Private Fernfachhochschule Darmstadt
Department of Computer Science

– – Diploma Thesis – –

Virtualisation of a SIM-Card using Trusted Computing

submitted by	:	Michael Kasper
mat.-no.	:	870970
evaluated by	:	Dr. Detlev Zimmermann
supervised by	:	Dr. Andreas U. Schmidt (FhG-SIT)
		Dipl.-Inform. Nicolai Kuntze (FhG-SIT)
submission date	:	30.04.2007

SIT

Fraunhofer Institut
Sichere Informations-
Technologie

Acknowledgements

First of all I like to thank Dr. Detlev Zimmermann, who supervised and guided me through the process of creation. I also would like to express my thanks to my supervisors at the FhG-SIT, Dr. Andreas U. Schmidt and Nicolai Kuntze. By their professional support, assistance and cooperation they have made this thesis possible.

And last but certainly not least, I would also like to thank my parents Renate and Jürgen, and my wife Heike for supporting me during my studies in every possible way and always giving me new motivation for the next steps of my work.

Abstract

The goal of this thesis is to examine, how subscriber authentication in mobile cellular networks could be implemented to the next generation of mobile phones and devices. In this context, we consider and evaluate several architectural directions and propose a novel solution of a software replacement for the Subscriber Identity Module (SIM) based on the *TCG MPWG Reference Architecture*. Therefor, we introduce a virtual software SIM (vSIM) with comparable usage and security characteristics like the smartcard-based solutions.

Our approach demonstrates the substitutability of a SIM card with an adequate trusted software module supported and protected by a trustworthy operating system. In particular, we propose several methods for authentication and enrollment of a subscriber, the practical design and implementation of this concepts and how to deploy it to a trustworthy operating platform. Furthermore, we propose a method for the remote-take-ownership of a device by the mobile network operator and the migration of subscriber credentials between devices.

We will focus the evaluation on a set of benchmarks which are seen as crucial for development and production, as well as for market and user's requirements of mobile devices such as mobile phones. Running a virtual SIM as a trusted and protected software on a mobile device allow significant expansion of services by introducing new usage scenarios and business models, cost reduction and more flexibility, while a high level of security is still available.

Keywords

Trusted Computing; Mobile Trusted Module; Mobile Communication; Identity Management; GSM Subscriber Authentication; Software SIM; Virtualization;

Contents

List of Abbreviations

AIK	Attestation Identity Keys
APDU	Application Protocol Data Units
AuC	Authentication Center
BSC	Base Station Controller
BSS	Base Station Subsystems
BTS	Base Transceiver Station
CRTM	Core Root of Trust for Measurement
EIR	Equipment Identity Register
EK	Endorsement Key
GSM	Global System for Mobile Communications, original acronym: Groupe Special Mobile)
HLR	Home Location Register
HPLMN	Home Public Land Mobile Network
IMEI	International mobile equipment identity
IMSI	International Mobile Subscriber identity
LAI	Location Area Identity
LAI	Location Area Information
LFSR	Linear Feedback Shift Register
ME	Mobile Equipment
MLTM	Mobile Local-Owner Trusted Module
MRTM	Mobile Remote-Owner Trusted Module
MSC	Mobile Switching Center
MSISDN	Mobile Station ISDN Number

MS	Mobile Station
MSRN	Mobile Station Roaming Number
MTM	Mobile Trusted Module
MTP	Mobile Trusted Platform
NSSS	Network- and Switching Subsystem
OMSS	Operation and Maintenance Subsystem
PLMN	Public Land Mobile Network
RAN	Radio Access Network
RIM	Reference Integrity Metrics
RSS	Radio-Subsystem
RTE	Root of Trust for Enforcement
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
RTV	Root of Trust for Verification
SAT	SIM Application Toolkit
SRK	Storage Root Key
SS7	Signaling System 7
TCG	Trusted Computing Group
TC	Trusted Computing
TE	Ttrusted Engine
TMSI	Temporal Mobile Subscriber Identity
TPM	Trusted Platform Module
TSS	Trusted Subsystem
UMTS	Universal Mobile Telecommunications System
VLR	Visitor Location Register
vMTM	virtual Mobile Trusted Module
vSIM	trusted virtual Subscriber Identity Module
WLAN	Wireless Local Area Network

List of Figures

List of Tables

Chapter 1

Introduction

In order to increase the level of security of mobile phones and mobile devices the *Trusted Computing Group (TCG)*, a consortium of leading hardware and software vendors, has recently published a specification draft [Tru06b]. This specification offers new potentials for implementing trust in mobile computing platforms by introducing a hardware-based trust anchor. This hardware chip is called *Mobile Trusted Module (MTM).* The TCG MPWG holds to the idea that all next-generation mobile phones will be equipped with a MTM compliant to that specification.

The goal of this thesis is to examine, how subscriber authentication in mobile cellular networks could be implemented to the next generation of mobile phones and devices. Therefore, several architectural solutions have to be considered and evaluated. We will focus the following benchmarks which are seen as crucial for development and production, as well as for market and user's requirements of mobile devices such as mobile phones: (1) *Security and Trustworthiness*, (2) *Cost-effectiveness*, (3) *Flexibility and Scalability*, (4) *Portability and Mobility*, and (5) *Usability, Compatibility and Acceptance.*

In the next sections we will give an outline of the thesis' background. Then we will explain the benchmarks given in the beginning. Finally we will sketch out the structure of this thesis.

1.1 Background of the Thesis

Mobile Communication is one of the fastest growing and most demanding telecommunication technology worldwide. From iGR's 2006 survey [iGR06], the industry has sold roughly 809 millions of new handsets in 2005. It is expected that the number grows to slightly over one billion by 2010 and smartphones will have a permeation of approximately 21 percent of all mobile handsets shipped worldwide.

Increasing Requirement for Trusted Mobile Devices

The evolution of next generation mobile handhelds offers an opportunity for new sophisticated and complex digital services and applications for heterogeneous mobile ecosystems. Thus, the door is opened for new vulnerabilities and security risks in the mobile domain as well. These include trojan horses, computer viruses, worms and corresponding attacks by a malicious adversary. By a combination of these several influences, as shown in Figure 1.1, the need for security and trust in an important point that rises with respect to the underlying technology.

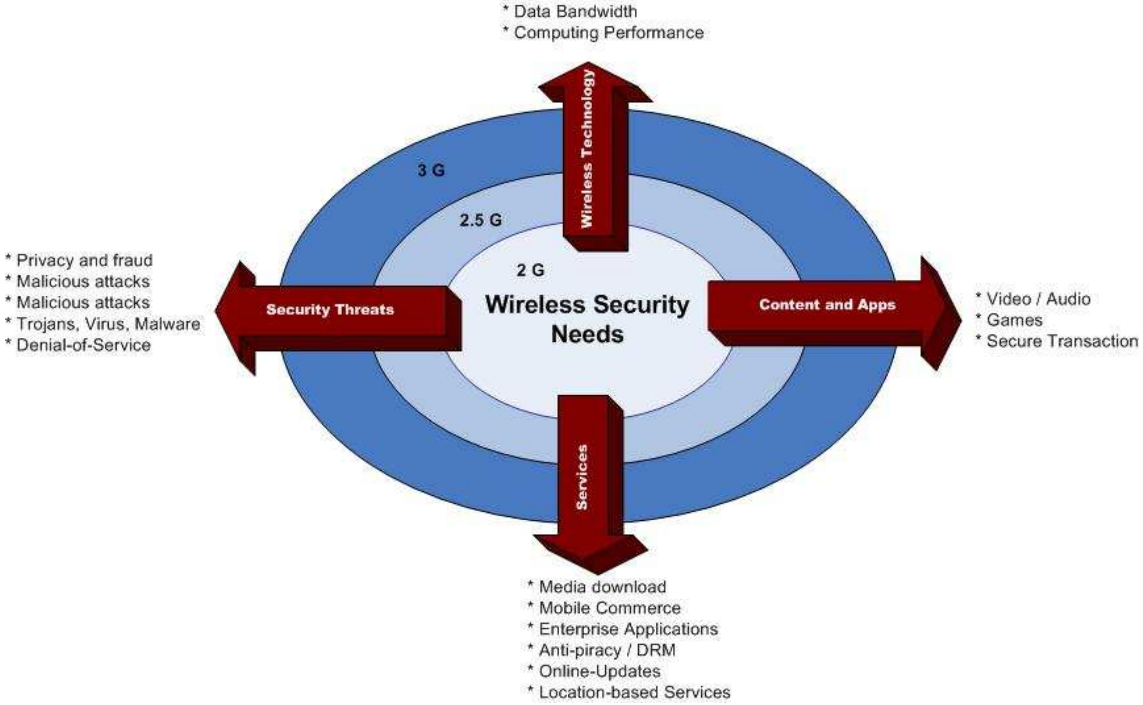


Figure 1.1: *Expansion of Wireless Security Needs*

Trusted Computing in mobile cellular Networks

The *Trusted Computing Group (TCG)* is a consortium of many leading hardware and software vendors. Their main objective is the consideration of how to increase the level of security and establish trust in computing platforms. Several specifications for Trusted Computing were released by this consortium, such as for a small scale embedded trust anchor, called *Trusted Platform Module (TPM)*, building the foundation of trust.

Meanwhile a TPM chip is going to be a part of more and more state-of-the-art desktop computers. And recently the TCG's approach for mobile phones and mobile devices has been drifting into public's spotlight. After three years of development, the TCG Mobile Phone Work Group (TCG MPWG) has published a specification draft in November 2006 [Tru06b], which offers new potentials for implementing trust in mobile computing platforms by introducing a hardware-based trust-anchor. This chip is called a Mobile Trusted Module (MTM) and has comparable properties and features to a Trusted Platform Module [Tru07, Tru05]. Currently, the TCG MPWG reviews a much more universal security architecture for mobile phones and devices, which is called *TCG Mobile Reference Architecture* [Tru06a]. It abstracts a trusted mobile platform as a set of tamper resistant trusted engines on behalf of different stakeholders.

The TCG MPWG holds to the idea that all next-generation mobile phones will be equipped with a MTM compliant to that specification. Through the support of such well known TCG MPWG members like Nokia Corp., Samsung Electronics Co., France Telecom or Ericsson it seems likely that an extensive integration in the upcoming next generation mobile handhelds and devices will happen.

