/ , November 2012

[10] Fællesforening for Danmarks Brugsforeninger (FDB) “Vi har smartphones nu - nu vil vi have tabletst”. [Online]. Available: http://fdb.dk/analyse/vi-har-smartphones-%E2%80%93-nu-vil-vi-have-tablets , November 2012.

[11] Stüber, G. (2001). *Principles of mobile communication, 2d ed*

[12] Gartner.com “Gartner Says Worldwide Sales of Mobile Phones Declined 3 Percent in Third Quarter of 2012; Smartphone Sales Increased 47 Percent”. [Online]. Available: http://www.gartner.com/it/page.jsp?id=2237315 , November 2012.

[13] Torr, P. (2005). Demystifying the threat modeling process. *IEEE Security & Privacy Magazine - LA English, 3*(5), 66. doi: 10.1109/MSP.2005.119

[14] Roland, M., Langer, J., & Scharinger, J. (2012). Practical attack scenarios on secure element-enabled mobile devices. *Near Field Communication (NFC), 2012 4th International Workshop on,* 19-24.

[15] M. Reveilhac, M. Pasquet “Promising Secure Element Alternatives for NFC Technology"

[16] Rankl, W., & Effing, W. (2003). Smart card handbook. Ch 1.

[17] Alimi, V., & Pasquet, M. (2009). Post-distribution provisioning and personalization of a payment application on a UICC-based secure element. *Availability, Reliability and Security, 2009. ARES '09. International Conference on,* 701-705.

[18] Mantoro, T., & Milisic, A. (2010). Smart card authentication for internet applications using NFC enabled phone. *Information and Communication Technology for the Muslim World (ICT4M), 2010 International Conference on,* D13-D18.

[19] Monteiro, D. M., Rodrigues, J. J. P. C., & Lloret, J. (2012). A secure NFC application for credit transfer among mobile phones. *Computer, Information and Telecommunication Systems (CITS), 2012 International Conference on,* 1-5.

[20] Peng Kunyu, Zheng Jiande, & Yang Jing. (2009). An identity authentication system based on mobile phone token. *Network Infrastructure and Digital Content, 2009. IC-NIDC 2009. IEEE International Conference on,* 570-575.

[21] Derawi, M. O., Witte, H., McCallum, S., & Bours, P. (2012). Biometric access control using near field communication and smart phones. *Biometrics (ICB), 2012 5th IAPR International Conference on,* 490-497.

[22] Qian Tao, & Veldhuis, R. (2010). Biometric authentication system on mobile personal devices.*Instrumentation and Measurement, IEEE Transactions on, 59*(4), 763-773.

[23] Jeon, W., Kim, J., Lee, Y., & Won, D. (2011). In Smith M., Salvendy G.(Eds.), *A practical analysis of smartphone security* Springer Berlin / Heidelberg. doi: 10.1007/978-3-642-21793-7_35

[24] Eldefrawy, M. H., Alghathbar, K., & Khan, M. K. (2011). OTP-based two-factor authentication using mobile phones. *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on,* 327-331.

[25] Damgård, I. - Lecture notes 1-6 for the course dSik on Aarhus university Computer Science, may 4. 2010.

[26] GlobalPlatorm, GlobalPlatform Card Specification Version 2.2.1", 2011.

[27] Balfanz, D., & Felten, E. W. (1999). Hand-held computers can be better smart cards.

[28] Francis, L., Hancke, G., Mayes, K., & Markantonakis, K. (2010). A security framework model with communication protocol translator interface for enhancing NFC transactions. *Telecommunications (AICT), 2010 Sixth Advanced International Conference on,* 452-461.

[29] Wei-Dar Chen, Mayes, K. E., Yuan-Hung Lien, & Jung-Hui Chiu. (2011). NFC mobile payment with citizen digital certificate. *Next Generation Information Technology (ICNIT), 2011 the 2nd International Conference on,* 120-126.

[30] Mutlugün, M., & Adalier, O. (2009). Turkish national electronic identity card

[31] Private communication, Bjarke Parner, Søren Harbo Jensen.

[32] ISO/IEC 7816-3 "Identification Cards - Integrated circuit cards - Electrical interface and transmission protocols"

[33] ISO/IEC 14443 "Identification cards - Contactless integrated circuit cards - Proximity cards".

[34] ISO/IEC 18092 (ECMA 340) "Information technology - Telecommunications and information exchange between systems - Near Field Communication - Interface and Protocol (NFCIP-1)".

[35] Cranor, L. F., & Garfinkel, S. (2005). *Security and usability : Designing secure systems that people can use*. New York, N.Y.: O'Reilly.

[36]NFC World “NFC room keys find favor with hotel guests”. [Online]. Available:http://www.nfcworld.com/2011/06/08/37869/nfc-room-keys-find-favour-with-hotel-guests, November 2012.

