D1.2

Report on MATTHEW Platform Specifications

Project number:	610436
Project acronym:	MATTHEW
Project title:	MATTHEW: Multi-entity-security using active Transmission Technology for improved Handling of Exportable security credentials Without privacy restrictions
Start date of the project:	1st November, 2013
Duration:	36 months
Programme:	FP7-ICT-2013-10

Deliverable type:	Report
Deliverable reference number:	ICT-610436 / D1.2/ 1.0
Activity and Work package contributing to the deliverable:	WP 1
Due date:	Aug 2014 – M10
Actual submission date:	Aug. 29th, 2014

Responsible organisation:	IFX
Editor:	IFX – Frank Scherber
Dissemination level:	Public
Revision:	1.0

Abstract:	This document describes the MATTHEW platform architecture with its components and interfaces highlighting the sub-components on which research and implementation will be focused during the following periods in the project. It includes the description of the MATTHEW demonstrators and how they are derived and tailored from the overall architecture.
Keywords:	MATTHEW, architecture, components, interfaces, demonstrators

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement n° 610436.

Editor

Frank Scherber (IFAT)

Contributors (ordered according to beneficiary numbers)

Holger Bock, Martin Buchsbaum, Frank Scherber (IFAT)

Gregory Capomaggio (GTO)

Giuliano Manzi (AMS)

Erich Wenger (IAIK)

Martin Deutschmann, Michael Höberl, Sandra Lattacher (TEC)

Pavel Kristof, Jakub Trnavský (IMA)

Pascal Paillier (CRX)

Executive Summary

This deliverable D1.2 "Report on MATTHEW platform specifications" is the second technical deliverable of the MATTHEW project and in the same time the final deliverable of work package WP1. It builds the basis for milestone MS1 "Use case and system architecture specified", due in month M10 of the project run time, which is end of August, 2014, and thus finalizes activities in WP1 "System Requirements, Architecture and Specification".

The MATTHEW consortium researched and developed the overall architecture of the MATTHEW platform and describes it in terms of its structural components and interfaces keeping in mind the interaction of those components, not only amongst themselves, but also towards the MATTHEW system environment with its background services. The project partners chose a role-based approach to cluster components of the architecture regarding the user role domain and the verifier role domain and point out which of those architectural components have to be further researched, developed and implemented within the following phases and work packages of the MATTHEW project. These hardware and software components are described in a more detailed granularity within this report, whereas other platform components that may be integrated as state-of-the-art instances are described with less detail. Finally, reference is given, how the MATTHEW architecture will be represented in the various use case demonstrators and which components of the architecture are needed for the different use cases.

This overall MATTHEW architecture builds the framework for research being performed in WP2 "Multi Entity Security" as well as for implementation of hardware and software in WP3 "Hardware Component development" and WP4 "Application development", respectively.

Contents

List of Figures

Chapter 1 Introduction

Mobile devices incorporating hardware-based security anchors have been around in various shapes and in several different system contexts may it be as user identification like with SIM cards in mobile phone applications or as secure elements (SE) for virtual wallet applications.

The ecosystem and MATTHEW system environment (see Figure 1 for an abstraction) in which research shall be performed is of course not only consisting of the mobile devices themselves and their secure elements, but consists also of the mobile networks, the internet, remote server applications like Trusted Service Manager, and potentially cloud based services.

Describing the architecture for MATTHEW could thus easily end up as yet-another-description of mobile networked devices without specific relevance for the work to be performed in the project. Also the selected use cases have very different character. The research in the area of mobile offline payment transactions is focusing on other specific problems than the one setting up a multi-user access control scheme. Still those two scenarios are more mature with respect to their technology readiness level than our third use case with transfer of rights and credentials in a privacy preserving context for ticketing applications. But nevertheless the three use cases have many things in common when it is about describing the ICT-architecture.

On the other hand MATTHEW is a STREP project and thus shall clearly focus on some dedicated, well specified research topics, but still the MATTHEW architecture shall represent a valid framework for all use cases from which requirements were collected and depicted in D1.1 – Report on use case and system architecture requirements. Thus it is a clear goal to formulate this architecture description in a way that it represents all components needed for the implementation of the use case relevant demonstrators while at the same time specifying clearly on which components or sub-components of the MATTHEW architecture the research is concentrated.

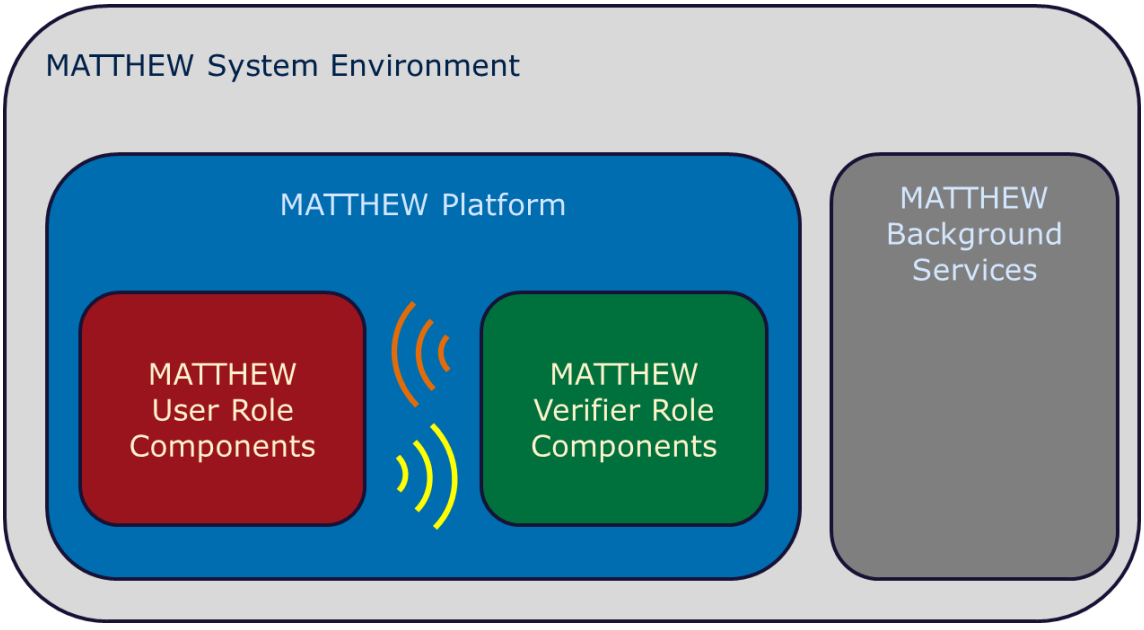

Figure 1: MATTHEW system architecture top level abstraction

To tackle this challenge the consortium decided to approach the architecture document starting from a role-based description: In chapter 2 we modularize our system description into what we call the MATTHEW platform – dealing with all components needed in the end user environment – and the area of background services.

The MATTHEW platform area is then further divided into the “User Role” part, usually consisting of one or more mobile devices with one or more secure elements and the “Verifier Role” part on the other side, which includes a mobile or fixed terminal with NFC communications means and the cryptographic counterpart – again potentially a secure element – deciding about success or failure of the cryptographic operation and protocol being performed.

In chapter 3 we describe the components worked on during the implementation phase of the MATTHEW project, with dedicated sub-chapters for the hardware of secure entities and a special focus chapter on novel RF transmission components and integration concepts. A chapter on software components describes the various SW layers and interfaces to users.

Finally, in chapter 4 we provide more details about the use cases and planned demonstrators and in which way they employ the framework of the MATTHEW system architecture.

Chapter 2 System Overview

The overall MATTHEW system architecture has been already previously depicted in section 3 of the D1.1 – Report on use case and system architecture requirements. While that very chapter in D1.1 was focussing on the architecture requirements we are now describing the structural break down of the MATTHEW system with respect to its components and their interfaces and give an indication on their functional behaviour.

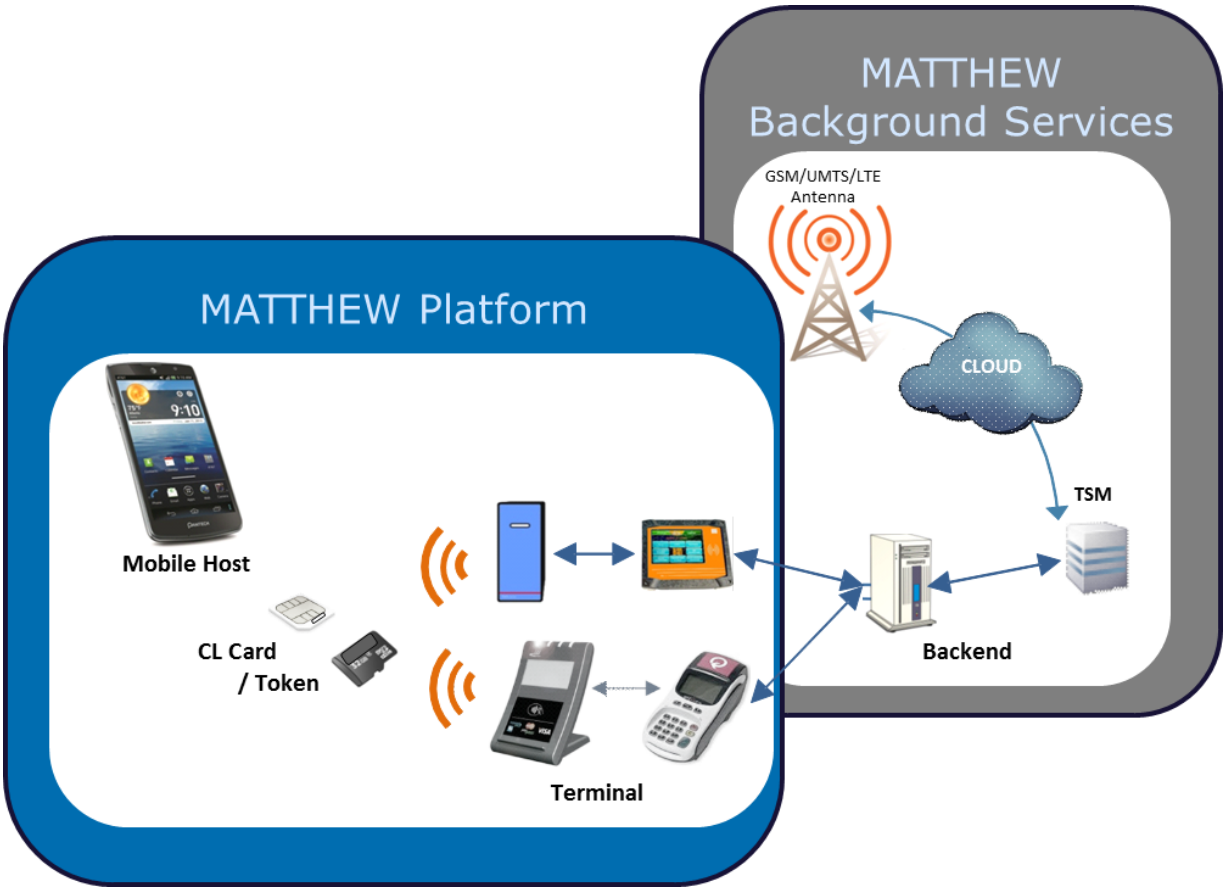

Figure 2: MATTHEW system architecture

2.1 The MATTHEW Platform

As depicted above we defined the MATTHEW platform as that part of the system that is present in the user environment and usually consists of those components that are under control of the user, like the mobile devices and their secure elements as well as the components with which the user wants to interact. One special but essential feature in our MATTHEW platform is the fact, that active transmission technology shall be used in the small form factors to achieve best communication performance despite the high degree of miniaturization.

The following picture summarizes all components of the MATTHEW platform needed for the user and verifier roles (components that are researched within the MATTHEW project are underlined).

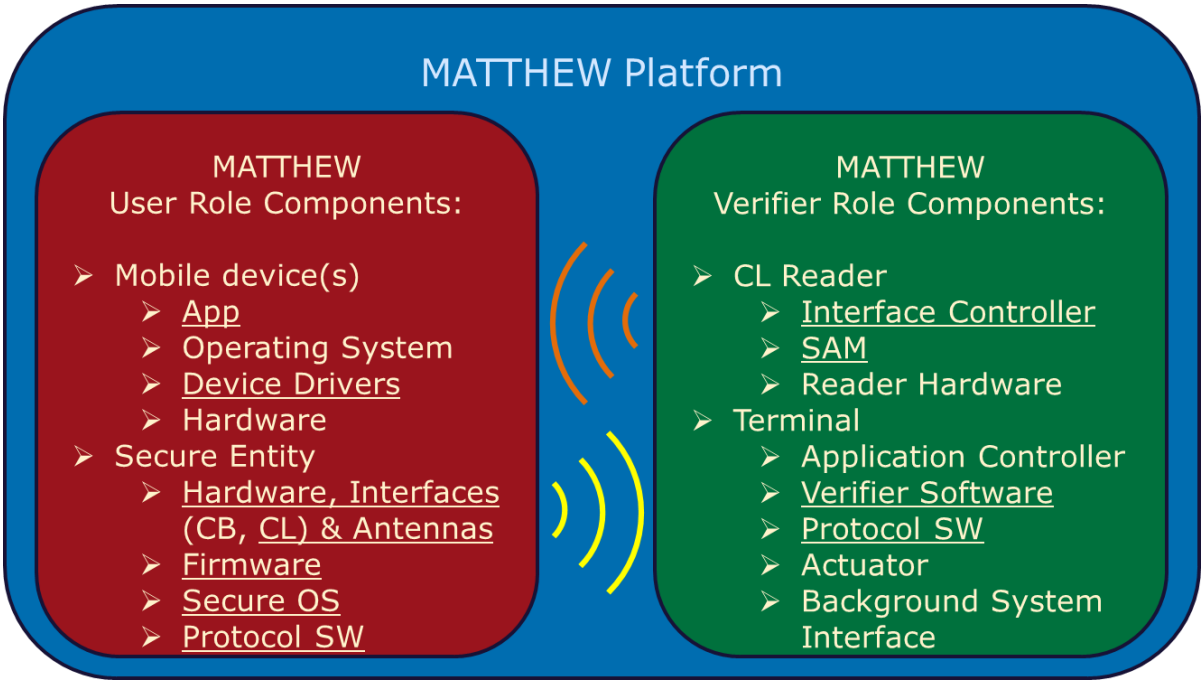

Figure 3: MATTHEW platform role and components using Active Transmission Technology

2.1.1 *"User" Role Domain*

In the context of MATTHEW the user role is represented by the user herself and by several components that allow the user to fulfil all necessary steps to perform the protocol needed to achieve what she wants, may that be performing a contactless payment transaction, access to an area with restricted access policy or transfer of credentials representing a transferable ticket.