1.2 Benchmarks and Evaluation Criteria

Due to the evolution of next-generation mobile devices, a reconsideration and evaluation of the current mechanisms for SIM-based subscriber authentication is reasonable. However, a sustainable evaluation of the different solutions is only feasible if a set of appropriate criteria and benchmarks is defined. Therefore, the following global benchmarks are identified:

(1) *Security*

The proposed architecture for subscriber authentication requires a minimal set of security characteristics, which are (at least) equivalent to conventional SIM cards, and an appropriate means to provide evidence of its trustworthiness.

(2) *Cost-effectiveness*

An important issue is the cost-value ratio of the underlying hardware architecture. In particular, this affects non-redundancy of cost intensive components without a significant loss of security, depending on the intended use-case.

(3) *Flexibility and Scalability*
Mobile services require an sufficient degree of flexibility and adaptability when external requirements arise, and the ability to be readily enlarged or modified.

(4) *Portability and Mobility:*
The associated subscriber credentials have to be removeable and readily transportable from one device to another. Thus, it enables a subscriber to use its credential with an arbitrary terminal.

(5) *Usability, Compatibility and Acceptance*
The proposed architecture has to ensure an appropriate level of usability and acceptance with a functional interface and grammar compliant to actual standards.

Based on these criteria we compare a set of aspired and existent approaches and weight up whether an equal (or even better) alternative, concerning these other presented solutions, is available or not. If so, we have to identify which technological pre-requirements are needed and how this solution is implementable.

1.3 Architectural Directions and Solutions

In our examination the following four different directions of Subscriber Authentication are considered:

- a single-trust anchor architecture using conventional SIMs,
- a dual trust anchor architecture using conventional SIMs,
- a single trust-anchor architecture using virtual SIMs, and finally
- a client-server architecture using remote SIMs.

Single Trust-Anchor Architecture using conventional SIMs This architecture is the established and proposed means for subscriber authentication in current used, mobile cellular networks, including the mobile communication systems GSM and UMTS [rGPP97b, rGPP07].

Dual Trust-Anchor Architecture using conventional SIMs This approach identifies an architectural direction with two coexisting hardware-based trust anchors, namely the SIM and the MTM. Each anchor is instructed to process different tasks. While the primarily task of the SIM is to identify and authenticate a local user in a secure manner, the MTM is mainly responsible for providing evidence of the trustworthiness of the device and its associated components.

This approach represents the *State-of-the-Art* for next-generation mobile handhelds, according to major members of the Trusted Computing Group.

Single Trust-Anchor Architecture using virtual SIMs Unlike the precedent architecture, the next approach identifies an architectural direction, based on only one hardware-based trust anchor. Through the capabilities of a Mobile Trusted Module Platform to support protected storage, an isolated execution and secure communication, a trusted platform is able to take over the SIM functionality.

In this approach, the specification is considered from a slightly different point of view as is is envisaged by the TCG. Even though, SIM-based authentication is the established and proposed means for subscriber authentication, an alternative is coming up with the advent of the TCG technology in mobile devices.

Client-Server Architecture using a central SIM-Storage Another approach for subscriber authentication in GSM networks is based on a client-server architecture [Imp]. The primarily idea behind this concept is to provide a central smartcard server for storage of any number of conventional SIMs. Each remote GSM client is equipped with a SIM emulation adapter. While performing network authentication, the client uses an already established internet connection and connects to the central SIM server in order to relay the authentication messages.

1.4 Virtual SIM as a means for Subscriber Authentication

We will see in the progress of this thesis, that the third approach *Single Trust-Anchor Architecture using virtual SIMs* turns out to be a suitable and sustainable solution, that is able to compete with the SIM-based solutions. We will demonstrate the substitutability of a SIM card with an adequate software replacement supported and protected by a trustworthy operating system, and is based on a single small-scale trust anchor conformant to the *TCG MPWG Reference Architecture*.

Supporting Protocols for Deployment and Management

Initially we inspect the TCG Reference Architecture in order to check whether this specification meets the demands of our benchmarks and which parts of the specification have to be concretize. Hence, we have to discuss and state more precisely the deployment and management aspects. In this context, we detail the following protocols:

- Remote-Take-Ownership of a vSIM Container,
- Subscriber Enrollment and vSIM Credential Roll-Off, and
- Migration of a vSIM Container and vSIM Credential

The first protocol identifies generic methods for Take-Ownership of a vSIM container by a mobile network operator. Also we show how user enrollment and key delivery mechanisms could be carried out efficiently. The last protocol considers the migration method of vSIM credentials between devices.

Conceptual Models for Subscriber Authentication

Once we are provided with the essential fundament, we are able to introduce a virtual software SIM with comparable usage and security characteristics like the traditional smartcard-based solution. Additionally, running a virtual SIM as trusted and protected software on a mobile device allows significant expansion of services by introducing new usage scenarios and business models, cost reduction, more flexibility and trustworthiness. These characteristics and requirements will be checked against the defined global benchmarks from Section 1.2 throughout this composition.

We design three intergraded conceptual models for authentication in mobile cellular networks using trusted computing.

- Model ”One”: Subscriber Access in mobile cellular Networks based on Trusted Computing with compatibility to GSM - Authentication
- Model ”Two”: GSM-Subscriber Authentication in mobile cellular Networks based on Trusted Computing with Remote Attestation for Restricted-Network-Access
- Model ”Three”: Generalized Subscriber Authentication in IT Networks Infrastructures based Trusted Computing with Remote Attestation for Restricted-Network-Access

These models show how a traditional SIM-Card could be replaced by a software emulation, which runs within an environment, protected and supported by a Mobile Trusted Module. We call this software emulation a "trusted virtual Subscriber Identity Module" - or in short - vSIM.

1.5 Outline of this Thesis

Apart from the present introduction, the thesis is divided in four more chapters. In Chapter 2, we introduce to the necessary background of GSM and Trusted Computing. It begins with an introduction to the GSM communication system in Section 2.1 and its identity management and investigate the existing authentication mechanisms and the technical design of a SIM card. In Section 2.2, we introduce the fundamental concepts of Trusted Computing and provide the reader with essential background information of the significant parts of the *TCG MPWG Reference Architecture*. The core of this thesis is constituted in Chapter 3. Initially, we explicate the use-case scenario in Section 3.1. Then, we give an overview of the vSIM architecture in Section 3.2. In Section 3.3, we consider the deployment and management of vSIM containers and credentials. Here, we discuss how *Remote-Takeownership*, *Subscriber Enrollment*, *vSIM Credential Roll-Out*, and *Migration* could be carried out efficiently. In Section 3.4, we present the conceptual models for subscriber authentication in mobile cellular networks. The last issue of this chapter is discussed in Section 3.5. Here, we introduce to the design of the prototypical implementation. The evaluation and analyis concerning the benchmark are accomplished in Chapter 4. In Section 4.1 a security analysis of the vSIM architecture is given. Section 4.2 focuses on the analysis of the different criteria for evaluation. In Section 4.3, we compare and evaluate our work against the other identified architectural solutions and SIM characteristics. Finally, Chapter 5 summarizes and concludes on our work and point out further research.

Chapter 2

Basics and Fundamentals

This chapter introduces the necessary background of GSM and Trusted Computing. We begin with an introduction of the GSM communication system and its identity management. We will investigate the existing authentication mechanisms and the technical design of a SIM card. Section 2.2 discusses the fundamentals of Trusted Computing and provides the reader with essential background information.

2.1 Identity-Management in GSM-Communication-Systems

In this section we recapitulate the GSM standard and take a look at the system architecture and its provided security mechanisms. This includes a description of the network components which build up the system, and how these elements are interconnected. Furthermore, we describe the Subscriber Identity Module (SIM) and detail the specified commands, which we intend to use for design and implementation of our virtual SIM solution.

Nevertheless the information society is at the threshold of using 3G UMTS [rGPP07, Wal00a] and researchers working on 4G Next-Generation-Networks, we focus on the slightly outmoded 2G GSM System [Wal00b]. There are several reasons for this choice. First authentication in GSM mobile cellular networks is based on such a hardware token which is placed into any handheld. Moreover, the GSM System is the Pan-European-Standard for digital cellular communication and holds, with approximately 2 billion subscribers [Wir06], the currently the largest SIM-based security infrastructure worldwide. Another important consideration is that the GSM authentication protocol is comparatively simple and our proposals can be easily transferred to UMTS, WLAN or any other technology.

If the reader is familiar with the fundamental background of GSM mobile cellular networks and, in particular, with the basics knowledge of the Subscriber Indentity Module, this chapter may be left out.

2.1.1 GSM System Architecture

The GSM mobile communication network is a cellular structured, wireless network. Each cell represents an elementary geographic area (Picocell, Microcell, Macrocell). In GSM 1.02 [rGPP01] a mobile cellular network is subdivided into three areas:

- Radio Subsystem (RSS),
- Network- and Switching Subsystem (NSSS), and
- Operation and Maintainance Subsystem (OMSS)

The functional architecture of such subsystems is illustrated in Figure 2.1, followed by a subsequent description of the depicted network components.