[37] NFC World “Emirates government begins NFC national id project”. [Online]. http://www.nfcworld.com/2012/04/11/315015/emirates-government-begins-nfc-national-id-project/ November 2012.

[38] Google “Google Wallet”. [Online]. Available: http://www.google.com/wallet/, November 2012

[39] S.L. Garfinkel, A. Juels, & R. Pappu. (2005). RFID privacy: An overview of problems and proposed solutions. *IEEE Security & Privacy - LA English, 3*(3), 34. doi: 10.1109/MSP.2005.78

[40] W. Chen, G.P. Hancke, K.E. Mayes, Y. Lien, & J. Chiu. (2010). NFC Mobile Transactions and Authentication Based on GSM Network. *Near Field Communication (NFC), Second International Workshop on,*. doi: 10.1109/NFC.2010.15

[41] Klaus Finkenzeller, & Dörte Müller. (2010). *RFID handbook* John Wiley & Sons Inc - LA English.

[42] Töyssy, S., & Helenius, M. (2006). *About malicious software in smartphones* Springer Paris. doi: 10.1007/s11416-006-0022-0

[43] Hsu-Chen Cheng, Wen-Wei Liao, Tian-Yow Chi, & Siao-Yun Wei. (2011). A secure and practical key management mechanism for NFC read-write mode. *Advanced Communication Technology (ICACT), 2011 13th International Conference on,* 1095-1011.

[44] Jensen, M., Schwenk, J., Gruschka, N., & Iacono, L. L. (2009). On technical security issues in cloud computing. *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on,* 109-116.

[45] Abd Allah, M. (2011). Strengths and weaknesses of near field communication (NFC) technology. *Global Journal of Computer Science and Technology, 11*(3)

[46] Haselsteiner, E., & Breitfuß, K. (2006). Security in near field communication (NFC).

[47] Madlmayr, G., Langer, J., & Scharinger, J. (2008). Managing an NFC ecosystem. *Mobile Business, 2008. ICMB '08. 7th International Conference on,* 95-101.

[48] NFC Forum “NFC Data Exchange Format (NDEF) 1.0” 2006-07-24

[49] NFC Forum “Simple NDEF Exchange Protocol 1.0” 2011-08-31

[50] NFC Forum “Logical Link Control Protocol 1.1” 2011-06-20

[51] Tan, W. H. (2009). Practical attacks on the mifare classic.

[52] Gold, S. (2011). Cracking wireless networks. *Network Security, 2011*(11), 14-18. doi: 10.1016/S1353-4858(11)70120-9

[53] Eronen, P., & Tschofenig, H. (2005), Pre-shared key ciphersuites for transport layer security (TLS).

[54] Rescorla, E. (2001). *SSL and TLS; designing and building secure systems*

[55] Boyd, C. A., & Mathuria, A. (2003). *Protocols for authentication and key establishment*. New York, N.Y.: Springer.

[56] Aarhus Universitet - Nøgletal. [Online] http://www.au.dk/om/profil/nogletal/ , November 2012

[57] Francis, L., Hancke, G., Mayes, K., & Markantonakis, K. (2010). In Ors Yalcin S. (Ed.),*Practical NFC peer-to-peer relay attack using mobile phones* Springer Berlin / Heidelberg. doi: 10.1007/978-3-642-16822-2_4

[58] Wang, X., Yin, Y., & Yu, H. (2005). In Shoup V. (Ed.), *Finding collisions in the full SHA-1* Springer Berlin / Heidelberg. doi: 10.1007/11535218_2

[59] Euromonitor International. Assessing Market Potential of NFC for Mobile Payments in 2012: From Swiping to Waving. Global Briefing 2011.

[60] GSM Association. Pay-Buy-Mobile. Business Opportunity Analysis. GSM Association, November 2007.

[61] Bluetooth, Bluetooth Specification Version 4.0, Architecture & Terminology Overview, Vol. 1 pages. 17-23.

[62] Phan, R. C. -., & Mingard, P. (2010 %TS An InternationalJournal). Analyzing the secure simple pairing in bluetooth v4.0. Wireless Personal Communications, 64(4), 719. doi: 10.1007/s11277-010-0215-1

[63] Sandhya, S., & Devi, K. A. S. (2012). Analysis of bluetooth threats and v4.0 security features. Computing, Communication and Applications (ICCCA), 2012 International Conference on, 1-4.

[64] NFC Basics [Online]. Available: http://developer.android.com/guide/topics/connectivity/nfc/nfc.html#p2p, December 2012

[65] Bluetooth, Bluetooth Specification version 2.1 + EDR, Simple Secure Pairing, Vol. 1 pages. 57-61.

Rawad Kilani and Kenneth Jensen, Mobile Authentication with NFC enabled Smartphones, 2013

Department of Engineering
Aarhus University
Edison, Finlandsgade 22
8200 Aarhus N
Denmark

Tel.: +45 4189 3000