Thus the MATTHEW architecture components that are part of this user role domain are the Secure Entity (microSD, mini- or nano-SIM, or embedded Secure Element) and the mobile device itself, may it be a smartphone or a tablet.

- "User" = Secure Entity + Mobile device

Also part of the user role domain is the active contactless communication interface with focus on novel robust integration concepts for improved manufacturability

2.1.2 *"Verifier" Role Domain*

The second domain we are focussing on in the MATTHEW project is the verifier side. On the one hand the contactless reader is in charge of initiating and controlling the contactless communication with the secure entities in the mobile device. On the other hand its interface controller has to handle the (Security Access Module) SAM interaction and to ensure the connection to the terminal, either a Point of Interaction (POI) terminal or an access control terminal, with their respective verifier software components.

- "Verifier" = CL Reader (with optional SAM) + Terminal (with verifier software)

User domain and verifier domain are connected via active contactless communication links in the MATTHEW platform

2.1.3 Background Services

In the area of the background services MATTHEW is relying as much as possible on existing and standardized components (like the K4 system or EMVCo, respectively). Nevertheless – especially in the area of transferable user rights with privacy protection – the overall protocol framework including components belonging to the area of background services has to be taken into account. Thus additional roles, like issuer and inspector have to be co-evaluated during the implementation phase of the project.

Chapter 3 Component Description

3.1 Security Hardware Components

Most of the security hardware components are part of the MATTHEW "User" role domain and are of special interest for the MATTHEW project, on the one hand as security anchors and on the other hand with respect to improved contactless NFC communication. One exception to this is the SAM, which may reside in the "Verifier" role domain in the contactless reader or terminal to protect secret keys in case of protocols using symmetric crypto primitives, like the AES-based CIPURSE protocol.

3.1.1 Secure Entities

A Secure Element (SE) is a tamper-resistant platform (typically a one chip secure microcontroller) capable of securely hosting applications and their confidential and cryptographic data (e.g. key management) in accordance with the rules and security requirements set forth by well-identified trusted authorities.

There are different form factors of SE: Universal Integrated Circuit Card (UICC), embedded SE and microSD. Both the UICC and microSD are removable. Each form factor links to a different business implementation and meets a different market requirement.

3.1.1.1 Micro Secure Digital Card – microSD

The micro Secure Digital (SD) is the smallest non-volatile memory card format, standardized by the SD Association, for use in portable devices such as mobile phones, digital cameras, GPS navigation devices, handheld consoles, and tablet computers. The micro SD card is actually the most common removable memory extension supported by such devices.

In 2010 the SD Association introduced the Advanced Security SD (ASSD) specification that defines a standard for accessing a secure element, embedded into an SD memory card and able to process cryptographic operations or to execute a full secured application. This specification offers to third parties a possibility to deploy their own secure solutions through the end user's mobile equipment by just providing a micro SD card.

During the last years many efforts have been made to provide contactless communication capability to micro SD cards. Unfortunately, an "all-in-one" contactless card architecture never emerged due to poor RF performances owed to the tiny size of the embedded antenna.

Furthermore, contactless communication between the embedded secure element and a remote RF coupler is possible by using the additional SWP pin, introduced by the SD Association, which allows for a microSD memory card to take advantage of an NFC-enabled host. Nevertheless, this solution is currently limited to a few pilots or dedicated small infrastructures in Asia due to the low deployment of mobile devices supporting SWP-compliant microSD cards.

Figure 4: SWP microSD memory card solution

3.1.1.2 Universal Integrated Circuit Card – UICC

Universal Integrated Circuit Cards (UICC) being the physical representation of subscriber identification modules (SIM) exist in 4 different form factors, the two most recently defined ones being the so called microSIM (2003) and nanoSIM (2012). Still widely used are also miniSIM UICCs. The figure below shows the concept of an active transmission system implemented into a miniSIM design. The miniSIM consists of a Secure Element, an active transmission based so called Active Communication Frontend (ACF) and a small antenna designed in a way that contactless communication is compliant with ISO/IEC 14443. The contact based communication protocol is implemented according to ISO/IEC 7816.

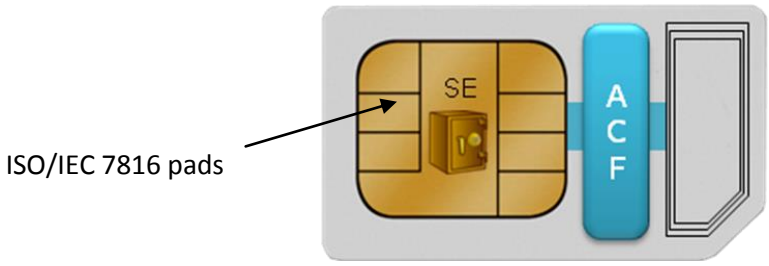

Figure 5: miniSIM with active transmission

A nanoSIM design is much smaller (12.3 x 8.8 mm² versus 25.0 x 15.0 mm²) and needs much more effort to achieve enough RF performance for active transmission. Special antenna approaches are

necessary to achieve a miniaturized component design that is able to fulfill the requirements on modulation strength from card to reader.

For integration into this form factor some well-known and established technologies like flexible PCBs (printed circuit board) and wire bonding for the connection of the ICs to the substrate will be used. Focus is to develop first an antenna component including ferrite material and a special construction of antenna loops. This component will be placed on a flexible PCB next to several passive SMD components for matching and the two wire-bonded ICs (AFC, SE). Finally the flexible board will be molded into a nanoSIM package.

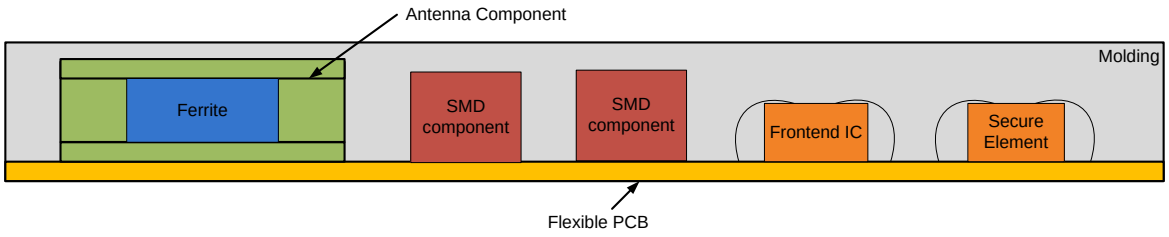

Figure 6: Stackup of nanoSIM prototype

3.1.1.3 Embedded Secure Element – eSE

An embedded Secure Element (eSE) is a hardware component with strong security functions which can be soldered directly to the printed circuit board of a mobile device. Since the eSE is physically connected to the end user's device it is supposed to be the type of secure element which is under direct control of the end user. Thus such a SE is a good candidate to store privacy relevant credentials. Even if the protocols researched in WP2 are not limited to certain form factors we assume that the user role component fitting best for the ticketing use case scenario with transfer of credentials is the one fixed to the mobile device, the embedded Secure Element.

3.1.1.4 Security Access Module – SAM

A Security Access Module (SAM) is a removable security processor which is performing verifier role functions for the validation of responses sent from secure elements during execution of security protocols. All security calculations on the reader side are calculated inside the SAM. All keys which are needed for the calculations are stored in special security regions of the SAM. The shape of a SAM is the same as of a standard SIM and inside the readers are standard SIM connectors implemented. Advantage of using a SAM as security processor is, that producers of readers do not need to know keys and all security algorithms which are necessary for the authentication process between the secure entities incorporating the user role and the reader.

SAM functionality in reader software

The simplest way for SAM functionality inside a reader is to use special software modules for the calculation of cryptograms and ciphering inside the reader. Advantage of this solution is that the reader is compact, no SAM connector is needed and the reader can be easily made resistant to rough weather conditions. Disadvantage of this solution is that for cryptogram calculation the same processor is used as for the reader and computing power of the reader processor may be insufficient. One more disadvantage is that keys and cryptographic algorithms must be known by the reader

manufacturer. Due to these disadvantages we will for the MATTHEW project use a SAM in standard SIM shape and protect the reader as a whole against rough weather conditions by other means.

Hardware SAM in JCOP card

Besides EMVCo being used for the payment use case, the CIPURSE standard from OSPT alliance has been selected as the security standard for the access control use case in the MATTHEW project. The alliance has announced to provide the SAM. But up to now there is only a paper specification available and no real SAM hardware. This specification implies what type of standard programmable card can be used for the creation of CIPURSE SAM cards. The card must provide a secure array for key storage and the card must have a security coprocessor supporting the calculation of DES, AES and HASH cryptograms. Therefore we will use Java Card with JCOP operating system.

Java Card technology was originally developed for the purpose of securing sensitive information stored on smart cards. Security is determined by various aspects of this technology:

Data encapsulation

Data is stored within the application, and Java Card applications are executed in an isolated environment (the Java Card VM), separate from the underlying operating system and hardware.

Applet Firewall

Unlike other Java VMs, a Java Card VM usually manages several applications, each one controlling sensitive data. Different applications are therefore separated from each other by an applet firewall which restricts and checks access of data elements of one applet to another.

Cryptography

Commonly used symmetric key algorithms like DES, Triple DES, AES and asymmetric key algorithms such as RSA are supported as well as other cryptographic services like signing, key generation and key exchange.

Applet

The applet is a state machine which processes only incoming command requests and responds by sending data or response status words back to the interface device.

The main problem is that JCOP cards do not fully support all necessary algorithms. For the CIPURSE scheme these algorithms must be written in JAVA code and can therefore be very slow. We tested it on the reader and reading time for authentication was about 2s which is a rather long time for an identification application.

Hardware SAM in security processor card

The best solution is to develop a native application SAM on a smart card hardware which supports all functions and cryptographic algorithms needed for the CIPURSE scheme. Here is a basic specification for the CIPURSE SAM requirements:

- It employs existing, proven open standards, including the ISO 7816 smart card standard, as well as the 128-bit AES. Designed for low-cost silicon implementations, the CIPURSE security concept uses an authentication scheme that is resistant to most of today's electronic attacks.
- Its security mechanisms include a unique cryptographic protocol for fast and efficient implementations with robust, inherent protection against Differential Power Analysis (DPA) and Differential Fault analysis attacks. Because the protocol is inherently resistant to these

kinds of attacks and does not require dedicated hardware measures, it should be both more secure and less costly. It is intended to guard against counterfeiting, cloning, eavesdropping, man-in-the middle attacks and other security threats.

3.1.2 Security Controller

One core component of the secure entities in the user role domain is the security controller with dedicated hardware security features to protect critical data like keys especially while performing cryptographic operations. In case of privacy enhanced protocols being performed it is often desirable that not only the security relevant, but also the privacy relevant protocol steps and operations are performed inside such a protected controller. An example block diagram is given below.

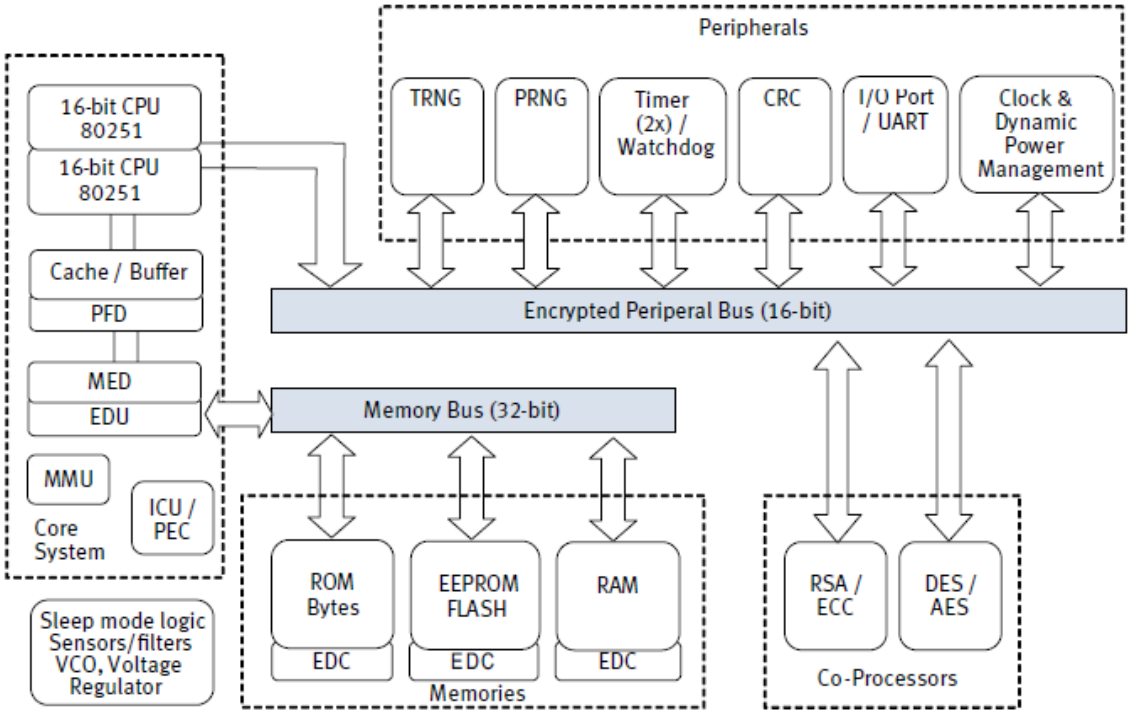

Figure 7: Top level architecture of a security controller

The Security Controller itself consists of four main blocks being connected via different bus systems.