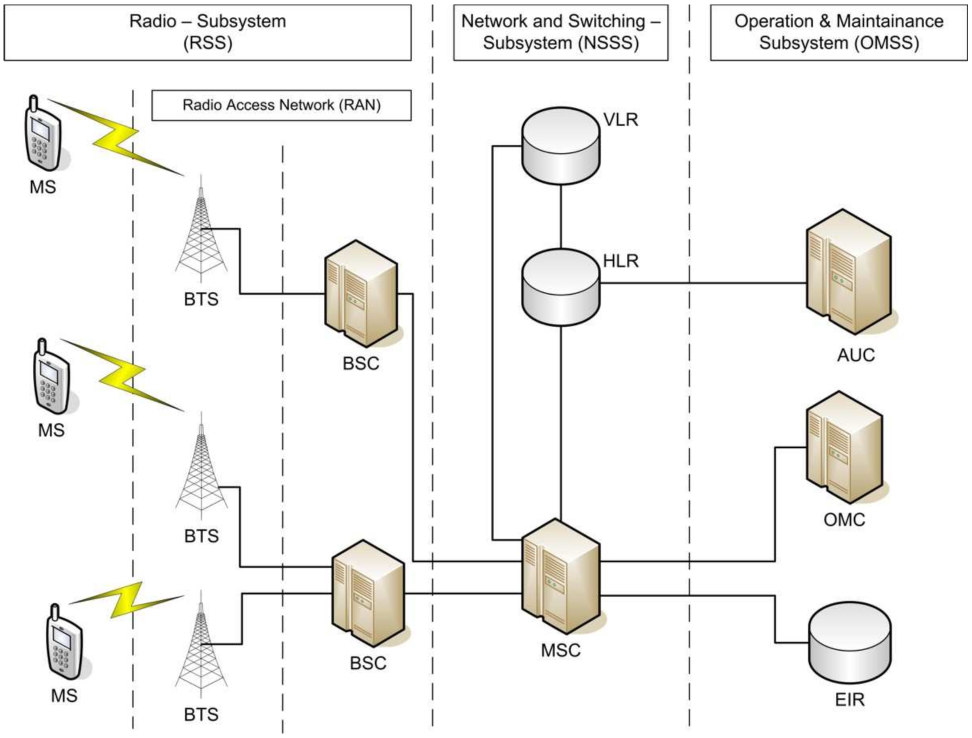

Radio – Subsystem (RSS)
Network and Switching – Subsystem (NSSS)
Operation & Maintainance Subsystem (OMSS)
Radio Access Network (RAN)
MS
BTS
BSC
MS
BTS
MS
BTS
BSC
VLR
HLR
MSC
AUC
OMC
EIR

Figure 2.1: *Functional Architecture of a GSM mobile Network*

2.1.1.1 Radio Subsystem (RSS)

The *Radio-Subsystem (RSS)* is responsible for the wireless communication within a mobile cellular network. It consists of the Mobile Station (MS) and the components of Radio Access Network (RAN).

- *Mobile Station (MS):* The *Mobile Station (MS)* is the terminal equipment of a subscriber. It is composed of the *Mobile Equipment (ME)* and the Subscriber Identity Module (SIM). If a ME additionally is equipped with an embedded hardware-based trust-anchor, as described in Section 2.2.5, it is termed as a *Mobile Trusted Platform (MTP)* throughout this thesis.

- *Radio Access Network (RAN):* A *Radio Access Network (RAN)* is formed by several *Base Station Subsystems (BSS)*. Each BSS consists of a *Base Station Controller (BSC)* with a set of associated *Base Transceiver Station (BTS)*. All radio related functionality is controlled by the BTC, which handles the protected communication between the MS and the BTS. In this context, we stress that the GSM standard only stipulate an encrypted channel between the MS and BTS, which leads to flaws and vulnerabilities to the GSM communication system [Eck04].

2.1.1.2 Network- and Switching Subsystem (NSSS)

The *Network- and Switching Subsystem (NSSS)* implements the underlying cellular switching and network communication technology. Significant components of a NSSS are the *Mobile Switching Center (MSC)*, the *Home Location Register (HLR)* and the *Visitor Location Register (VLR)*.

- *Mobile Switching Center (MSC):* The *Mobile Switching Center (MSC)* represents the central node of a NSSS and controls a set of dedicated BSCs and associated BTSs. The main task of a MSC is to set up circuit-switched connections so that data can be transferred between the communication participants. All communication are relayed and maintained across cell boundaries, so that all inbound and outbound connections are controlled, managed and monitored by this network component.

- *Home Location Register (HLR):* The *Home Location Register (HLR)* is a data repository assigned to a MSC. It stores individual information of each subscriber (e.g. IMSI, MSISDN, SST) and temporary informations (e.g LAI, MSRN) of the subscriber. In a GSM network, at least one HLR is present and every subscriber is assigned to one specific HLR.

- *Visitor Location Register (VLR):* The *Visitor Location Register (VLR)* denotes another data repository of a MSC. It contains subscriber informations, similar to the HLR, but it stores only information for external subscribers who are temporary located inside the area of responsibility of a MSC. Typically, a VLR refers to several MSCs, but one MSC always uses one VLR.

2.1.1.3 Operation and Maintenance Subsystem (OMSS)

The main tasks of the *Operation and Maintenance Subsystem (OMSS)* are administration, authorization and accounting of subscribers, and maintenance of the GSM network components.

The OMSS consists of the *Authentication Center (AuC)*, the *Equipment Identity Register (EIR)* and the *Operation and Maintainance Center (OMC).*

- *Authentication Center (AuC):* The *Authentication Center (AuC)* is an important component in the security architecture and is usually integrated as a part of the HLR. It provides information of identification and authentication of a subscriber, and precomputes authentication triplets, which are used to identify and authorize a subscriber. Such a triplet consists of a challenge $RAND$, the correct response $SRES$, and the communication key $\mathcal{K}_c$.

- *Equipment Identity Register:* The *Equipment Identity Register (EIR)* holds a black, white and grey list aiming to protect against device theft. The EIR checks the status of a MS while performing the login procedure. An device is either listed as an non-approved, valid or as monitored.

- *Operation and Maintenance Center (OMC):* The OMC acts as the operation and maintenance center in a GSM network. It is used for installing, configuring and supervising network components of a MNO.

2.1.2 GSM Security Architecture

The following subsection introduces to the GSM Security Architecture and shows how authentication and confidentiality are provided. The GSM security mechanisms are described in detail in GSM 02.09 and GSM 03.20 [rGPP97a, rGPP91].

2.1.2.1 GSM Security Algorithms

The GSM security architecture [rGPP97b] holds three algorithms to perform subscriber authentication, session key-generation and encryption of the communication channel. In respect to our primary interest, we focus on the subscriber authentication algorithm 'A3' and the key generation algorithm 'A8'. These two will build up the essential vSIM Core Algorithms from Section 3.4.1. Moreover, we introduce to the encryption algorithm 'A5' for the sake of completeness.

Subscriber Authentication Algorithm 'A3' One of the primary security functions of the SIM is the authentication algorithm A3. This procedure assures that the network login process is initiated by an authorized subscriber. Therefor, the MNO verifies the identity of a subscriber through a challenge-response protocol, using the symmetric key K_i as illustrated in Figure 2.2.

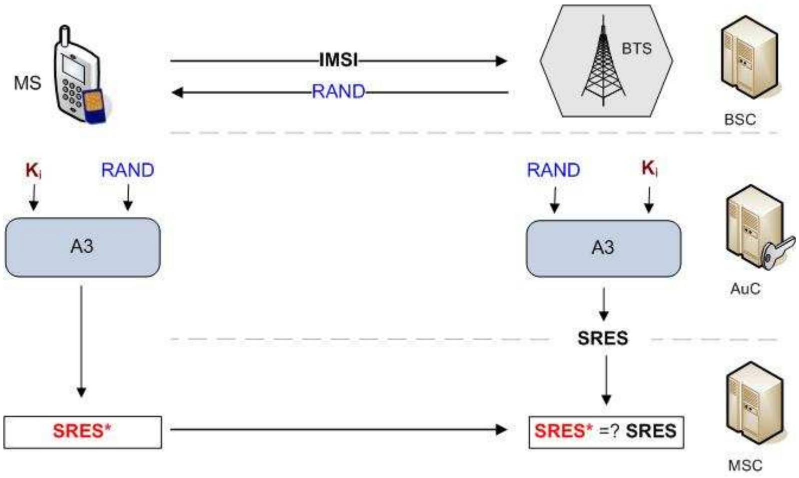


Figure 2.2: *GSM Challenge Response Authentication*

Once a local operator has successfully authenticated itself to the MS by a given authentication data (PIN), the MS executes the *GSM Auth Algorithm* command and sends a unique identifier (IMSI, TMSI) to the MSC of an available GSM network. The MSC responses with a cryptographic challenge, which consists of a 128 bit number called $RAND$, which was extracted from a triplet received from the AuC. When the MS receives $RAND$, it relies it to the SIM for computing the answer, called 'Signed RESponse ($SRES$)'. The SIM feeds its A3 algorithm with the RAND and the secret key K_i to produce a 32-bit $SRES^*$. It is transferred out of the SIM to the ME, where it is then transmitted to the MSC. After the MSC has received

$SRES^*$ from the ME it compares the value with the $SRES$ either received from the triplet or directly (using the AuC). If the two values are equal, the network assumes the MS as legitimated and grant network access. If the two values are different, the MSC denies service access to the MS.

An important fact is that a subscriber's secret key K_i is never transmitted over the network, because it is always independently computed in both, the SIM and the AuC.

GSM Key Generation Algorithm 'A8' The second algorithm of a SIM is used for generation of the session key K_c. This key is used later by A5 in order to protect the communication between the MS and the BTS. K_c is an 64 bit session key, which is computed by the steps illustrated in Figure 2.3.

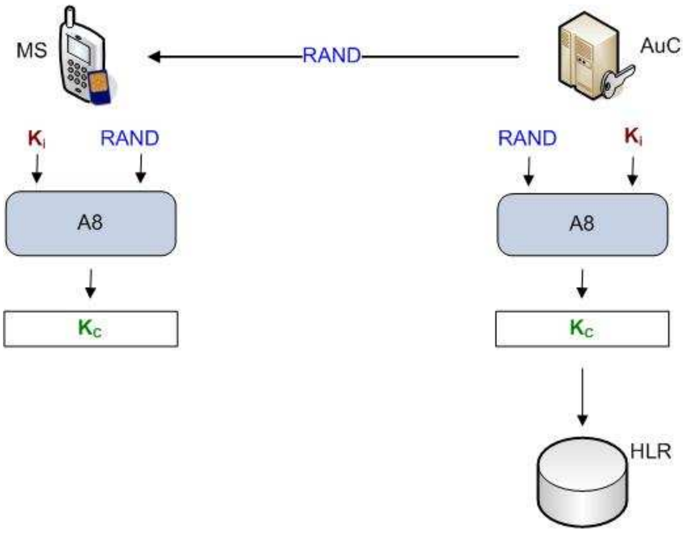


Figure 2.3: *GSM Key Generation Schema*

An identical symmetric communication key K_c is used on both sides, the MS and the BTS. Therefor, the AuC sends a $RAND$ to the SIM. It also generates K_c and includes the key into the authentication triplet, as detailed above. At the MS side, the K_c is stored by MS until it is updated at the next authentication. Usually, the K_c is used by the participants for protecting several consecutive sessions.

In practice, A3 and A8 are implemented in combination by the MILENAGE algorithm [rGPP02], also called for short A38. Using the $RAND$ and the secret key K_i, the SIM executes this algorithm to produce K_c and $SRES^*$.

GSM Encryption Algorithm 'A5' The encryption algorithm A5 is used to protect both, signaling- and communication data between the MS and the BTS. It is a stream cipher based on three clock-controlled *Linear Feedback Shift Register (LFSR)*'s using a ciphering key K_c. The encrypted data stream is obtained by a logical xor operation of the input data and a ciphering bit stream. The detailed cipher algorithm A5 is described in [rGPP06]. The algorithm is initialized with the generated session key K_c from above and the current frame number. Figure 2.4 illustrates A5.

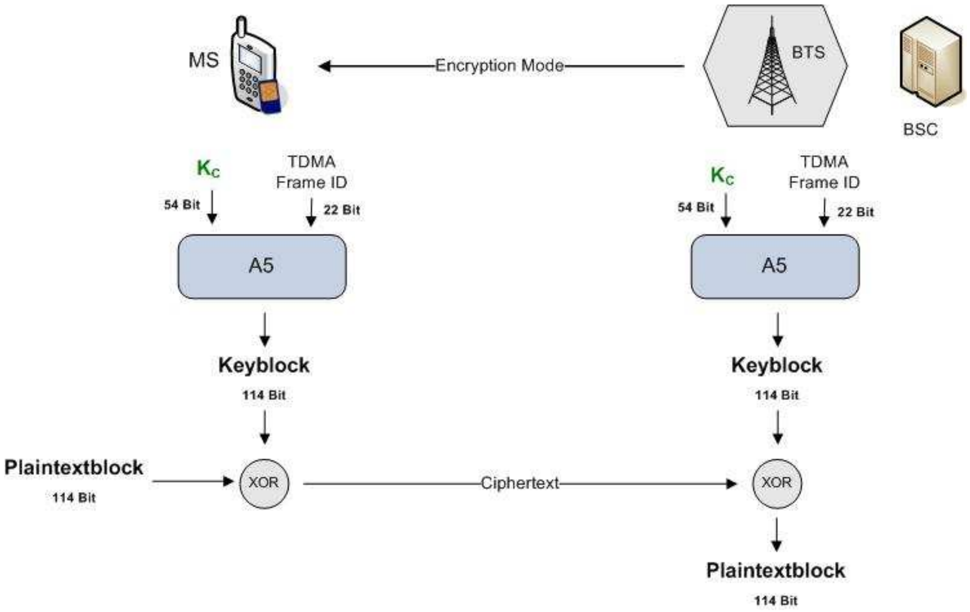


Figure 2.4: *GSM Encryption Schema*

2.1.3 The Subscriber Identity Module (SIM)

A Subscriber Identity Module (SIM) is the established means for subscriber authentication in GSM-900/1800 mobile cellular networks. Typically, it is a removeable smart card based on a embedded integrated circuit chip. It holds at least the subscriber credential, as well as the authentication algorithms A3/A8, and provides protected storage and protected execution functionality for security-sensitive data.

2.1.3.1 SIM Security and Usage Characteristics

In our further discussion and resulting comparison with a vSIM, we have to look at the following five significant characteristics. In general, a SIM Card is characterized by a (1) protected and isolated execution environment, (2) protected storage functionality, (3) strong authentication, (4) portability and mobility, and (5) preallocation of subscriber credentials.

1. *Protected and Isolated Execution Environment:* One of the fundamental characteristic of a SIM is to provide a protected execution environment, where application code and data can be processed without being exposed. This implies a tamper resistant and secure execution environment having an adequate protection to detect and prevent from hardware and software attacks [Sma01].

2. *Protected Storage:* A SIM holds a tamper resistant environment for secret keys and data. All keys are stored inside the protected storage, which is inaccessible by the environment outside of the SIM.

3. *Strong Authentication:* The authentication process in GSM networks requires at least two factors, namley 'Something you know' (PIN) and 'Something you have' (MS, SIM). This is performed without transmitting the secret authentication data over the network.

4. *Portability and Mobility:* A SIM has to be removeable and portable to other devices. Hence, it enables a subscriber to use its credential with an arbitrary device, and vice versa, if no constriction is imposed by the MNO.

   The GSM standard explicitly differentiates between device and subscriber identity. A MS and a subscriber have two different unique identifiers with different intended usage characteristics.

5. *Preallocation of Subscriber Credentials:* During the manufacturing process the individualization of a SIM is done. Within this process, a SIM is equipped with Subscriber Credentials including the secret identification key K_i and the unique identifier $IMSI_i$.

Hence, it holds suitable techniques to prevent from unauthorized access of security-sensitive data (e.g. K_i, K_c, LAI), and unauthorized manipulation of protected data (e.g. K_i, IMSI, SST, CHV). This security characteristics correspond with the requirements defined in Section 1.2. The first three items can be mapped to the security benchmark (1) and item *Portability and Mobility* is mapped to benchmark 4 directly. However, the aspired solution does not require the property *Preallocation of Subscriber Credentials* mentioned above.

2.1.3.2 SIM File System and Structure

A SIM Card implements a hierarchical file system for storage of subscriber and network information. A *Master File (MF)* is located at root of this file structure, that holds all the other files in a hierarchical structure. *Dedicated folders (DF)* group *Elementary Files*. GSM defines two levels, directly under the MF. The DF_{GSM} contains GSM application files and $DF_{TELECOM}$ contains application service files.

Elementary Files (EF) contains the stored information, which are available within a SIM card. These files are either transparent, linear fixed, or cyclic. Transparent EF's are binary files that can store information of varying length. Linear fixed EF stores information of a fixed length. Finally, a cyclic EF stores the information consecutively using overwritable fixed-length data records.

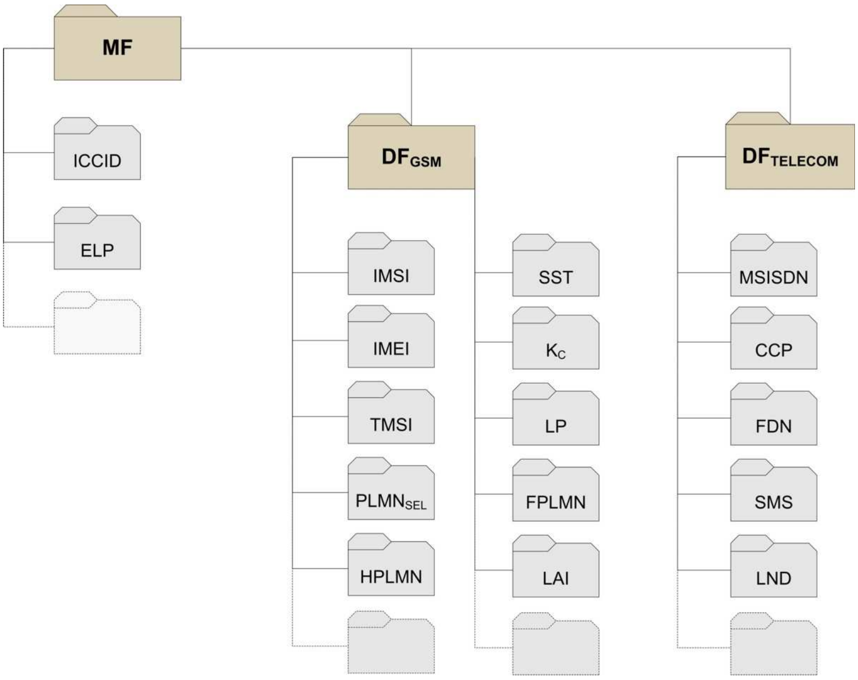


Figure 2.5: *SIM File-Hierarchy*

2.1.3.3 Contents of the Elementary Files

The next paragraphs introduce to mandatory elements of the file system compliant to GSM standard.

Contents at the MF **level** There are two EFs at the MF level. The *ICC identification (ICCID)* is an unique identification number for a SIM card. With this identifier, a SIM card represents the digital identity of a subscriber. The *Extended language preference (ELP)* contains codes of the preferred languages of a subscriber.

Contents of Files at the GSM Application Level The EFs in the Dedicated File DF_{GSM} contain network related information such as:

The *International Mobile Subscriber identity (IMSI)* is a unique identifier of a subscriber within the GSM system. The IMSI is normally never transmitted over the air interface in cleartext. The *International mobile equipment identity (IMEI)* is a unique device number of the mobile station. This value is stored by the mobile station and in the EIR.

The *Temporal Mobile Subscriber Identity (TMSI)*, in conjunction with the location area information represents the temporally unique subscriber identity. The TMSI and the LAI are assigned by the VLR. Thus, it provides pseudonymity of a mobile station and a subscriber in order to prevent from being unauthorized traced.

The *Session Cipher Key (K_c)* is a secret key used by the stream-cipher A5. K_c is used for encrypting data transmitted between MS and BTS via the radio-interface.

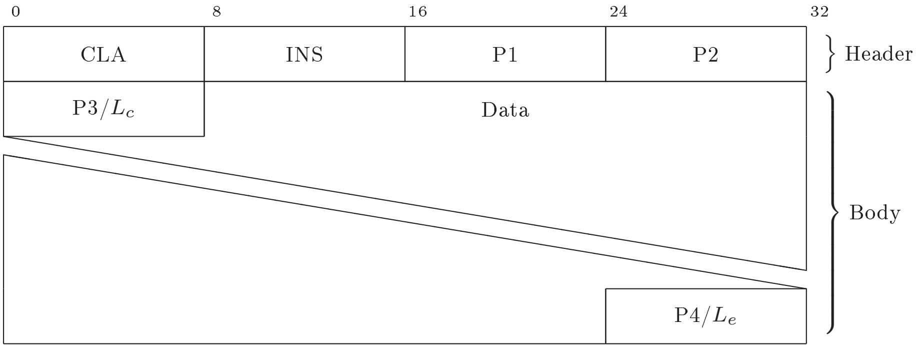
Similar to the ELP, the *Language Preference (LP)* contains the a set of language codes. This information, are determined by the subscriber and defines its preferred languages. The *PLMN Selector* $PLMN_{sel}$ contains the available codes of the *Public Land Mobile Network (PLMN)*s. This information is set by the subscriber and defines its the preferred PLMNs. The *HPLMN search period (HPLMN)* holds the update interval between searches for the HPLMN. The *SIM Service Table (SST)* contains a table of available services that can be used or enabled by an subscriber. The *Forbidden PLMNs (FPLMN)* include the codes of forbidden mobile cellular networks, wherein service access and roaming is not permitted.