- CPU block
- Memory block
- Crypto co-processor block
- Peripheral block

In the CPU area a dedicated dual core is implemented to allow for error checking against run-time fault attacks. A memory encryption block provides a fast cipher to encrypt data being stored to EEPROM/FLASH or RAM and to decipher code or data on its way from the memory block to the CPU block, coming from ROM, RAM or EEPROM/FLASH.

The memory block provides secured storage for code and data, being divided into a ROM area, which contains static configuration data, EEPROM/FLASH for code, personalized non-volatile data and keys, as well as RAM for intermediate results of calculations.

The crypto co-processor block is connected to the CPU block via an encrypted peripheral bus and contains hardware engines for symmetric (DES – for legacy applications – and AES) as well as for asymmetric (RSA, ECC) cryptographic algorithms.

The peripheral block contains on the one hand the random number generators which are needed for key generation and session key establishment as well as for signature algorithms and on the other hand the interfaces which are described in more detail in the following section.

3.1.3 Communication Interfaces

The current concept for security protocol implementations based on microSD and nanoSIM is making use of three main interfaces:

- Contact-based communication interface according to ISO/IEC 7816 from the secure element to the baseband IC of the mobile phone. In microSD implementations this communication will be tunnelled through the flash controller.
- For both of the form factors, microSD as well as nanoSIM, the active transmission frontend will be connected via ACLB (Analog Contactless Bridge) to the secure element. The bridge carries transparent signals based on the inputs at the contactless interface according to ISO/IEC 14443. In other words, the frontend amplifies the small signals at the small antenna of the contactless interface to a transparent signal called ACLB.
- For configuration of the active transmission frontend via the flash controller in microSD implementations an additional SPI interface can be used. In nanoSIM implementations (where no flash controller is available) a dedicated mechanism within the ALCB interface has to handle this.

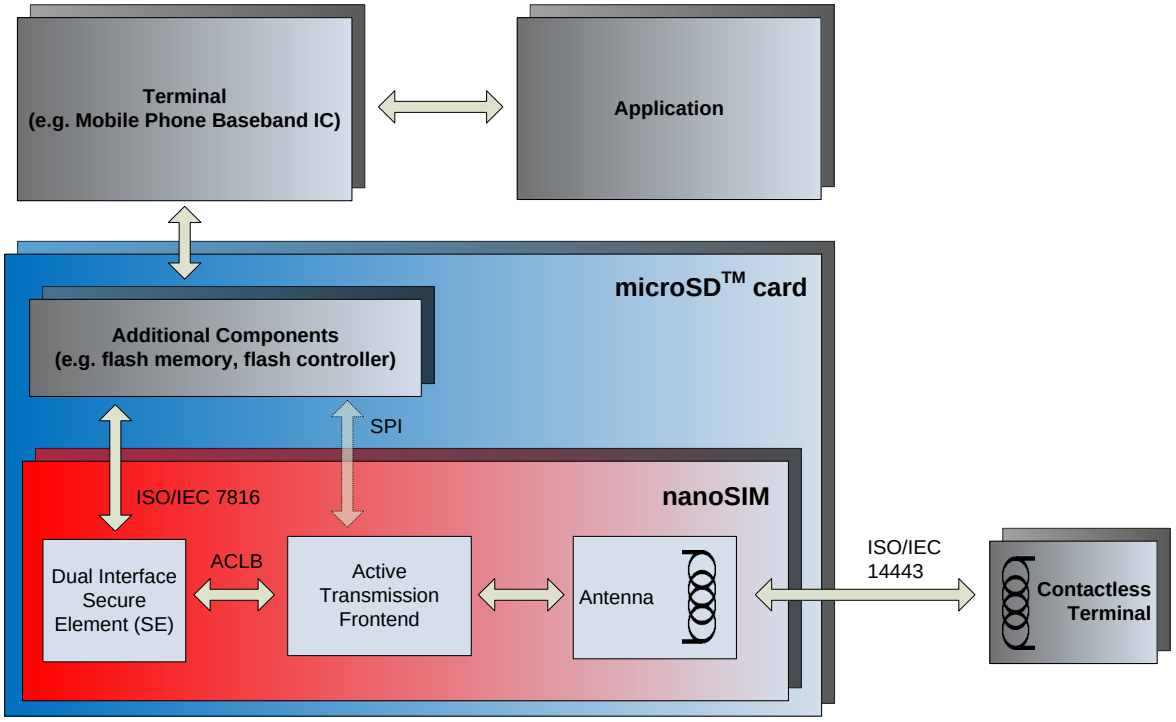

Figure 8: Concept for active transmission in microSD or nanoSIM

3.1.4 Physically Unclonable Functions

In the MATTHEW context Physically Unclonable Functions (PUFs) appear as an optional component in the user role domain to allow for binding of data (like keys) or code of software to a dedicated instance of an electronic IC and thus prevent credentials from unauthorized use of privileges connected to such credentials.

A *Physically Unclonable Function* is a function that maps a set of challenges to a set of responses based on an intractably complex physical system [3]. It is a physical object which can be seen as a static source of randomness [1], which produces unpredictable but as such reproducible and instantiation-dependent output values [2]. Since the hardware characteristics are sensitive to environmental variations, the output behaviour of these functions is in general supposed to be inconstant over environmental parameters like temperature or supply voltage. For micro- or nano-electronic implementations of PUFs electric noise is potentially influencing the responses. Each re-read output value will be slightly different to the previous ones even when the module is always stimulated with exactly the same input. Thus, due to the physical nature of their implementations, PUF responses are generally neither perfectly reproducible nor perfectly random. [2] To minimize these sources of imperfectness either dedicated design and implementation options or mathematical post processing – or both – are needed.

A PUF is called intrinsic if it is embedded on an IC and if the basic building blocks are regular digital primitives for the chosen manufacturing technology [6]. Some intrinsic realizations of PUFs are described in the following:

Arbiter PUFs belong to the group of delay-based PUFs; one input signal for each of two signal paths is provided at the same time at the front-end, while the arbiter decides at the backend on which paths the signal arrives first. The difference of the arrival times is based on variation in manufacturing and is uniquely defined for each PUF.

SRAM PUFs: an SRAM cell is mainly comprised of two inverters, which are not perfectly the same due to randomly distributed manufacturing variations. If it is powered up, it falls into one of the two possible binary states; since this kind of PUF produces a binary string and does not need any quantization processes, it is highly practicable.

Other memory based PUFs are building on the principle of SRAM PUFs, but use other digital components than inverters: **Butterfly PUFs** (butterfly-cell), **Latch PUFs** (NOR gates), **Flip Flop PUFs** (flip flop cell).

The unreliability is reflected by the hamming distance which can be expressed as the expected bit error rate between two responses ω and ω' which result out of the same challenge C:

$$dis(\omega, \omega')$$

There are two applications PUFs are mainly utilized for. Firstly, PUFs prove to be a suitable candidate to develop secure authentication protocols, however this is not in the scope of MATTHEW. Further PUFs can be used to build secure key generators. The concept utilizes PUFs to generate a secret key, based on the intrinsic HW parameters. The concept comprehends two phases, namely one time enrolment (in secure environment) and reconstruction (in the field). The approach comes along with significant advantages regarding security:

- The key is not present when device is powered down
- The key is only valid for the given device, depending on identifiable intrinsic randomness

Normally PUF instantiations are part of the design of ICs and therefore already considered before manufacturing of the device. Sometimes however the need arises to secure existing hardware elements, which do not necessarily include a PUF already. Some research has therefore been carried out towards the usage of existing memory cells to "build" a PUF. A lot of electronic devices need memory to store data temporally. Therefore it is sometimes possible to utilize these cells as a PUF. Evaluating these cells is crucial during the design process to gain knowledge about their error rate, their distribution and to be sure that these cells are not initialized by an operating system. Within MATTHEW we will follow this path and plan to include some results in the course of the work in WP2.

PUF unit structure and Interfacing

If an embedded system includes a PUF unit, two modules – key enrolling/generator and key reconstruction – and a command interface are necessary to integrate the PUF.

These modules can be addressed by the application designer through a high level command interface to create, store or load a key on-the-fly without the need of understanding the physical characteristics of the dedicated implementation.

A high level workflow using the modules via the command interface may look as follows:

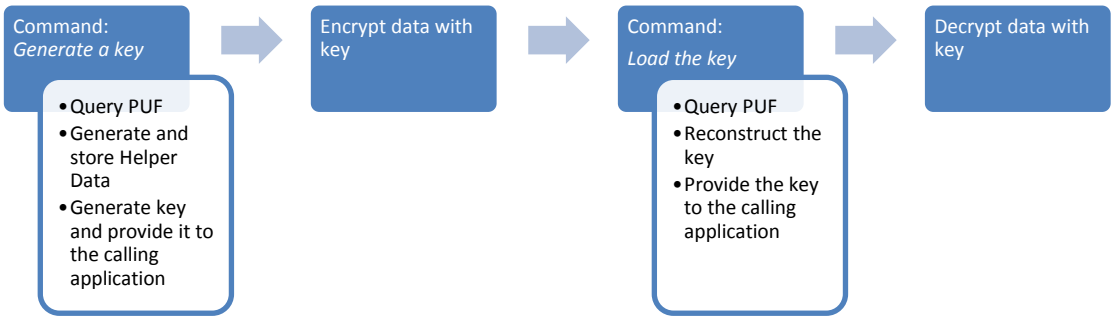

Figure 9: High level command interface for integrating a PUF into an embedded system

The following scenario describes the possible integration of the previously mentioned modules into an application or protocol: During the first start-up of the application a key is generated with the help of the module for generating a key. This key is then used to encrypt certain sensitive data. As soon as this data has to be decrypted the key can be reconstructed with the second module and it is possible to decrypt the data. If the key is erased from the encryption module's internal memory it is not present in the system component anymore. However, if the key is needed again, it can be reconstructed with one single call of the according module.

With respect to integration in the hardware architecture of a security controller the PUF modules may be connected to the other system components via an encrypted peripheral bus. In the following figure such an integration of the PUF modules into the security controller top level architecture is depicted.

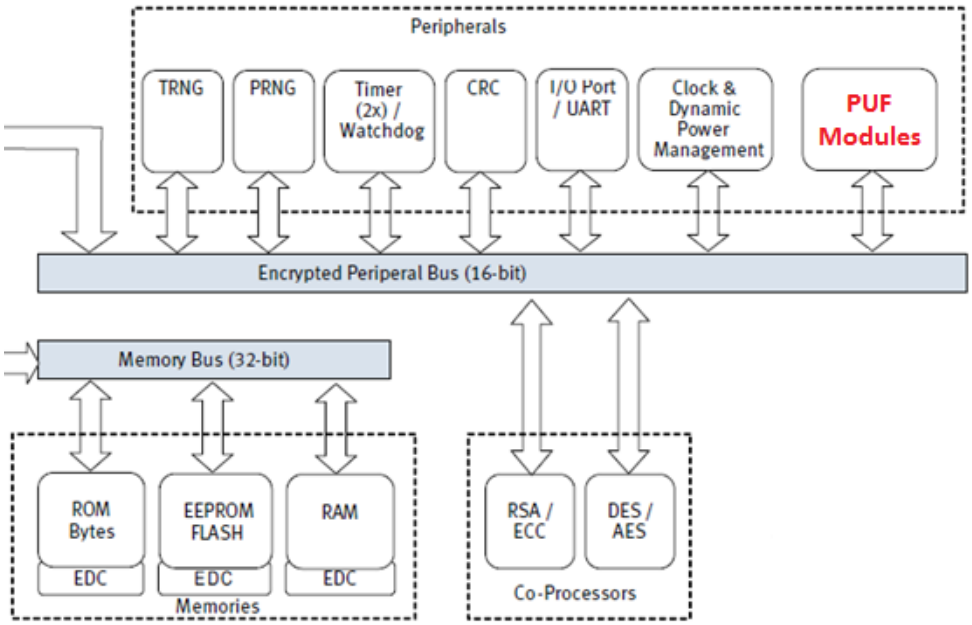

Figure 10: Integration of PUF modules into a security controller

Helper Data Algorithms

Since PUF responses are noisy functions and non-uniformly distributed, they cannot directly be used as secret keys within cryptographic proceedings. Therefore additional Helper Data Algorithms (HDA) need to be implemented, taking care of essentially two key requirements, reproducibility (failure rates of about 10^{-6} are considered) and uniform distribution (keys need to have maximum entropy).

An important construct when designing helper data algorithms is the fuzzy extractor, which is to some extend a concrete instantiation of an HDA. A fuzzy extractor is a mechanism based on error correcting codes to generate reliable bit strings out of noisy PUF responses. In other words, fuzzy extractors allow for extraction of some randomness R from w and then successfully reproduce R from any string w' that is close to w [4].

Basically two construction variants are described in literature, the code-offset and the syndrome construction. Both methods employ a binary [n,k,t] block code C with error correction capability t.

Code Offset Construction

The code-offset construction generates helper data, by summing up the response and a code word c. The helper data is later used to correct the noisy response during the reconstruction. The helper data is considered non-sensitive and stored either on the device (NVM) or externally. The key is defined as the hashed response. Alternatively the response may be used as the mask and code word as key.