Contents of Files at the Telecom Level: The EFs in the Dedicated File $DF_{TELECOM}$ contain service related information. The *Mobile Station ISDN Number (MSISDN)* is the dialing number of the mobile station. *Capability Configuration Parameters (CCP)* store parameters of capabilities of the mobile cellular network and the ME. The *Fixed Dialling Numbers (FDN)* contains dialling or service numbers. The *Short Messages Service (SMS)* contains data in accordance with GSM 03.40, which have been received or send by the MS. The *Last Number Dialled (LND)* stores the last dialled numbers or requested services.

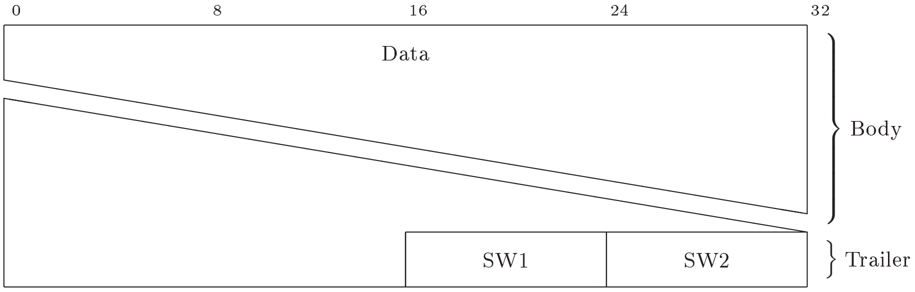
2.1.3.4 SIM I/O Communication

The GSM ETSI 11.11 [rGPP97b] specifies the ME communication between the ME and the SIM in accordance to the ISO 7816-3 standard. It utilizes *Application Protocol Data Units (APDU)*, which are sent across the interface between the SIM and the ME. An APDU can be a command APDU or a response APDU. A command APDU has the following general format:

APDU Command Messages APDU command messages consists of five data fields. The first field embeds a class of instruction (CLA). The CLA is always set to A0 in GSM networks. The instruction code (INS) indicates the particular commands as listed in Table 2.1. $P1$, $P2$, $P3$ and $P4$ are parameters for the command. $P3/L_c$ and $P3/L_e$ contain the length of the outgoing and expected incoming data segment respectively.



APDU Response Messages The response messages is returned in three data fields of a APDU. The data field has a length of $P4/L_e$ and contains information requested by a command. This is followed by the return code that consists of two status bytes $SW1$ and $SW2$, indicating the success or failure of the requested command.



2.1.3.5 SIM Commands

In context of the present thesis, we are interested in two related GSM specifications, namely the GSM11.11 [rGPP97b] and GSM11.14 [rGPP04]. The GSM 11.11 specification defines the operational commands for GSM SIM cards which are are categorized into File Operation Commands, Security Commands, and Miscellaneous and obsolete Commands. In Table 2.1 an overview of such commands are given. With the additional GSM 11.14 SIM Application Toolkit specification a SIM card can be accessed for administration and maintenance by a remote MNO. These commands are shown in Table 2.2.

Command	**Ins**	**P1**	**P2**	**P3**	**Short Description**
File Operation Commands					
SELECT	A4	00	00	02	Selection of a EF
STATUS	F2	00	00	len	Returns information concerning the current directory
SEEK	A2	00	t/m	len	Searches for a pattern through the current linear fixed EF
INCREASE	32	00	00	03	Increases counter value of the current EF
INVALIDATE	04	00	00	00	Invalidates the current EF
REHABILITATE	44	00	00	00	Rehabilitates the invalidated current EF
READ BINARY	B0	high	low	len	Reads a string of bytes from the current EF
UPDATE BINARY	D6	high	low	len	Updates the current transparent EF with a string of bytes
READ RECORD	B2	#rec	mode	len	Reads a complete record in the current EF
UPDATE RECORD	DC	#rec	mode	len	Updates one complete record in the current EF
Security Commands					
VERIFY CHV	20	00	#CHV	08	Verifies the CHV presented by the User
CHANGE CHV	24	00	#CHV	10	Assigns a new value to the relevant CHV
ENABLE CHV	28	00	01	08	Enables the enquiry of a CHV

Command	**Ins**	**P1**	**P2**	**P3**	**Short Description**
DISABLE CHV	26	00	01	08	Disables the enquiry of a CHV
UNBLOCK CHV	2C	00	00/02	10	Unblocks a CHV which has been blocked by three consecutive wrong CHV presentations
RUN GSM ALGORITHM	88	00	00	00	Procedure for authenticating the SIM to a GSM network (A3/A8)
Miscellaneous and Obsolete Commands					
SLEEP		00	00	00	Obsolete command for putting the smart card into a low-power state
GET RESPONSE		00	00	00	Command specific to T=0 for requesting data from the smart card

Table 2.1: *GSM 11.11 SIM Commands*

The GSM 11.14 standard details the *SIM Application Toolkit (SAT)*. It allows the remote-execution of specific applications of a MNO.

Command	**Ins**	**P1**	**P2**	**P3**	**Short Description**
SIM Application Toolkit Commands					
ENVELOPE	C2	00	00	00	Transfers a SIM Application Toolkit command from the ME to the SIM
FETCH	12	00	00	00	Transfers a SIM Application Toolkit command from the SIM to the ME
TERMINAL PROFILE	10	00	00	00	Transmitting of ME capabilities to the SIM Application Toolkit
TERMINAL RESPONSE	14	00	00	00	Transfers the ME response of a previously fetched SIM Application Toolkit command to the SIM

Table 2.2: *GSM 11.14 SIM Application Toolkit Commands*

2.1.3.6 File Access Control

Each elementary file has four individual attributes that indicate the access condition by 16 different states, which are numbered from 0 to 15 in increasing order of security. Each attribute is dedicated to a specific file operation command, namely the commands READ, UPDATE, INVALIDATE or REHABILITATE. The different *Access Control Conditions* are described in the Table 2.3.

Level	**Access Condition**	**Description**
0	Always	The access condition "Always" means that the file may allways be accessed by the associated command.
1	CHV1	Files with the access condition "CHV1" (Card Holder Verification #1) are only accessible after a successfully verification of CHV 1 using the first subscriber authentication data (PIN number).
2	CHV2	Access is only allowed after a successfully verification of CHV 2 using a second subscriber authentication data (PUK number)
3	Reserved	
4 - 14	ADM	The access conditions 4-14 are only available for administration by the MNO. The definition of access condition ADM does not preclude from "ALWAYS", "CHV 1", "CHV 2" and "NEVER" if required.
15	Never	A file can never be accessed using the associated command over the SIM/ME interface. It may be accessed only internally by the SIM Card.

Table 2.3: *File Access Conditions*

2.2 Trusted-Computing on Mobile Platforms

This section provides the reader with the essential fundament of the Trusted Computing Technology. We shortly describe the basic architecture and functionality of a trusted computing platform and its particularities in the mobile domain.

It is organized as follows. In Subsection 2.2.1, we initially introduce the basic terminology. Furthermore, the generic TPM architecture is discussed in Section 2.2.2. Based on this architecture, we will inspect the offered features and functionalities with regard to our objective in Subsection 2.2.3. Finally, in Subsection 2.2.4 we explore the significant parts of the *TCG MPWG Reference Architecture*. This is subdivided into four more parts which detail and concretize the concepts and the requirements of the TCG MPWG architecture.

For more comprehensive information about the fundamentals of Trusted Computing, we recommend the books “Trusted Computing” [Mit05], “IT-Sicherheit” [Eck06] or the official TCG Specification Website [TCG]. The latest information about Trusted Computing on mobile platforms, and especially the TCG MPWG specification are available online at TCG MPWG website [Tru06b].

2.2.1 Trusted Computing Terminology

Basically, the specialized meaning of *Trust* is, that an entity always behaves in the expected manner for the intended purpose. In this context, *Trusted Computing* terms a technologies for increasing trust in computer platforms based on a small-scale embedded trust anchor building the foundation of that trust.

The TCG calls this trust anchor a *Trusted Platform Module (TPM)*, which is used to securely store and use asymmetric keys and is able to provide evidence about its and the platform’s trustworthiness. With regard to mobile devices such a chip is called Mobile Trusted Module. The TPM/MTM, in collaboration with a capable software service [Tru06c], can provide evidence to a third party about the trustworthiness of itself and the host platform.

A *Trusted Computing Platform* is a computing platform that has an embedded trusted component, in form of such a TPM/MTM, which it uses to create a foundation of trust. It runs a *Trustworthy Operating System* (e.g. EMSCB/Turaya [EMS], IBM sHype [SVJ$^+$05]), which implements a security architecture. Typically, such a platform is built upon a small manageable, stable and evaluable security kernel for TC-enabled hardware platforms such as standard desktop PCs, servers, embedded systems and mobile devices. It provides fundamental security mechanisms and a protected and isolated execution environment.

2.2.2 TPM Architecture

The TPM supplies so-called protected capability and shielded locations. Protected capability identifies a set of commands with exclusive permission to access shielded locations. These locations are tamper-resistant and protected areas to operate on security-sensitive data. Both, shielded locations and protected capabilities are implemented within the TPM and therefore they are robust against software attacks.

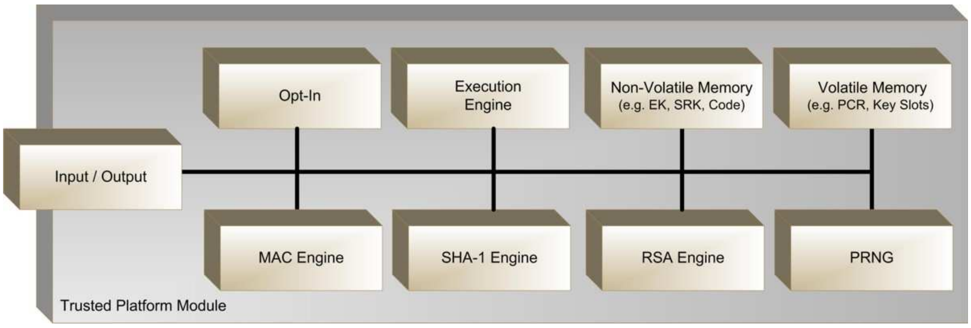


Figure 2.6: *TPM Architecture*

The main components of a TPM are illustrated in Figure 2.6. The functions can be arranged in cryptographic components, non-cryptographic components and memory components:

2.2.2.1 Non-Cryptographic Components

The *I/O Component* is the communication interface to a TPM. It implements a *Security Controller*, that enforces security policies and routes and forwards the messages to target components and handles encoding and decoding on the external and internal bus. The *Execution Engine* executes the commands received through the I/O component.

2.2.2.2 Cryptographic Components

The TPM has a number of built-in cryptographic functions. The *RSA Engine* is a implementation of the RSA-algorithm within the TPM. The implementation must conformant to the PKCS#1 [JK03] and has to support key lengths are 512, 1024, and

2048 bits. The *SHA-1 Engine* provides functionality to compute a collision-resistant compressed digital representation of the input value, called SHA-1 message digest with the length of 160 bits, according to the FIPS-180-1 standard [Nat02]. The *HMAC-Engine* is used for computation of a keyed-hash-value. It is used as proof of data integrity and authentication of input and output data as described in RFC2104 [KBC97]. The *Pseudo Random Number Generator* generates, in cooperationwith the key generation capability, asymmetric and symmetric keys, as well as nonces to provide freshness.

2.2.2.3 Memory Components

A TPM stores its persistent data within a *Non-Volatile Memory.* It holds the *Endorsement Key (EK)*, *Storage Root Key (SRK)*, *Owner Authorization* data and persistent flags. Moreover, this memory has a set of *Key Slots* caching and activating RSA keys within the TPM.

The data in the *Volatile Memory* is only available during power activity. It will be lost and reseted when the device is switched off or rebooted. An important part of this memory type are Platform Configuration Registers (PCR). PCR's store platform integrity metrics as 160-bit SHA1 values in at least 16 discrete registers. In Table 2.4 the PCR's allocation of a Mobile Trusted Platform is shown.

PCR Index	**Description**
PCR_0	All relevant characteristics of the hardware platform configuration.
PCR_1	All measurements pertaining to the RTE are to be measured into PCR1
PCR_2	RTS+RTR and RTV+RTM block of the TE_{DM} is to be measured into PCR2.
PCR_3	Reserved for engine load events for the TE_{DM}
$PCR_4 - PCR_6$	Reserved for DM proprietary measurements
PCR_7	Reserved for measurements of Trustworthy Operating System (of TE_{DM})
$PCR_8 - PCR_{12}$	Reserved for DM proprietary measurements
$PCR_{13} - PCR_{16}$	Unallocated at present

Table 2.4: *Allocation of PCRs*

2.2.3 Core Features and Functionality

This subsection inspects the core features and functionalities of a Trusted Computing Platform:

- *Integrity Measurement*,
- *Platform Attestation*,
- *Binding, Sealing, Signing and Sealed-Signing*, and
- *Protected Storage*,

2.2.3.1 Integrity Measurement

The integrity measurement requires recording and measuring the platform state at the beginning of system boot process. For this, the concept of trusted boot (or authenticated boot) is introduced. The anchor of this process is an initial measurement program code of the TPM, called the *Core Root of Trust for Measurement (CRTM)*. It starts and measures itself. This initial integrity value is saved in a log and extended to the PCR of the TPM. Afterwards, the CRTM loads and measures the BIOS. The resulting integrity measurement is also stored and logged before the BIOS starts. This process of measurement, reporting, and logging is iterated by each component. The trusted operating system finally performs these tasks for every software component loaded. The TCG calls this *Transitive Trust* or *Chain-of-Trust*. Basic principle of this process is that each trusted component measures its successor before passing the control to it. A measured integrity metric is reported to the TPM by extending it to the PCR. Therefore, the TPM applies the SHA-1 hash algorithm on the input value. The PCR extend operation works as follows:

$$PCR_N \quad \leftarrow \quad H(PCR_N \parallel IM_C)$$

A measured integrity metric (IM_C) of a platform component C is concatenated with the previous PCR_N hash value. The SHA-1 algorithm repeats its execution on this result and stores the new value as a new one into PCR_N. This register value holds the current state S_i of the platform (or platform subsystem; see Table 2.4) that is represented by a specific PCR_i register and can be requested by an external verifier.

2.2.3.2 Platform Attestation

Proving the current platform state to a local or remote verifier is done by attestation. It provides evidence that the device is unchanged and the platform is in a trustworthy configuration. This attestation process is abbreviated by the following term, which is detailed below.

$$ATTEST(S_i) \quad \leftarrow \quad SIGN_X(NONCE \parallel S_i), LOG$$

An idealized process of the platform attestation is illustrated in Figure 2.7. At the beginning of this process, the verifier generates a random value $NONCE$ and sends it to the TPM of the corresponding trusted computing platform (Step 1). The platform feeds the TPM with the received value (Step 2) and instructs it to proof that it is a genuine TPM and testify what it has seen during the trusted boot process. The TPM chooses an eligible PCR value holding the current state and signs it with an appropriate attestation identity key AIK (Step 3). Afterwards, the resulting signature is sent back to the verifier, which checks the response against reference values (Step 4,5).

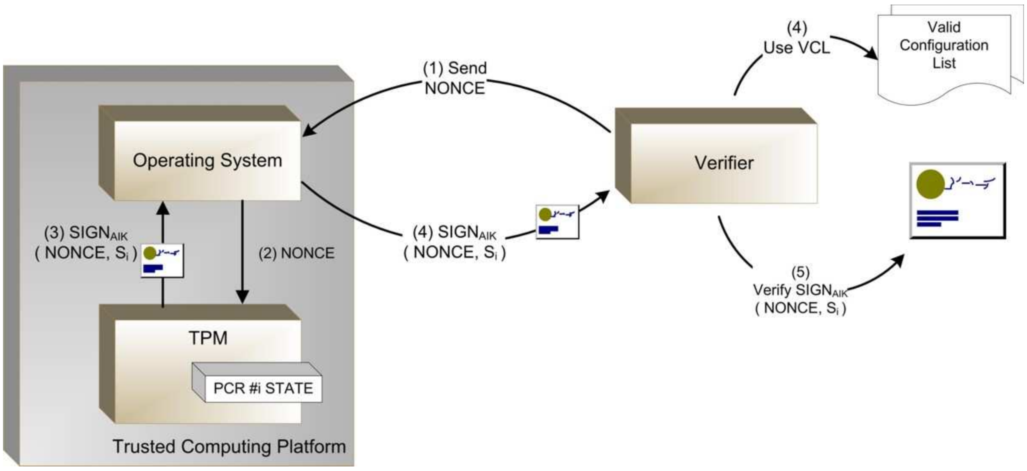

(1) Send NONCE
(4) Use VCL
Valid Configuration List
Operating System
Verifier
(3) SIGN_AIK (NONCE, S_i)
(2) NONCE
(4) SIGN_AIK (NONCE, S_i)
(5) Verify SIGN_AIK (NONCE, S_i)
TPM
PCR #i STATE
Trusted Computing Platform

Figure 2.7: *TCG Platform Attestation (simplified)*

2.2.3.3 Signing, Binding, Sealing and Sealed-Signing

The TCG defines the four message classes for data protection:

- *Binding*,
- *Sealing*,

- *Signing*, and
- *Sealed-Signing.*

In the next paragraphs, the reader will be introduced to these four protection classes. For each class, we will give a specific nomenclature that we will take up later in the protocol description of Chapter 3.

Binding: The binding functionality is used to encrypt and decrypt a message using the public portion of a binding key. A binding key BK_X is a migrateable or non-migrateable key that is managed by the TPM. In case of a non-migratable private key, only the creator of the key is able to use it. For this reason, the encrypted data is bound to the platform and can only be used by the associated TPM. If a binding key is migratable, it is transferable between multiple TPM devices and has no special significance beyond encryption. The sender uses the binding key of the intended recipient X to encrypt the message.

$$[Data]_X \quad = \quad BIND(Data, BK_X^{pub})$$

In case of using migrateable keys for encryption, we use either asymmetric key-pairs or symmetric keys. If a symmetric key is used, we associate the computation with an key-hashing algorithm (HMAC). Therefore, an additional mechanism for integrity protection is provided.

$$[Data]_X \quad = \quad ENC(Data, BK_X)$$

The encrypted data blob is recoverable by decrypting it using the unbind functionality of a TPM. This command decrypts a blob with an associated private binding key that is stored inside the TPM.

$$Data \quad = \quad UNBIND([Data]_X, BK_X^{priv})$$

Sealing: The sealing functionality of a TPM encrypts a message using a public key with additional chaining it to a set of platform integrity metrics. The encrypted data is associated to a set of PCR values and a non-migratable binding key BK_X. The PCR values represents a specific state S_0 of the computing platform.

$$[Data]_{S_0,X} \quad = \quad SEAL(Data, S_0, BK_X^{pub})$$

The TPM only decrypts the data successfully when the current platform state S_i matches to the initial state S_i.

$$Data \quad = \quad UNSEAL([Data]_{S_0,X}, S_i, BK_X^{priv}) \wedge (S_i = S_0)$$

Signing The signing functionality of a TPM ensures integrity and non-repundation of a message. Integrity protection ensures that a message was not modified. Also non-repudiation guarantee that a transferred message can not later be denied by the originating entity. In order to prevent misuse, signing of a message digest is only possible with signing keys SK_X.

$$SIGN_X(data) \quad \leftarrow \quad SIGN(H(data), SK_X^{priv})$$

Sealed-Signing The sealed-signing functionality of a TPM is used to seal a message digest using a signing-only private key and chaining it to the a range of platform integrity metrics. Therefore, the signing operation is also linked to a set of PCR values to guarantee that the platform that signed the message meets a required configuration requirement.

$$SIGN_{X,S_i}(data) \quad \leftarrow \quad SIGN(H(data), SK_X^{priv}, S_i)$$

The verifier can then check the platform state S_i supplied in the signature, which is equivalent to inspecting the signing platform's configuration at the time the signature was generated.

2.2.3.4 Protected Storage

The TPM provides potentially unlimited amount of storage for protected objects such as signature keys, storage keys, binding keys or generic data (e.g. symmetric keys, authentication data, arbitrary security-sensitive data). This is attained with an implementation of a key hierarchy, where only keys are stored under the protection of the physically tamper-resistant hardware. The TCG specifies two types of keys: *Non-migrateable Keys* are generated inside the TPM and never leaves this environment. *Migratable Keys* are generated either inside or outside of the TPM and can be exported. Each key in the hierarchy is encrypted using the public portion of the parent storage key. The root of this hierarchy is called the Storage Root Key (SRK). It is stored and protected within the non-volatile memory of TPM/MTM.