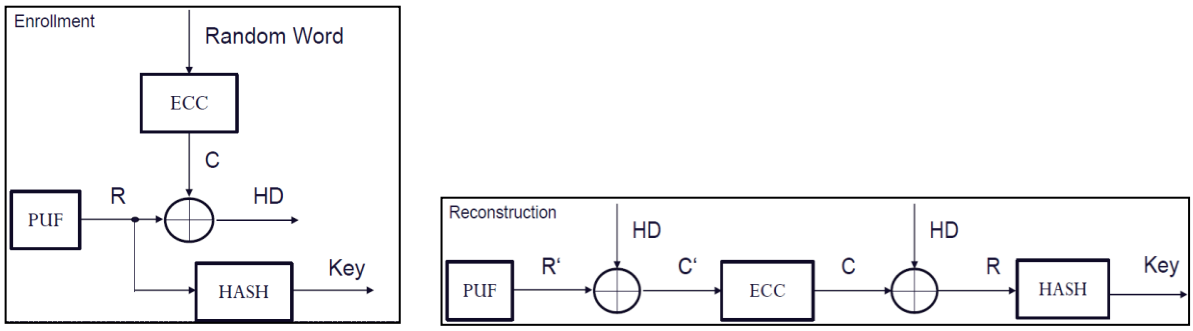

Figure 11: Sub-components for key enrolment and reconstruction by means of a code offset construction

Syndrome Construction

The syndrome construction requires a linear block code, as it employs the parity check matrix H. The enrolment starts with multiplying the response with the parity check matrix. The result is the helper data which is again stored in some kind of NVM. The helper data is added to the product of R' multiplied by H^T during reconstruction. The resulting syndrome is decoded to receive the error vector, which needs to be again added to the noisy response to receive R again. The key is defined as the hashed response.

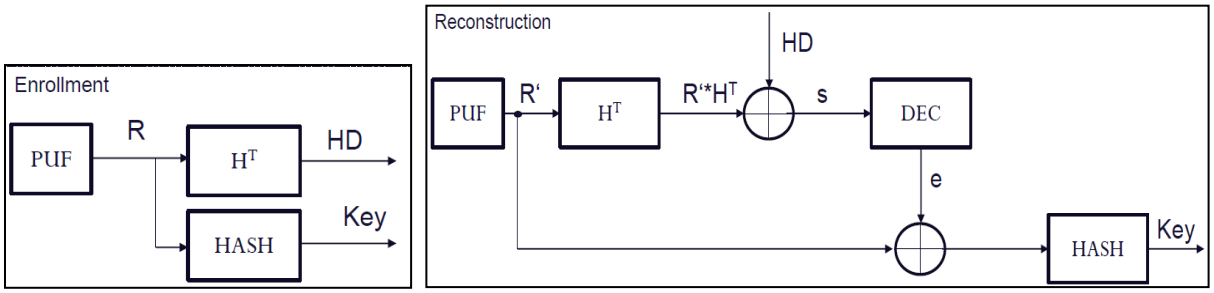

Figure 12: Sub-components for key enrolment and reconstruction by means of a syndrome construction

PUF Specifications within the MATTHEW Platform

Within MATTHEW we consider PUFs being used as a tool for the implementation of certain security functionalities. There are essentially two applications we envision:

- Key generation
  Within the ticketing use case we will need a secure group signature key. We will utilize a PUF to serve as key generator, making sure the required specifications are fulfilled.

- SW/HW binding
  The PUF can also be used to bind arbitrary information (e.g. software, sensitive data...) to a hardware device. Regarding the Ticketing use case, this means binding a ticket to a dedicated device. In this way illegitimately copying of the ticket to another device is avoided. On the other hand, it has to be worked out how the transferability property of the ticket (which is desired) can still be realized.

3.2 Novel RF Transmission Components

The RF-IC under development by ams AG (project name AS39-24/25) is an advanced NFC active tag front-end with “Active Transmission Technology” (ATT). This front-end can be connected to already existing contactless secure elements over ACLB or NFC-WI interfaces. In the MATTHEW context the AS39-24/25 is a component of the secure entity and belongs to the user role domain.

Target applications of this new RF-IC are all applications (like the ones specified in D1.1 and in general all mobile payment applications) where, due to the presence of large metal parts in proximity of the HF coil antenna, the usage of traditional systems based on passive load modulation is challenging. In those particular applications the combination of a standard secure element with an ATT-based RF front-end boost will be the perfect solution and allows payment applications through SIM, microSD and nanoSIM. Besides ATT, the new RF-IC will include two other innovative pieces of technology.

The first piece is Antenna Auto Tuning (AAT); it allows the RF-IC to adjust its coil antenna tuning.

The second is Automatic Power Control (APC) which dynamically adjusts the output power of the RF-IC and consequently controls the level of load modulation. Special attention will be put on power management in order to minimize power consumption during operation of the RF-IC and during listening mode which will make it possible to reduce the overall system’s power consumption and therefore reduce battery drain.

The AS39-24/25 will enable the development of products that are compliant to all relevant standards for HF RFID communication and contactless payment – e.g. ISO14443, FeliCa™, EMVCo.

3.2.1 Active Transmission Frontend Chip

Active boost technology generates a tag response by actively transmitting a signal which is synchronous to the reader field allowing the reader to tag communication one order of magnitude lower in terms of antenna coupling factor than with conventional passive load modulation. The AS39-24/25 will be an advanced RF-IC RFID analogue front-end that will allow the tag to produce a reply to a reader command by means of an active transmission.

Active transmission implies that the RF-IC generates a signal fully synchronous with the reader carrier. By adjusting the transmitted amplitude and modulation, the RF-IC emulates a traditional passive tag answer and is able to comply with all current RFID-related standards defined for passive load modulation systems. Together with the extraction of the reader carrier signal frequency and phase (synchronization) the AS39-24/25 will demodulate the reader to tag transmission and will make the demodulated signal available to other ICs connected to it – e.g. the secure element. In the case where a SE is connected to the AS39-24/25, the demodulated signal will be sent for example over ACLB or NFC-WI interface. The AS39-24/25 will allow the SE to transmit back to the reader by using the same interface (ACLB or NFC-WI). It will receive the data to be transmitted from the SE and will generate the analogue signals according to the required air interface protocol.

Figure 13 gives an overview over the AS39-24/25 system architecture concept. The AS39-24/25 will have on one side the interface to the SE over ACLB or NFC-WI (or eventually to a microcontroller via SPI) and on the other side the interface to the coil antenna.

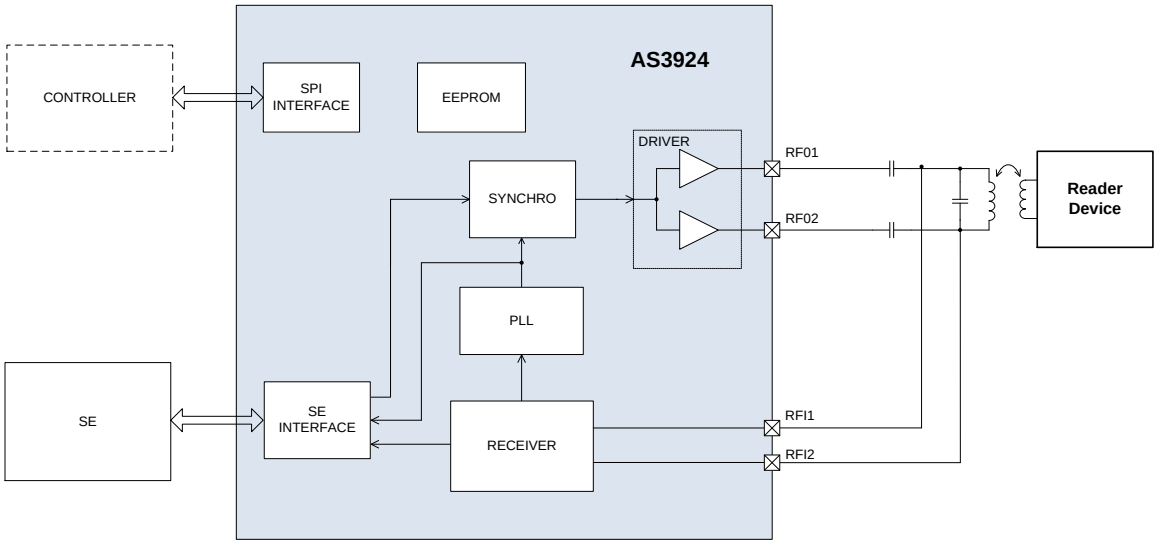

Figure 13: AS39-24/25 system architecture concept

Figure 14 shows an example of a one bit transmission according to ISO14443 type A at 106 kb/s. The top canvas (1) depicts the signal on the reader antenna. The second canvas (2) depicts the envelope of the signal on reader antenna, showing modulation while the AS39-24/25 transmitter is active. Canvas (3) depicts the current of the transmitter driver which is activated during the modulated periods of subcarrier. The bottom canvas depicts the signal on tag's antenna coil. Due to the antenna Q factor the amplitude increases while the transmitter is active (activation time is too short to reach steady state), when the transmitter is disabled the oscillation induced by the transmitter is decaying. In the second half of the bit period, there is no transmission. After the signal induced by the activation of the transmitter completely decays, the signal induced by the reader carrier signal is reinstalled on the tag antenna.

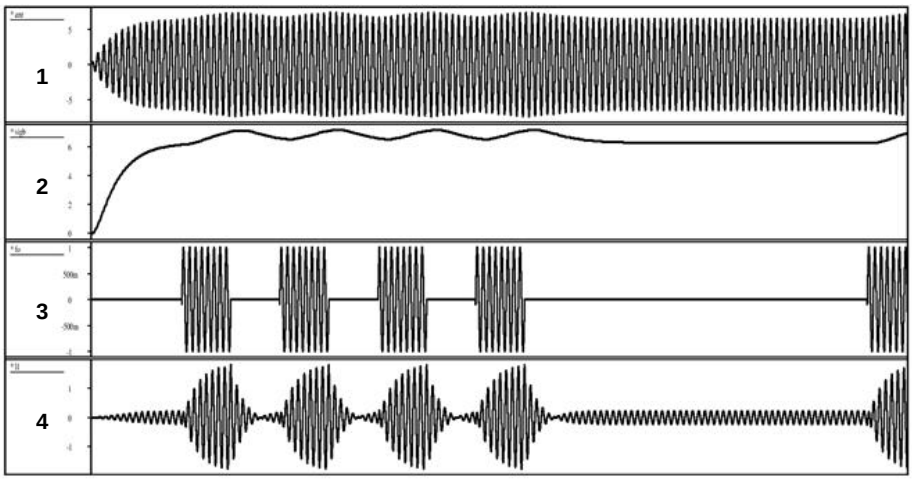

Figure 14: Example of ISO14443A 106kb/2 transmission

3.2.2 Novel Antenna Design Options

ams AG together with the MATTHEW project partners is investigating new coil antenna designs with very small form factor to be used together with AS39-24/25 in mobile payment applications.

The coil antennas under investigation are designed in a way to produce a dominant magnetic field distribution along different axis. Mobile devices have many metal parts – e.g. PCB, battery pack, LCD, shield. Using the classic planar coil antenna concept that generates a field dominantly orthogonal to the antenna plane – and therefore orthogonal to the smart card, microSD etc… plane – may lead to a weak magnetic field outside the mobile device. This is due to losses introduced by the metal surfaces and the electro-geometrical configuration. In order to overcome this problem a novel 'solenoid' antenna concept able to generate a magnetic field having coplanar dominant components is under investigation. Numerical simulations will be followed by practical experiments. It is possible that a hybrid concept (solenoid + planar) will be simulated.

An overview over the two antenna coil concepts under investigation and their magnetic field distribution (2D plot) along specific directions is presented in Figure 15 and Figure 16, respectively.

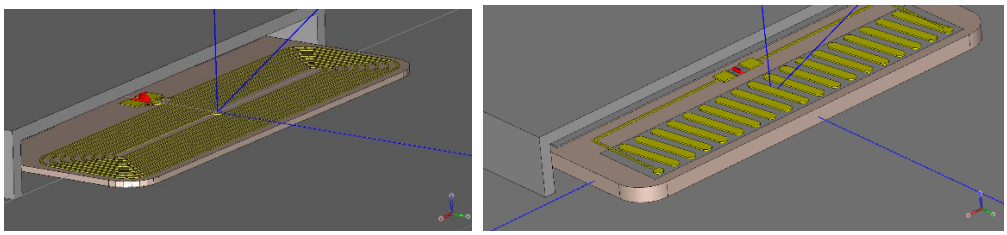

Figure 15: (left canvas) Planar concept; (right canvas) Solenoid concept

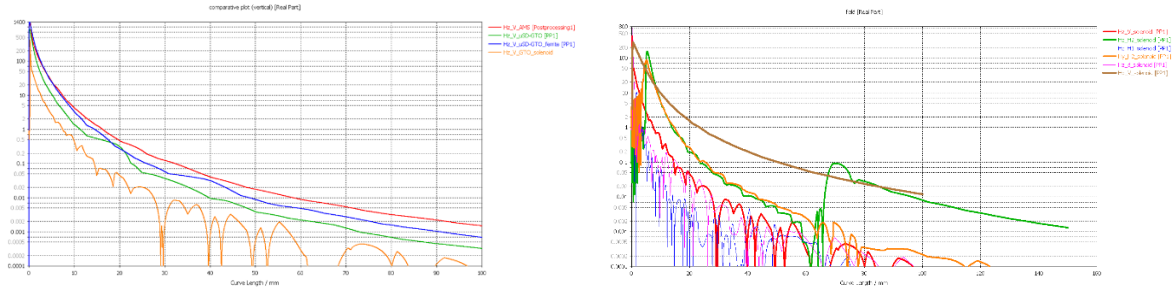

Figure 16: H field along y-axis in case of four different coil antennas: (left canvas) Hz field distribution; (right canvas) Hx field distribution

Once the proper coil configuration is defined and selected, the associated matching network needed to connect the RF-IC to the coil antenna will be optimized to achieve optimal power consumption in all operation conditions and comply with ISO standards.

The correct matching of the antenna to the RF-IC by discrete passive components is very important to achieve maximum RF performance in metal environment. The matching is a trade-off between the antenna current during active modulation and the maximum current ratings for a nanoSIM or a microSD card. The matching should be optimized for a nanoSIM or microSD card within the socket in the mobile phone considering all metallic influences.

Two main simulation set-up and relative test scenarios will be developed. The first one is required to optimize the current consumption and the matching. The second one is needed to optimize the load modulation according to the PICC test procedure defined in ISO 10373. These two simulation configurations are shown in Figure 17 and Figure 18. Simulation results will be validated by experiments and the final solution will be implemented in the system hardware demonstrator.

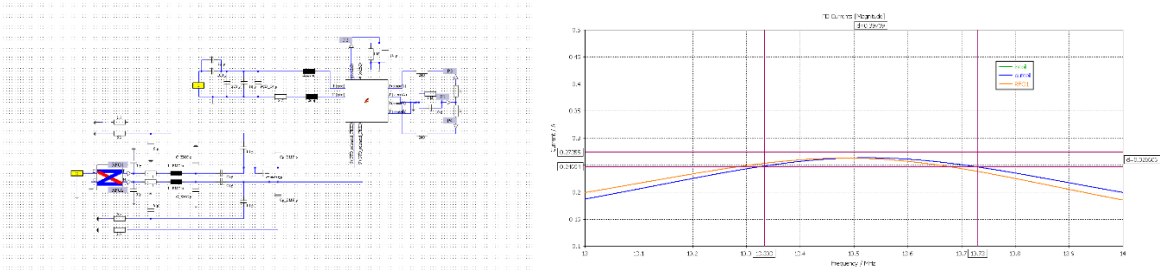

Figure 17: (left canvas) RF-IC matching configuration model; (right canvas) optimized current consumption in operational condition

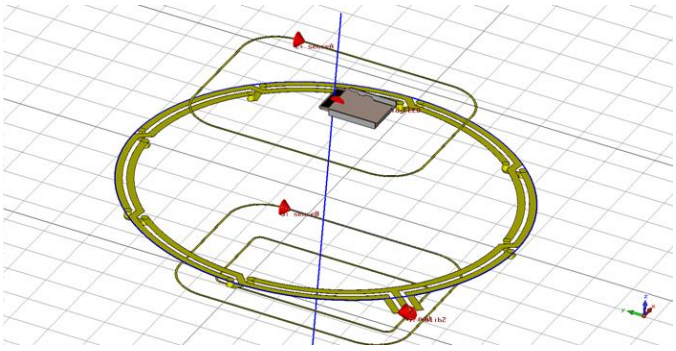

Figure 18: 3D Simulation model of an ISO10373-6 class-6 set-up with a microSD card

3.3 Software Components

MATTHEW is consisting of many software components. Some of the most security critical software components are a secure operating system, secure and securely implemented protocols, secure point-to-point connections, and secure and securely implemented cryptographic primitives. While some of those components already reached a very mature state, others are still in their infancy. Nevertheless, within MATTHEW we will both perform iterative research on mature software components and perform MATTHEW-oriented, foundational research whenever necessary.

In the following, all the software components within the MATTHEW platform are described with special focus on the requirements of the MATTHEW project.

3.3.1 Security Software Bricks

Apart from the secure operating system that handles multiple processes and the communication of the multiple secure elements with each other, cryptography is one of the key enablers of many security concepts. Cryptographic primitives can be classified in keyless, symmetric-key, and

asymmetric-key primitives. Several textbooks like [9] give a comprehensive but detailed overview on such primitives

Within MATTHEW all of those cryptographic primitives and their secure implementations are required. However, privacy preserving protocols, group signature schemes, and attribute-based credentials highly rely on advanced asymmetric primitives. Those primitives can be based on RSA-like cryptographic problems or on discrete logarithm problems. Especially the set of Elliptic Curve Cryptography based bilinear maps, also known as pairings, seems to be a key enabler for future privacy preserving protocols.

In this mathematical concept from each of two groups an element is taken and then mapped to a third group. The first two of those groups are usually, but not necessarily, defined over elliptic curves. The target group that is being mapped to is usually an extension of a prime field. As one can see from this, pairings involve many different concepts that need to be brought in line with each other before it is possible to decide on a cryptographic scheme and to concretely perform an implementation.

Again, there are many excellent sources (e.g. [12], [13] [14]) that give a detailed description of pairings and all underlying cryptographic primitives.

3.3.2 Privacy Enhancing Technologies

A key challenge that is tackled within MATTHEW is the investigation of cryptographic protocols which enable, e.g., secure authentication without revealing one's identity. Different classes of such protocols try to address the privacy problem in different ways:

- Blind signatures make it possible for the signee to have a blinded message signed. As the message stays secret to the signer, blind signatures have been used for many different privacy preserving protocols.
- Group signatures enable the blinding of the signer. As the signer is indistinguishable from a set of different signees, her signatures cannot be tracked nor linked to another one by a malicious party.
- Anonymous credentials go one step further and generalize the concept of group signatures. A party receives credentials from a trusted entity and later on the party can prove the possession of the credentials to a verifying entity without revealing her complete identity but only the attributes that are strictly necessary. The data material exchanged during this phase is referred to as a presentation proof, and is mathematically bound to the credentials in use.

What those types of protocols usually have in common is the existence of a powerful trusted third party. However, there are some characteristics with which the used cryptographic protocols can be distinguished:

- If signatures or presentation proofs are linkable, it is possible for the recipient of multiple signatures or presentation proofs to check whether some of them come from the same party without necessarily knowing which one. This is a feature that is wanted in some privacy preserving protocols but not wanted in others.
- Another feature is the opening of signatures or presentation proofs. If opening is possible a trusted third party (a.k.a. the opener or inspector) is able to de-anonymize the signature or presentation proof and fully identify the party it originates from. Again, it depends on the application whether this feature is wanted.

- However, the most critical feature that is required for the transferring of credentials and the invalidating of malicious users is the possibility of revocation. As some credentials might have to be invalidated at some point, the trusted third party must have the capability to notify all potential verifiers that some credentials have been revoked. This feature is one of the most critical features and has up to now not even been sufficiently solved for currently existing public-key infrastructures (PKI). Technically, there exist at least two solutions for today's PKIs, namely revocation lists and online certificate status protocols, however, for performance reasons hardly anybody is checking whether certificates were revoked. Among privacy preserving authentication techniques, enforcing revocation also poses a number of critical problems in terms of performance (too slow) or user-friendliness (loss of user experience).

With respect to the current state of research it is perceived that revocation will be a base component for the transfer of credentials, one of the fundamental research goals within MATTHEW.

3.3.3 Transfer of Credentials

While it is easy for two parties to share a single bus ticket made out of paper, it is a huge challenge to ensure privacy and to have the possibility to transfer credentials between the two parties. The fundamental difference between a ticket printed on paper and an electronic ticket is that it may be hard to copy a printed paper while it is easy to copy digital data. A typical example of this observation is the use of bank notes versus the one of electronic money. However, e.g., for public transport systems it might be a key feature for customers to share their bus ticket that is valid for a longer period.

Therefore, especially within MATTHEW it is a key challenge to investigate protocols that enable the transfer of credentials between multiple parties and entities. There are two ways to transfer credentials:

- It is possible to transfer credentials online, if both entities have access to the trusted third party. Hereby the trusted third party might invalidate the original credentials (the original ticket) and generate new credentials (a new ticket) for the new entity. For that to work it is necessary to invalidate, to revoke, the original ticket and tell all potential verifiers and inspectors that the original ticket was revoked. Therefore, the online transfer scenario only works if revocation works and revocation is thought of already during the design phase of the cryptographic protocol.
- What users actually want is to transfer their tickets offline, without internet access. Solving this challenge might be possible using special cryptographic protocols or with special hardware that ensures the invalidation of the original credential, or a combination of both. An offline transfer of credentials works if it can be ensured that (i) both parties can cryptographically proof that they are authentic devices, (ii) at no point in time both parties store the full credentials within their non-volatile memories, and (iii) no malicious user is able to pose as authentic device.

It is subject to further research within MATTHEW to find a cryptographic protocol that fulfils these functional requirements, cryptographic requirements, and practical requirements. This protocol will rely on a secure implementation which, in turn, will rely on a secure operating system.

3.3.4 Secure Entity Operating System and Application

3.3.4.1 Global Architecture Overview

The secure entity operating system implements the **Java Card** platform specification, developed by Sun Microsystems technology (later a subsidiary of Oracle Corporation), and used to run securely Java-based applications (applets) on smart card devices. Java Card aims at defining a standard smart card computing environment and API allowing the same Java Card applet to run on different smart cards, much like a Java applet runs on different computers.

Furthermore, the secure entity operating system supports the **Global Platform** specification that defines the secure and interoperable deployment and management of multiple applications on secure chip technology. This ensures that secure solution issuers are not locked into single source commercial relationships.

The synoptic represented bellow describes the software architecture of the secure entity operating system:

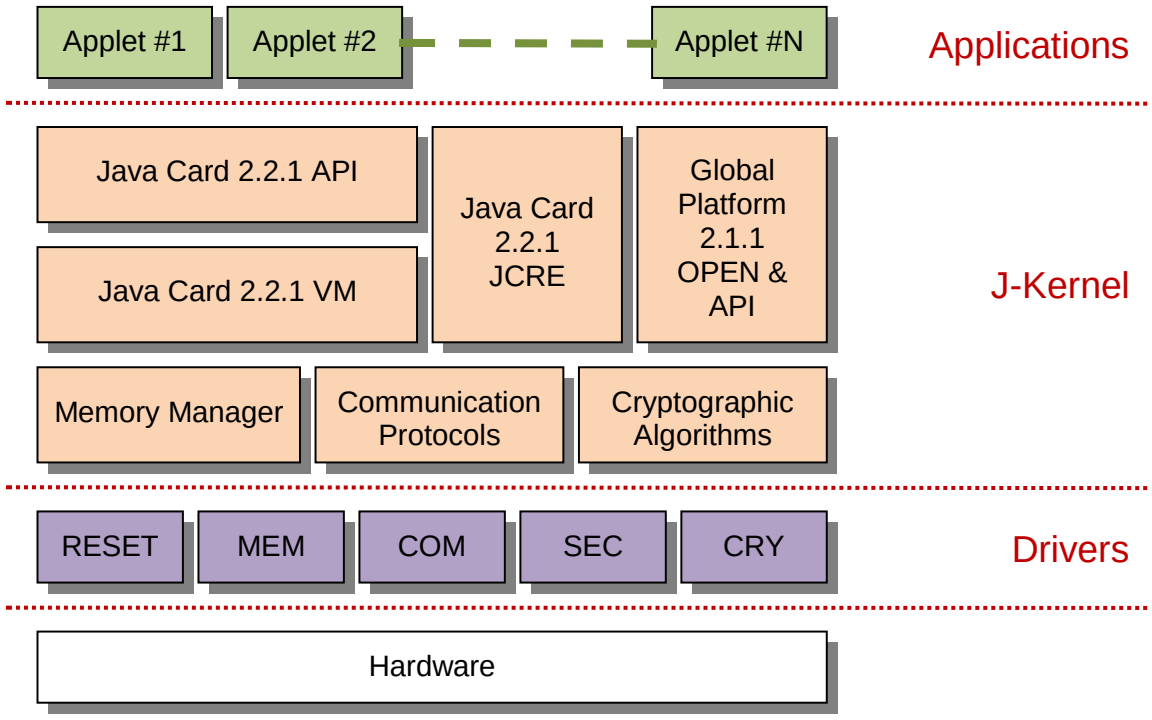

Figure 19: Security entity operating system architecture

The software architecture is mainly divided into 3 sub-systems relying on clearly defined APIs. Each sub-system is divided into several software modules performing specific operations.

3.3.4.2 Driver Sub-System

The driver sub-system is the hardware abstraction layer specially developed and optimized for a specific hardware platform. This software layer can be partially or totally replaced to address a different secure element while keeping the J-Kernel untouched.

- The **RESET** module is in charge of the device start-up, the hardware/software interrupts management, and the memory management unit (MMU) configuration.
- The **MEM** module embeds the EEPROM/Flash driver and the backup management (transaction mechanism).
- The **COM** module manages the low level communication interfaces and the associated hardware (Contact T=0 & T=1 driver, Contactless RF driver, Timers).
- The **SEC** module is used as abstraction layer to access the main security features like counter-measures, random, CRC or fault protection.
- The **CRY** module is in charge to execute low level algorithms (DES, AES, RSA, SHA1…SHA512, ECC…)

3.3.4.3 J-Kernel Sub-System

The J-Kernel sub-system is a portable, Java Card-based system able to execute secure smartcard application (Applets). The J-Kernel sub-system matches the APIs well defined by the Java Card specification, allowing application developers to deploy their applets on all Java Card-based platforms.

- The **Java Card JCRE** module is the Java Card Runtime Environment that is in charge of the internal resources of the secure element and the life cycle of installed applets. Particularly, this module is the Java Card Virtual Machine.
- The **Java Card API** module exposes the Java Card and proprietary (if required) application programming interfaces to Java Card applets.
- The **Java Card VM** module embeds the Java Card interpreter that executes the byte-code while verifying the execution is safe.
- The **Global Platform** module executes and controls calls to Open Environment functions (Security Domains, Data Authentication Pattern…)
- The **Memory Manager** module controls the memory allocation and memory access of the software platform.
- The **Communication** module is in charge of managing the different communication protocol stacks used to communicate with the secure element. The Communication module mainly embeds the ISO7816 T=0 and T=1 protocol stack to access the secure chip in contact mode, and the ISO14443-4A/B stack to access the secure chip in contactless mode.
- The **Cryptographic** module processes high level algorithms (CBC, MAC, PKCS1, paddings…).