Because of memory limitation, inactive keys will be encrypted and moved to external storage. If a specific key has to be available, the TPM/MTM requests that key from the external storage and loads it into the key-slots of its volatile memory.

An example of such a protected storage is illustrated in Figure 2.8. The public portion of the SRK is used to encrypt the private part of the storage key. These storage keys are used to encrypt the subordinated private keys and security-sensitive data. -

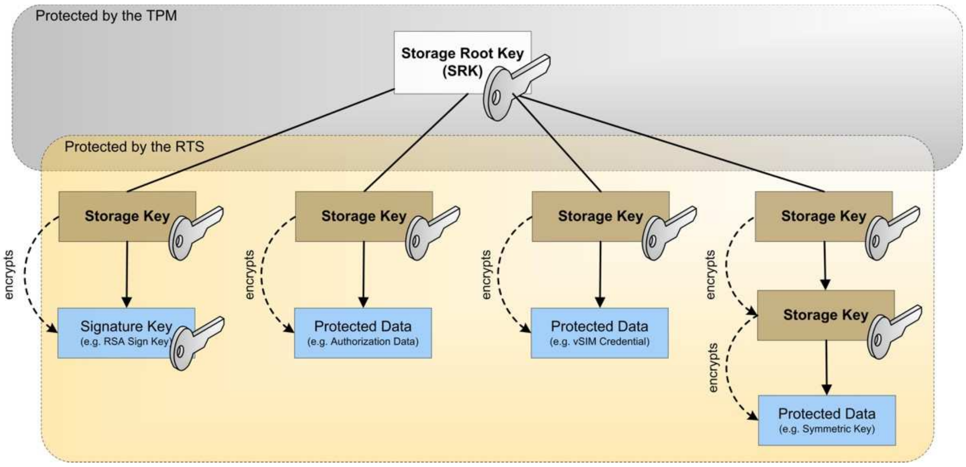


Figure 2.8: *TPM Protected Storage*

An important feature of *Protected Storage* is that protected objects can be sealed to a specific platform configuration, as described in paragraph 2.2.3.3. This is done during the creation process of the protected object by indicating the required platform state that must exist if the data is to be revealed.

2.2.4 TCG MPWG Reference Architecture

The TCG MPWG has developed an architecture on a high level abstraction for a trusted mobile platform, which offers numerous opportunities of variation in design and implementation. In this section, we reflect essential parts of this architecture and concretize significant platform components in terms of our objective.

An important aspect of the *TCG Mobile Reference Architecture* is the potential to virtualize significant parts of a trusted mobile platform as trusted software applications and services. The trusted execution chain is built on a Mobile Trusted Module (MTM). The implementation of this chip depends on the security requirements of its specific use-case. In order to reach a high-level of protection and isolation, a MTM could be implemented as a slightly modified Trusted Platform Module (TPM).

A trusted mobile platform is characterized as a set of multiple tamper-resistant engines, each acting on behalf of a different stakeholder. Broadly, such an platform has several major components: trusted engines (TE), trusted services (TS) customized by trusted ressources (TR). A generalized trusted mobile platform is illustrated in Figure 2.9.

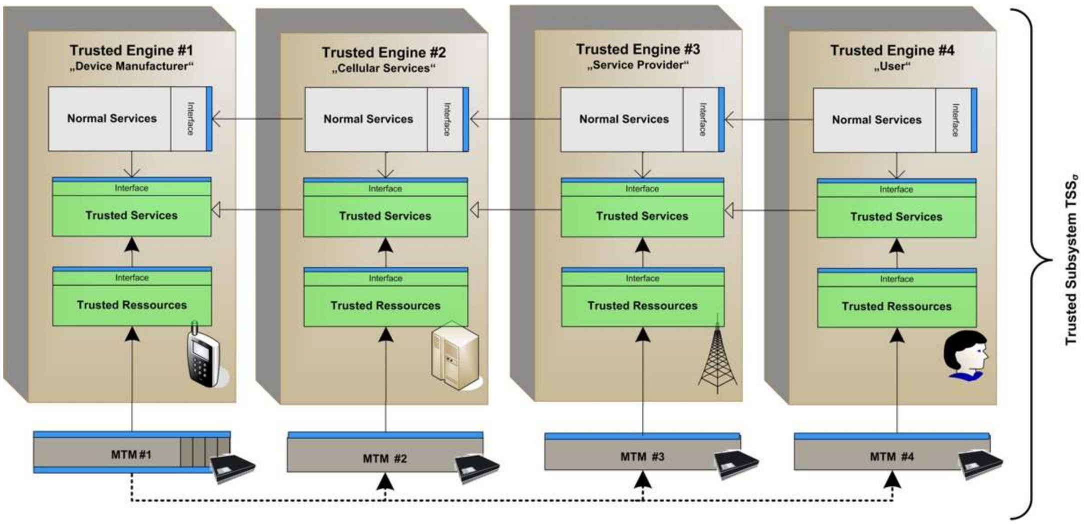


Figure 2.9: *Trusted Mobile Platform Architecture*

We group a *Trusted Subsystem (TSS)* as a logical unit of a trusted engine with its interrelated hardware compartment. A TSS of a stakeholder σ can formally described by an tuple

$$TSS_\sigma \quad = \quad \{TE_\sigma, TS_\sigma, TS_\epsilon, TR_\sigma, SP_\sigma, SC_\sigma\}$$

In each trusted subsystem TSS, either a remote or local entity acts as a stakeholder, who is able to configure its own subsystem. A stakeholder defines its security policy SP_σ and system configuration SC_σ within an isolated and protected environment. The *TCG MPWG Reference Architecture* specifies following principle entities: the local stakeholders *Device Owner (DO)* and *User (U)*; and the remote stakeholders *Device Manufacturer (DM)*, and more general *Remote Owners (RO)* (e.g. Communication Carrier, Service Provider).

The functionality of a TSS either is based on dedicated ressources of an embedded engine TE_σ, or may provided by trusted services TS_ϵ of external engines.

Each subsystem is able to enforce its security policy SP_σ and subsysten configuration SC_σ. As a consequence, the functionality of a trusted subsystem TSS_σ is constrained by the available resources TR_σ with its derived trusted services TS_σ, by the offered functionality of external trusted services TS_ϵ, by the security policy SP_σ and given system configuration SC_σ of a engine's stakeholder.

All internal functionality of a TSS_σ are isolated from other subsystems by the underlying security layer. It is only accessible, if a proper service interface is defined and exported. A TSS_σ relies on the reputation of the stakeholder σ as basis for that trust. Therefor, each stakeholder issues a security policy SP_σ and a set of credentials, belonging to embedded trusted components of its subsystem TSS_σ. This contains reference measurements, quality assertions and security-critical requirements.

2.2.4.1 Trusted Engines

The most important concept within the *TCG MPWG Reference Architecture* is that of *Trusted Engine (TE)*s. The purpose of a trusted engine is to provide confidence in all its embedded services, which are internally or externally provided by the engine. It is a protected entity on behalf of a specific stakeholder that has special abilities to manipulate and store data, provide evidence of its trustworthiness and the current state of the engine. Figure 2.10 shows a generic trusted engine. In general, each engine has at least following abilities:

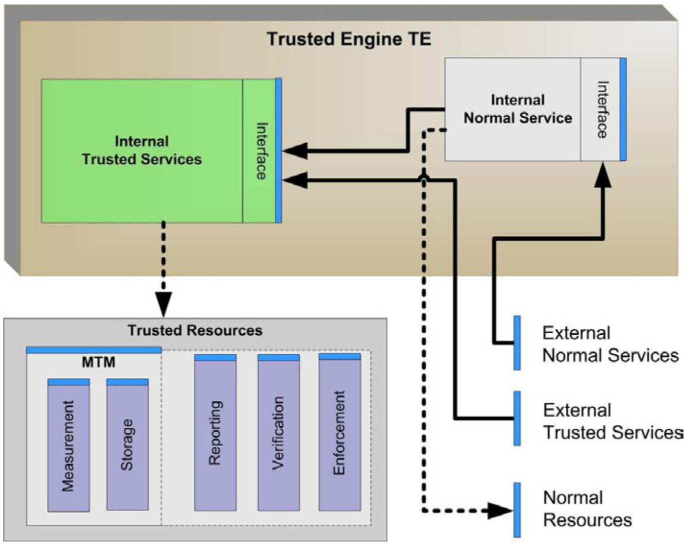


Figure 2.10: *Generic Trusted Engine*

- implement arbitrary software functionalities as trusted and/or normal services,
- provide the evidence for its trustworthiness,

- report the evidence of its current state,
- obtaining and using Endorsement Keys (EK) and/or Attestation Identity Keys (AIK),
- access a set of trusted ressources, and
- import and/or export services, shielded capabilities and protected functionality.

In order to undertake a definite categorization, the TCG MPWG differentiates engines according to their functional dispensability.

Mandatory and Discretionary Engines Therefore, an engine is either dedicated to a mandatory (of *DO* or *DM*) or a discretionary domain (of *DO*). Engines inside a mandatory domain are permanently located on a trusted platform and holding security-critical and essential functionality. All essential services of a trusted mobile platform should be located inside the mandatory domain, which does not permit a local stakeholder to remove a remote owner from the engine. Mandatory engines have access to a *Mobile Remote owner Trusted Module (MRTM)* to guarantee that a valid and trustworthy engine' state is always present. Non-essential engines and services are replaceable by the device owner *DO* and should be located inside the discretionary domain. Engines inside the discretionary domain are controlled by the device owner *DO*. Discretionary engines are required to be supported by a *Mobile Local owner Trusted Module (MLTM)*.