3.3.4.4 Application Sub-System

The application sub-system is the higher software layer that contains the Java Card applications. According to the targeted market, applets can be pre-installed by the card issuer (e.g. SIM application) at the secure element personalisation stage, or installed in the field by the customer or final user post-issuance (e.g. Payment or Transport application).

In case the secure element can access a data connection through a GSM operator network (e.g. SIM card or microSD card plugged into a mobile phone), the installation of the applet could be done over the air thanks to a service provider TSM.

Applets can be developed in such a way to cover a wide range of applications which require a high level of security. Particularly, the MATTHEW project is mainly focusing on the Mobile Payment, Access Control and Ticketing applications.

Mobile Payment Applets

The embedded secure mobile payment solution is split into 2 different applets:

- The **Proximity Payment System Environment** (PPSE) applet is, in a first time, selected by the remote point of sale (POS) terminal. At this stage, the secure element returns to the POS terminal an exhaustive list of all mobile payment applications available.

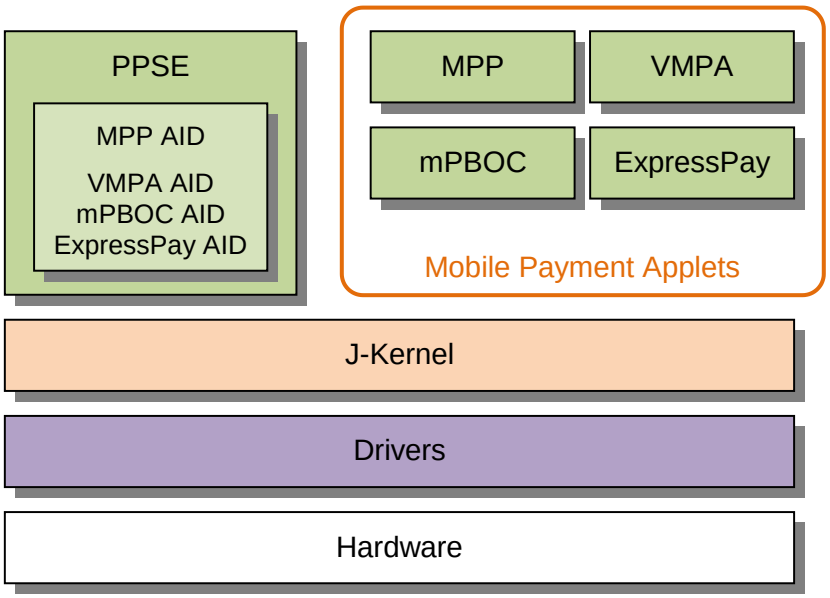

Figure 20: Synoptic of PPSE applet

- According to the supported payment schemes, the POS terminal selects the appropriate **Mobile Payment** applet that generates a transaction cryptogram as a proof of payment. Nowadays, many mobile payment applications are proposed by different financial services corporations, like :
    - MasterCard Mobile Paypass (MPP)
    - VISA Mobile Payment Application (VMPA)
    - American Express ExpressPay
    - People's Bank of China Mobile Application (mPBOC)
    - Japan Credit Bureau (JCB) J/Speedy

  In the context of the MATTHEW project, only the MPP mobile application is described in details as it will be selected for the mobile payment demonstrator.

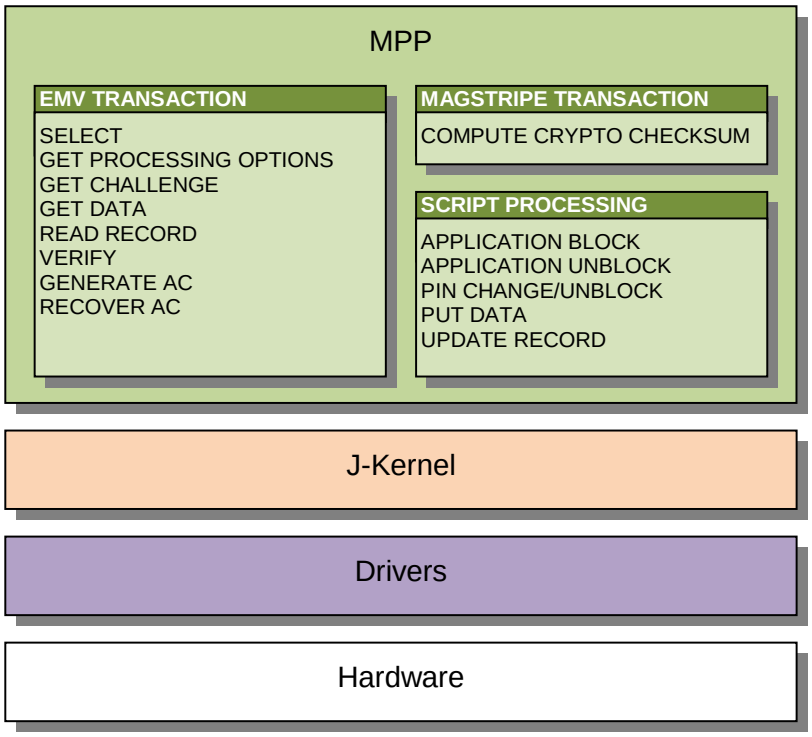

Figure 21: Synoptic of MPP applet

The MPP mobile application is able to support the 3 transaction schemes defined by MasterCard:

- The **Paypass Magstripe** payment transaction scheme which is close to a reading operation of a magnetic stripe, with extra security features.
- The **Paypass MChip/4** payment transaction scheme that permits a safer transaction using for example Dynamic Data Authentication (DDA) mechanism.
- Thanks to the OTA connectivity, the **Management** transaction scheme is used by a distant Authorization Server (e.g. Bank TSM) to administrate the mobile payment application installed into the secure element (e.g. transaction counter reset, PIN code unblock, etc...).

Access Control Applets

Two of the required features of the access control applets are to provide a secure mutual authentication between the user and the verifier and then a secure transfer of user's identification data to the verifier. For this purpose the CIPURSE cryptographic protocol was chosen. The protocol is based on an open standard and meets the requirements by means of advanced cryptography. The authentication phase engages a well-known three-way challenge and response technique. The data exchange is secured by AES encryption and MAC integrity protection.

There are two separate applets, one for the user and the other for the verifier. Both of them will implement CIPURSE protocol functionality but the other features will be different:

- The access control **User applet** is deployed on the user's secure element. It holds cryptography keys for the authentication with the verifier and user's identification data. After successful authentication it enables the verifier to read the identification data.
- The access control **Verifier SAM applet** is deployed on a separate secure element (JCOP card) which is connected to the verifier card reader. This applet does not process the actual communication with the user applet. It just holds the keys and provides the functionality for the cryptography to the reader. The reader manages the communication with the user and provides the identification data.

Due to the fact that the CIPURSE protocol is a standard it would be possible to replace the SAM by a secure element with a native or hardware CIPURSE implementation in the future without changing the other parts of the platform. The advantage of the native or hardware implementation could be a better optimization of the cryptography algorithms which could lead to a shorter identification time.

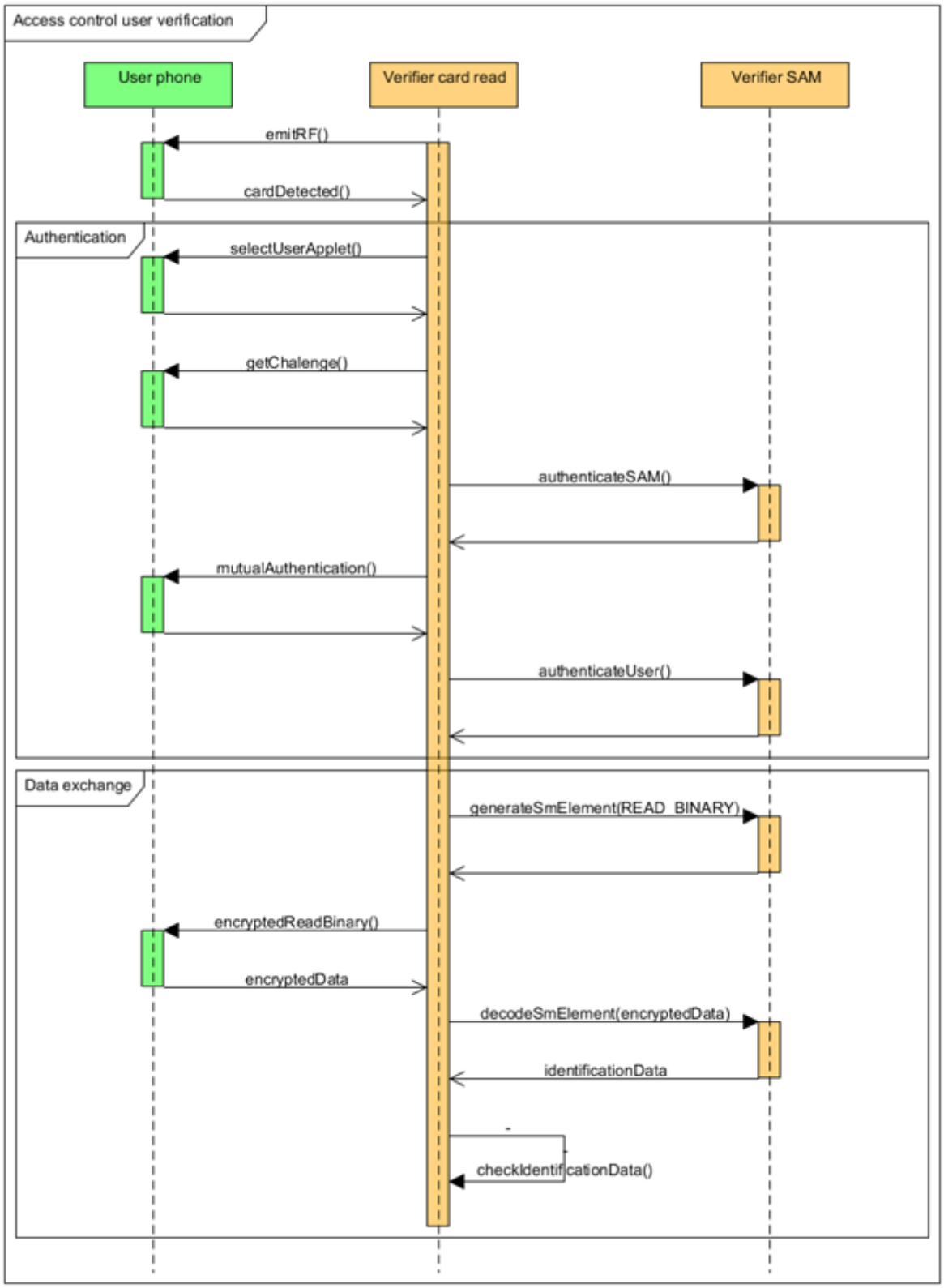

Figure 22: Access control applets communication

3.3.5 Mobile Device Application Software

This section is mainly focusing on the handset software architecture, and the different handset application families the platform requires to achieve the MATTHEW project goals. The mobile device application software belongs to the MATTHEW user role domain.

The implementation details of each software library and application are out of the scope of the project as they are fully dependent on the handset operating system.

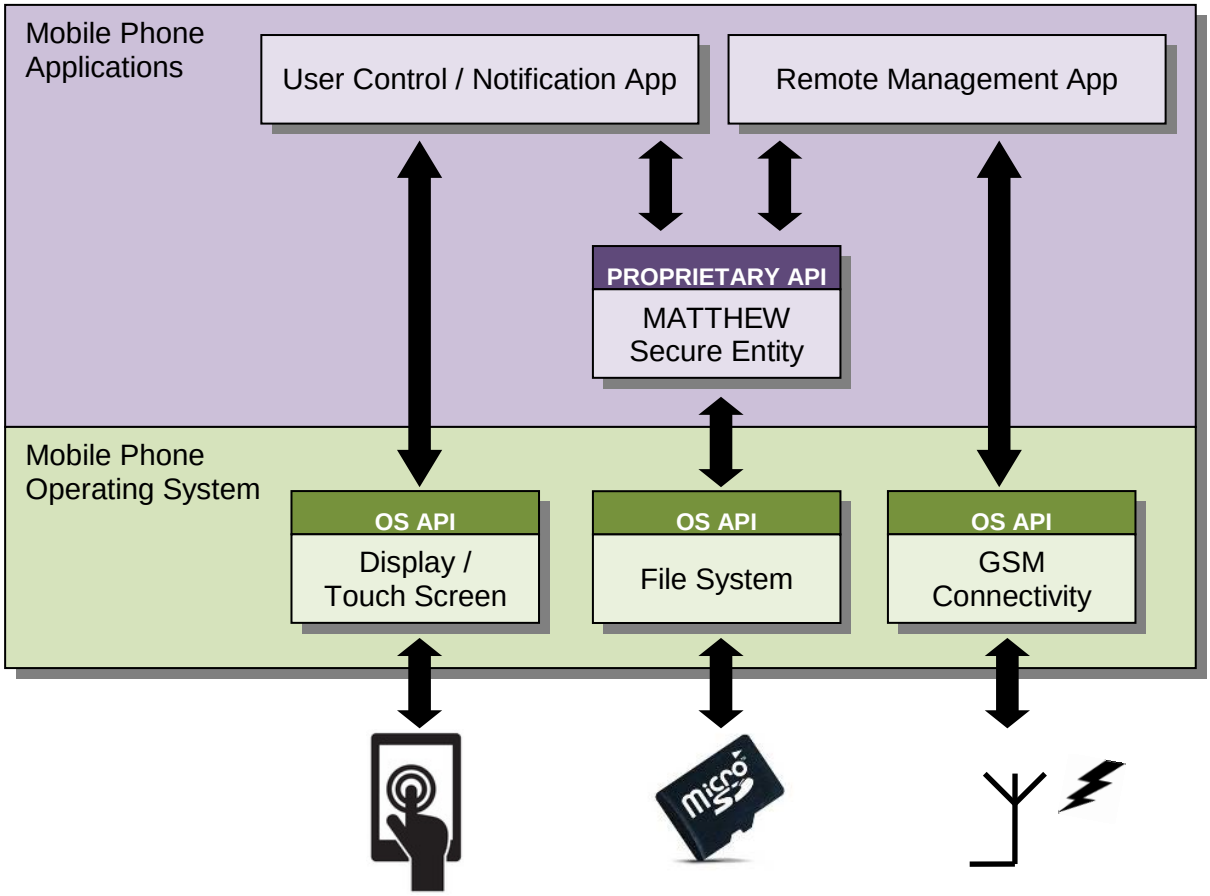

Figure 23: Mobile phone software overview

Basically, the handset operating system provides a well-defined API that a higher level library or application can use to access specific device functionalities:

- The **Display / Touch Screen** software library can be used by an application to interact with the end user (e.g. to enter a PIN code/password, enable the contactless interface, display an information message, etc...).
- The **GSM Connection** software library permits an application to communicate with a distant server (e.g. Trusted Service Manager) to control or administrate the embedded secure element. Optionally, the remote connection can be performed over another wired/wireless link like USB or WiFi according to the constraints of the use case, or according to the server accessibility.
- To pilot a microSD based MATTHEW platform, many specifications already exist like the Advanced Security SD (ASSD) specification from the SD Association, or the Mobile Open API

pushed by the SIM-Alliance. Nevertheless, these API are poorly deployed in the field and do not appear as a standard solution. That is why the common **File System** software library still is the best option to target a wide range of devices.

In order to abstract the complexity of the file system management to the application developer (furthermore, note that file system function calls / management can be pretty different according to the mobile device operating system), a proprietary **MATTHEW Secure Element** library should be developed for each operating system targeted. The MATTHEW Secure Element library provides the following API:

Handler getInstance()

This method allows getting a unique instance of a MATTHEW microSD card. When this method is called for the first time an instance is created and stored then returned. Thus, for all the following calls this same instance will be returned.

The developer may invoke this method at any time to reset the hardware's configuration.

This method is responsible for checking if the inserted microSD is correct or not through the initialization process.

String getVersion()

This method returns the version of the library.

String getFirmwareVersion()

This method retrieves the MATTHEW microSD firmware version. This method is a good way to verify that a MATTHEW microSD card is present and valid.

Void terminate()

This method terminates the microSD session and leaves the microSD card in a disabled state.

Boolean getFieldPresence()

This method returns true if the microSD card is inside the reader magnetic field. This method is only available when the microSD is in contactless mode.

Void selectMode (Byte mode)

This method allows toggling the embedded secure element between the 3 possible modes; ISO7816 contact mode, ISO14443 contactless mode and Idle mode.

There are three values to use in the parameter:

- CONTACT: Select the ISO7816 contact mode.
- CONTACTLESS: Select the ISO14443 contactless mode.
- IDLE: Place the secure element in Idle mode.

The default value when the first instance is created is Idle mode.

Byte getCurrentMode()

This method returns the current mode of the embedded secure element (see selectMode method).

Void enableAPDUChaining(Boolean enable)

This method enables or disables the automatic management of the APDU chaining. When enabled, then the library will send directly the Get Response command upon reception of SW 61XY or resend the APDU command with Le=XY when receiving SW6CXY. When disabled, it is up to the application to decide what has to be done when receiving SW 61XY or SW6CXY.

String coldReset()

This method performs a cold reset of the embedded secure element and retrieves the ATR.

This method can only be invoked when CONTACT mode is selected. The secure element must be reset before sending APDUs.

String warmReset()

This method performs a warm reset of the embedded secure element and retrieves the ATR.

This method can only be invoked when CONTACT mode is selected. The secure element must be reset before sending APDUs.

Byte[] sendAPDU(Byte[] apdu)

This method allows communicating with the secure element via APDU exchange. The communication is used over command and response APDUs.

This method can only be invoked when CONTACT mode is selected.

In the context of the MATTHEW project, 2 different handset application families may be developed.

- **User Control and/or Notification Application:** Such application permits to the end user to control through the developed Graphic User Interface (GUI) the MATTHEW platform behavior. For example, the user may decide to start a payment or access control operation, interacting with the GUI. Furthermore, the application may pop up an information message or notify the end user about the completion of an operation. For example, the application may display the amount of the payment transaction.
- **Remote Management Application:** These are usually “hidden” applications that run on the system background and are used by the service provider to control remotely the microSD embedded secure element. Operations are usually trigged periodically or according to a specific end user’s manipulation (for examples transaction counters returned to the service provider server when a threshold level is passed).

Chapter 4 Deployment Scenarios

Connecting back to the description of the use cases from which the requirements for the platform had been derived and summarized in deliverable D1.1 this chapter describes the MATTHEW deployment scenarios and the corresponding demonstrators as dedicatedly tailored sub sets of the MATTHEW application framework. As a reference the application framework overview is shown here again. In this picture the handhelds and secure entities in the blue encircled area belong to the MATTHEW user role domain whereas the components which are part of the verifier role domain are marked in yellow. The background services are outside of these two areas.

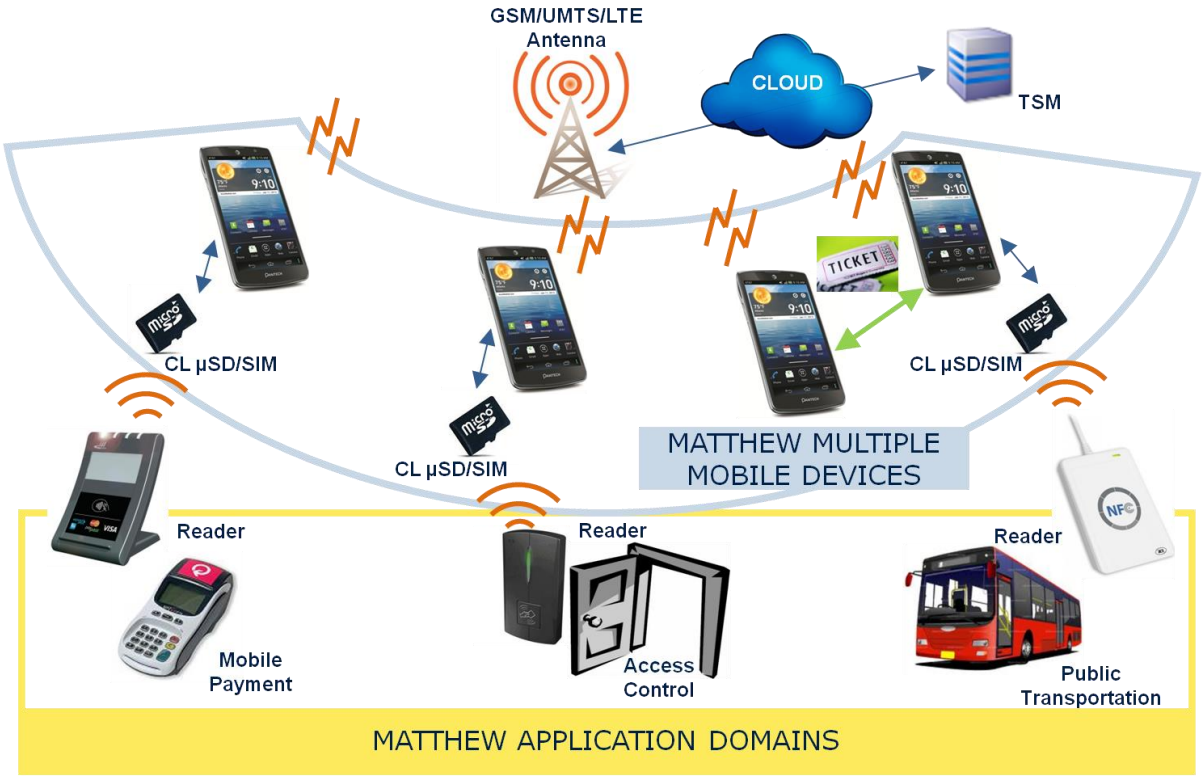

Figure 24: MATTHEW application framework overview

4.1 Use Case “Mobile Payment”

The Mobile Payment use case intends to demonstrate the full compliancy of the active contactless technology developed in the MATTHEW project, embedded in a microSD form factor, in terms of RF and timing performances as well, in front of a standard contactless payment infrastructure.

A complete interoperability of the MATTHEW platform with the current deployed POS terminals is mandatory to support and promote such payment application.

4.1.1 Demonstrator

The Mobile Payment use case demonstrator is based on:

- An offline standalone **VivoPAY 5000 POS terminal** running Paypass MagStripe demo transactions representing the MATTHEW verifier role domain.
- A **MATTHEW microSD platform**, plugged into a **Samsung Galaxy S4** mobile phone.
- A **Proximity Payment System Environment** (PPSE) applet and a **MasterCard Mobile Paypass** (MPP) applet, installed on the embedded secure element, supporting Paypass MagStripe and (optionally) Paypass MChip/4 transaction schemes. The MPP applet is personalized to support only single tap transaction scheme.
- An **Android Payment Demo** application running on the mobile phone, embedding the MATTHEW Secure Element library. This application is used to start a payment transaction (enable contactless interface), and display to the end user the transaction amount after the payment has been performed.

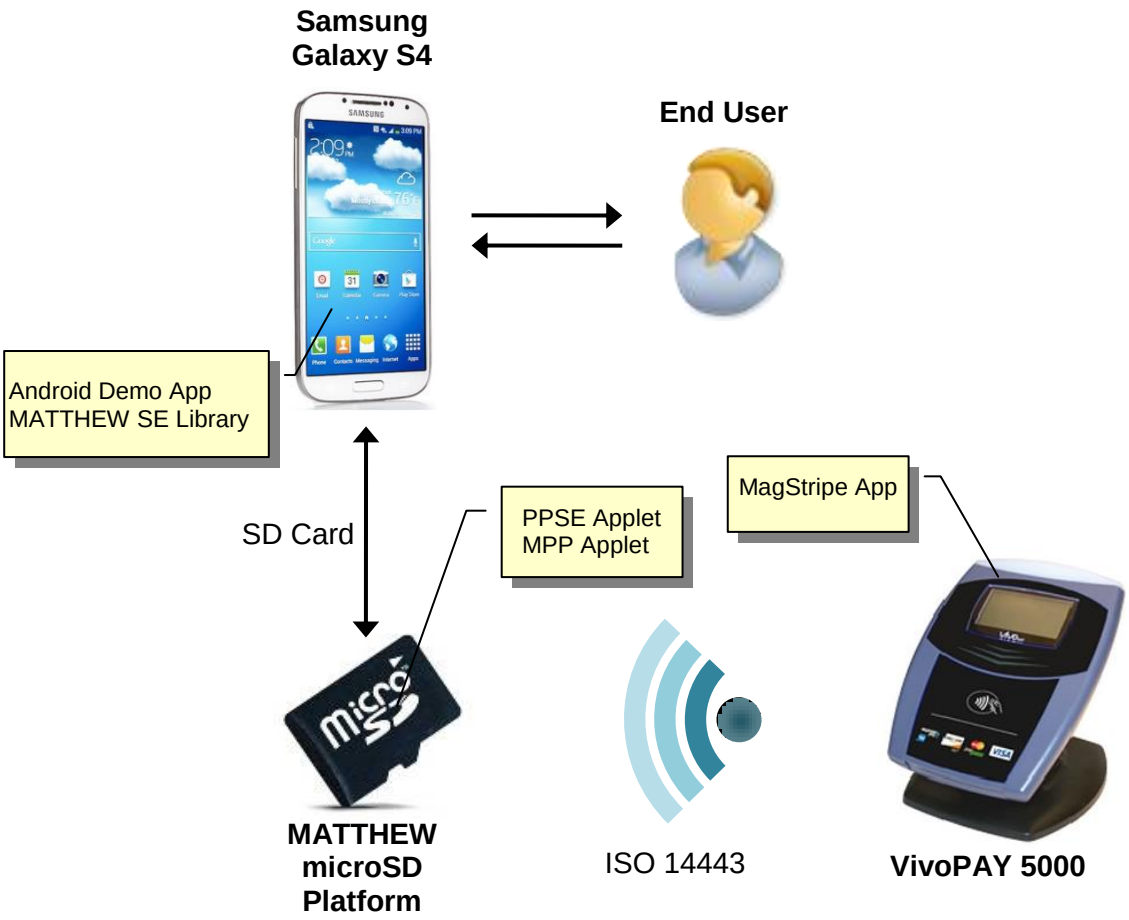

Figure 25: MATTHEW "mobile payment" demonstrator

The microSD platform together with the PPSE and the MPP builds the secure entity and represents – supplemented by the Android payment demo – the MATTHEW user role domain

4.2 Use Case "Access Control"

Objectives

The aim of the second use case is to demonstrate the MATTHEW platform in an access control system. This chapter describes different ways in which the MATTHEW platform will be used for access control. Independently from the physical form factor – microSD card or SIM card – the MATTHEW user and verifier role components can be used for the access control system in two modes:

- In mode one only an NFC secure element will be used in the handheld without any cooperation with the operating system of the handheld. The reader will only communicate with the MATTHEW secure element.
- In mode two a dedicated software routine in the reader will be activated after successful communication between the MATTHEW secure element and the reader and in the next step the reader will communicate via NFC directly to the handheld's NFC application.

The first case is completely independent of the operating system of the handheld, the second case is dependent on the operating system and for demonstration we will use OS ANDROID v4.2 or higher.

4.2.1 Demonstrator

Subsystem overview

The access control subsystem consists of these parts:

K4Server
K4Master
Netmodul
Reader
K4Manager
K4Alarm
K4Sled
K4Vrátnice
K4EIBServ, K4UserAuthService, K4AlarmMailService …
Kamery Mobotix
Conection to CardPress

Functions

The access control subsystem provides services (K4Server, K4Manager, K4Alarm,...) which integrate all functions for the card management and access control transaction management. The K4Server is connected to the K4Master which works as an intelligent concentrator. All other components are online connected to this concentrator. Based on the decision of the concentrator the access control terminals (CKP4, CKP22...) enables or disables the input or output of persons through the gates of the systems. The access control terminals can also work offline but then some of the functions like e.g. ANTIPASSBACK will not be available. All data and transactions are stored in the database server. Connection of all components is done by LAN, connection of access control terminals and K4Master can be done also by RS485.

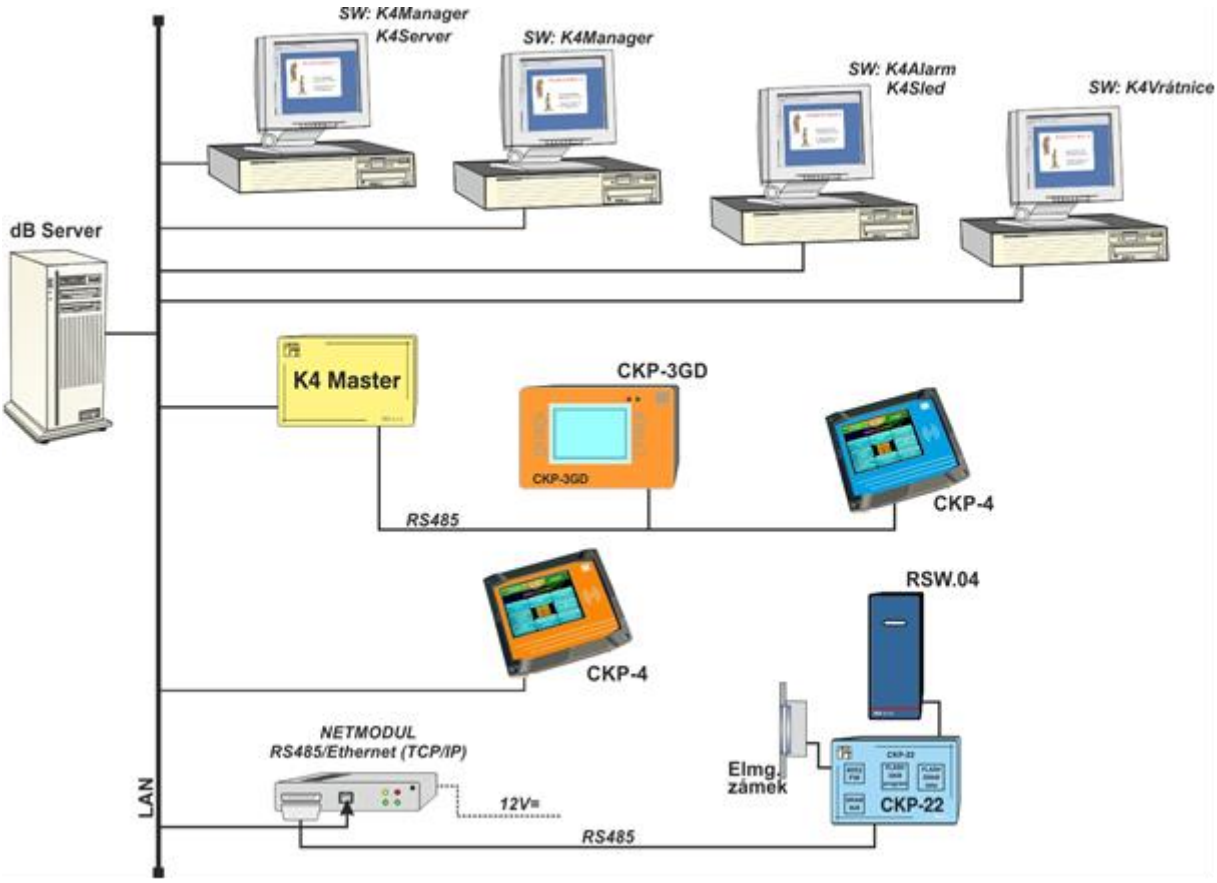

Figure 26: Access control subsystem

Components

Database server:

We are using a standard ORACLE or MICROSOFT SQL database with two types of data:

Configuration data

- Description of system configuration
- Persons and their access rights
- Cards, photos, fingerprints etc.

Transaction data

- Record of door openings
- Log of events
- Log of user actions

K4Server

- Transfer data between database and terminals
- Generate configuration data for K4Masters
- Keep connection to all K4Masters
- Read from K4Masters records of door openings
- Connect other applications

K4Master

- Receive configuration from K4Server
- Keep connection to access control terminals
- Receive record of door openings and send it to K4Server
- Realize ANTIPASSBACK if enabled
- Check of PIN (optional)
- Process cards with special functions (optional)
- Connect to other systems (optional)
- Cipher communication (optional)

CKP4 Terminal

- Communicate to the K4Master
- Store configuration of terminals
- Store transactions
- Send transaction to the K4Master
- In offline state check if card is enabled and if yes, open door
- Display messages to the user

RSW04

- Read data from card
- Send data to the CKP4 Terminal
- Indicate the system status optically and acoustically

Interfaces

The communication between the access control terminal and the K4Master runs over LAN or RS485 using a special IMA protocol. In the future a switch to an OSDP (Open Supervised Device Protocol) protocol provided by SIA (Security Industry Association) is planned. The databases are connected to the K4Server by a specific protocol which enables the communication to an ORACLE or MICROSOFT SQL database.

Implementation

For the demonstration only a part of the access control components will be used. To the terminal CKP4 an offline configuration with the list of enabled card numbers will be downloaded. The same numbers will be personalized to the MATTHEW card. When the card is read CKP4 will check the card number against the offline database and if it matches the door will be opened.

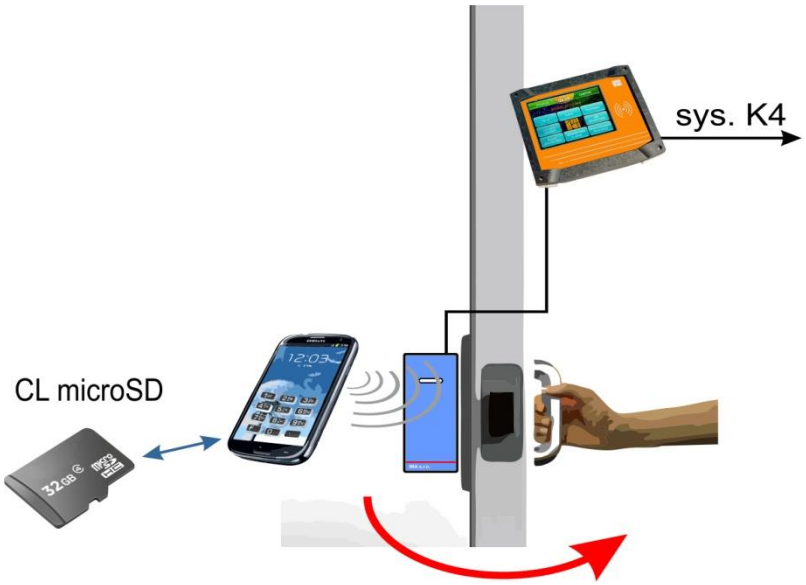

Figure 27: Access control demonstrator

Access mode description

The possible access control modes are detailed in D1.1, Section 2.2 "*Requirements from Use Case 2 – Advanced access control application".*

4.3 Use Case "Ticketing"

The use case for the advanced ticketing application has been described in D1.1, Section 2.3. This use case discusses an advanced ticketing application with multiple parties (User, Issuer, Verifier, and Inspector), multiple transaction scenarios (Issuance, Presentation, Inspection, Revocation, Transfer) and is currently under intensive research. The technology readiness level of the privacy enhanced transfer of credentials is perceived as TRL3 to TRL4 and thus expectations towards the degree of integration of a protocol demonstrator are not the same as in the other use cases.

At this stage, a ticketing system featuring some of the desired properties has been sketched in the context of single-use and multiple-use tickets, i.e. tickets that can be used a prescribed number of times (only once or say, 5 times in total) and that we collectively refer to as limited-use tickets. Thus limited-time tickets i.e., tickets that are used a virtually unlimited number of times within a prescribed time window, are not yet supported. Besides, in the current system, full revocation is not satisfactorily achieved yet.

It is planned to cooperatively research on the cryptographic protocols needed for the advanced ticketing application with privacy enhanced transfer of credentials, but not to constrain the design space by limiting the consortium to a certain platform or architecture. Also when it comes to securely implementing a to-be-researched protocol, it has to be verified which architectures or platforms will provide sufficient performance and security (e.g. against implementation attacks). Therefore, in the pursuit of the advanced ticketing use case, which is closely linked to the research done within work package WP2, components of the MATTHEW platform will be used whenever it is possible and the platform is suitable, but the focus is going to be on novel protocols, novel architectures, and novel secure implementations that will be investigated for their practicability and their security characteristics.

4.3.1 Demonstrator

As a consequence of the aforementioned rather low technology readiness level, the demonstrator for the use case “ticketing” is expected to have the shape of laboratory-style software prototypes, either on PCs or embedded devices. MATTHEW user role components as well as software routines representing the MATTHEW verifier role domain may be implemented as dedicated abstract models in high level modelling languages like MATLAB or Mathematica, or in Java or C. Nevertheless, the results of the investigations on performance efficiency and security strength of the proposed protocol steps will be demonstrated as planned.

Chapter 5 List of Abbreviations

ACF	Active Communication Frontend
ACLB	Analog Contactless Bridge
APDU	Application Protocol Data Unit
AES	Advanced Encryption Standard
API	Application Programming Interface
ATT	Active Transmission Technology
CBC	Cipher Block Chaining
CL	Contactless
ECC	Elliptic Curve Cryptography
ECC	Error Correction Code
EMV	Europay International, MasterCard and VISA
eSE	Embedded Secure Element
GSM	Global System for Mobile Communications
IC	Integrated Circuit
ISO	International Organization for Standardization
JCB	Japan Credit Bureau
JCOP	Java Card Open Platform
JCRE	Java Card Runtime Environment
MAC	Message Authentication Code
MPP	Mobile PayPass
NFC	Near Field Communication

OS	Operating System
OSDP	Open Supervised Device Protocol
OSPT	Open Standard for Public transportation
RF	Radio Frequency
PC	Personal Computer
PIN	Personal Identification Number
POI	Point of Interaction
POS	Point Of Sales
PPSE	Proximity Payment System Environment
PUF	Physically Unclonable Function
RSA	Rivest, Shamir, Adleman
SAM	Security Access Module
SD	Secure Digital
SIM	Subscriber Identification Module
STREP	Specific Targeted Research Project
SWP	Single Wire Protocol
TRL	technology readiness level
UICC	Universal Integrated Circuit Card
VM	Virtual Machine
VMPA	VISA Mobile Payment Application

Chapter 6 Bibliography

[1] Pappu, R.S. *Physical One-Way Functions.* Phd thesis, Massachusetts Institut of Technology, 2001.

[2] Maes, R., Van Herrewege, A., Verbauwhede, I. *PUFKY: A Fully Functional PUF-based Cryptographic Key Generator*. Lecture Notes in Computer Science Volume 7429, pp 302-319, 2012.

[3] O'Donnell, C. W., Suh, G.E., Devadas, S. *PUF-Based Random Number Generation.* Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

[4] Dodis, Y. , Ostrovsky, R. , Reyzin, L., Smith, A. *Fuzzy Extractors: How to generate strong keys from biometrics and other noisy data*. SLAM Journal on Computing, 38(1): 97-139, 2008.

[5] Bösch, C., Guajardo, J., Sadeghi A.-R., Shokrollahi, J., Tuyls, P. *Efficient Helper Data Key Extractor* on FPGAs. International Association for Cryptologic Research, 2008.

[6] Deutschmann, M. *Cryptographic Applications with Physically Uncloneable Functions.* Master Thesis, Alpen-Adria-Universität Klagenfurt, 2010.

[7] Guajardo, J., Kumar, S.S., Schrijen, G.-J., Tuyls, P., *FPGA Instrinsic PUFs and Their Use for IP Protection.* Lecture Notes in Computer Science Volume 4727, pp 63-80, 2007.

[8] Van den Berg, R., *Entropy analysis of physical unclonable functions.* Master's thesis, Technical University of Eindhoven, August 2012.

[9] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. 1996. Handbook of Applied Cryptography (1st ed.). CRC Press, Inc., Boca Raton, FL, USA.

[10] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, editors. Handbook of El liptic and Hyperel liptic Curve Cryptography. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2006.

[11] D. Hankerson, A. J. Menezes, and S. Vanstone. Guide to Elliptic Curve Cryptography. Springer, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 2004.

[12] Rudolf Lidl and Harald Niederreiter. Introduction to finite fields and their applications. New York, NY, USA: Cambridge University Press, 1986. isbn: 0-521-30706-6.

[13] [Craig Costello. Pairings for Beginners. 2013. url: http://www.craigcostello.com.au/pairing/.

[14] J.H. Silverman. The Arithmetic of Elliptic Curves. Graduate texts in math- ematics. Springer, 2009. isbn: 9780387094946. url: http://books.google. at/books?id=Z90CA_EUCCkC.