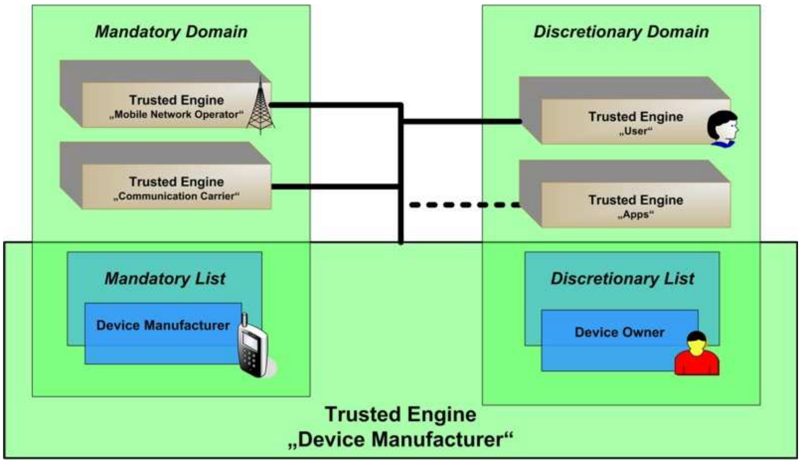


Figure 2.11: *Mandatory and Discretory Engines*

2.2.4.2 Trusted Ressources and RoTs

As illustrated in Figure 2.10, an internal trusted service has access to several trusted ressources. The TCG calls this ressources *Root-Of-Trusts (RoT)* representing the trusted components acting on base of the trusted execution chain and provide functionality for measurement, storing, reporting, verification and enforcement that affect the trustworthiness of the platform. The following RoTs are defined for the mobile domain:

- *Root of Trust for Storage (RTS)*
- *Root of Trust for Reporting (RTR)*,
- *Root of Trust for Measurement (RTM)*,
- *Root of Trust for Verification (RTV)*, and
- *Root of Trust for Enforcement (RTE).*

Each RoT vouches its trustworthiness either directly by supplied secrets (EK, AIK) and associated credentials, which are only accessible by authenticated subjects of the stakeholder, or indirectly by measurements of other trusted resources. These resources are only mutable by authorized entities of a stakeholder.

We group several logical self-contained RoTs to avoid dispensable interfaces and communication. In a typical arrangement, the RTS and RTR represent one unit, while the RTM and RTV build another unit within an TSS_σ. However, we note that the RTV and the RTM depend on protected storage mechanisms, which are provided by the RTS. So, that is also plausible, to implement all RoTs together as a common unit within an engine.

Root of Trust for Storage and Reporting The RTR and RTS are the trusted resources that are responsible for secure storage and reliable reporting of information about the state of trusted mobile platform. A RTS provide PCRs and protected storage for an engine and stores the measurements made by the RTM, cryptographic keys, and security sensitive data. A RTR signs the measurements with cryptographic signing keys of TSS_σ.

Root of Trust for Measurement and Verification In general, a RTM is a reliable instance to measure platform components and provide evidence of the current state of a trusted engine and its embedded services. The mobile domain, this functionality is extended by a local verifier, which checks the measurements against a given *Reference Integrity Metrics (RIM)*. This device sided verifier offers assertions to the integrity values. Without this ability each mobile device has to contact a central service provider for checking the integrity values. This process can be done instantly as the measurements are performed by using a combination of RTM and RTV.

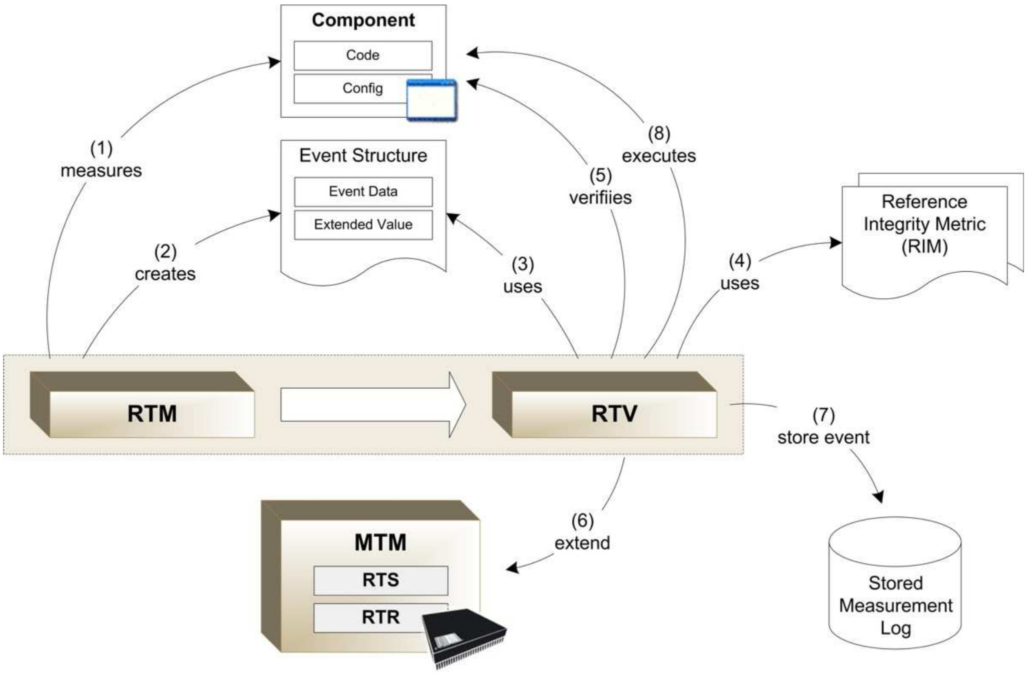


Figure 2.12: *Measurement and Verification Process*

During the so-called local attestation process the verifier receives the log and a signed PCR value as well as the certificates to verify the signature. PCRs are signed by special signing keys whose operation is limited to sign PCRs. These *Attestation Identity Keys (AIK)* are created by the TPM. Figure 2.12 depict such a *Measure→Verify→Extend* process that is detailed as follows:

Initially the RTM measures a software component (Step 1) and creates a so-called Event Structure (Step 2). A Event Structure contains extend value (actual result of digest) and extend data. As indicated in Figure 2.12 the RTM assigns the verification task to the RTV. Then the RTV uses the Event Structure (Step 3) with the taken measurements (Step 4) and verifies it against a set of available *Reference Integrity Metrics (RIM)* (Step 5). If the verification is accepted, the RTV extends the data to a dedicated PCR (Step 6) and stores the Event Structure in the

Stored Measurement Log (SML) (Step 7). The SML contains the Event Structures for all measurements in the TPM and can be stored in any storage (e.g. hard disk). Finally, the RTV executes the software component (Step 8).

Root of Trust for Enforcement A RTE is required if an engine uses allocated ressources and services. In this case, such RoT acts as a trusted boot loader and ensures the availability of all allocated trusted resources and services within that trusted subsystem. Each trusted software compartment boots from a dedicated Root-Of-Trust-For-Enforcement that creates the allocated trusted components for an allocated trusted engine.

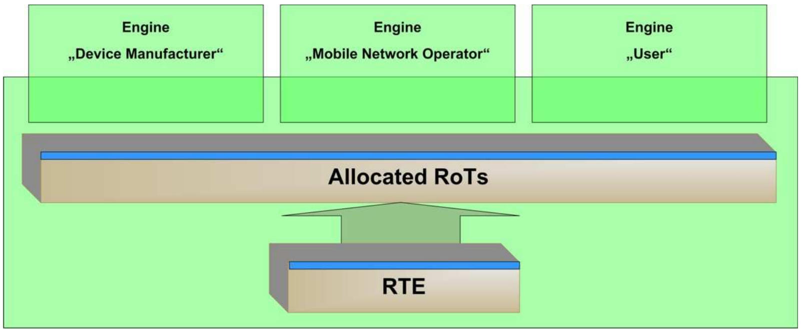


Figure 2.13: *Enforcement of Allocated Root-of-Trusts*

Figure 2.13 illustrates schematically such a boot process of allocated resources. In this example, the RTE (or an derived instantiation agent) is responsible for the correct instantiation of the three trusted engines TE_{DE}, TE_{MNO} and TE_U. If the RTE is challenged, it is able to provide evidence of a current status of the instantiation of an confided allocated resource.

2.2.4.3 Services of a Trusted Engine

A trusted engine integrates all functionality by customizing available platform ressources as software services. Such a service offers computation, storage, or communication channel to other internal or external services and applications based on dedicated or allocated ressources. The TCG MPWG categorizes such services into: trusted services, normal services, and measured services.

A trusted services customizing trusted resources. Thus, a trusted service is implicitly supplied with an EK or AIK in order to vouch its trustworthiness. Trusted services are intended to provide reliable measurements of its current state and to provide evidence of the state of other normal services or resources.

Normal services are customizing normal resources and implementing functionality, which are not able to provide evidence of their trustworthiness by their own capabilities. However, normal services are able to access internal trusted services to use its provided functionality. Therefore, internal normal services are able to vouch their trustworthiness by associated integrity metrics that have been measured by a trusted service.

2.2.5 Mobile Trusted Module

The generic term *Mobile Trusted Module (MTM)* refers to a dedicated hardware-based trust-anchor. It is typically composed of a RTS and RTR and has comparable characteristics with a TPM. According to their design objective the TCG MPWG distinguishes between *Mobile Remote-Owner Trusted Module (MRTM)* and *Mobile Local-Owner Trusted Module (MLTM)*. Both, the MRTM and the MLTM must support a subset of TPM commands as specified in [Tru06b]. Additionally, a MRTM has to support a set of commands to enable local verification and specific mobile phone functionality.

The *TCG MPWG Reference Architecture* does not exclude to utilize a TPM v1.2 (or even a TPM v1.1) as a MTM, if an appropriate interface consisting of a set of commands conformant to the TCG MPWG specification and associated data structures are provided. In this context, we expect three different solutions for isolation, key management and protection of TSS_σ .

Standard TPM-based Model The first one uses a non-modified standard TPM to build the trusted computing base of this system. The secret keys are stored into a single key-hierarchy on behalf of DO as specified in [Tru05]. In this case, an adversary or malicious local owner may be able to access the secret keys of a remote stakeholder and take control of a remote owner compartment. A DO can also disable the whole MTM or corrupt mandatory engines of remote stakeholders.

Software-based MTM-Emulation Model The second solution uses a software-based allocated MTM-emulation with an isolated key-hierarchy per vMTM instance. All sensitive and security-critical, such as EK or SRK, are only protected by software mechanisms outside the tamper-resistant environment of an dedicated MTM. However, the advantage of this approach is a high performance, since a vMTM is completely implemented as a software object [BCG$^+$06, Str05].

Generic MTM-based Model supporting multiple Stakeholder and virtual MTMs

In order to circumvent resulting drawbacks and attacks, we clearly favor a solution with a higher level of security. For this reason, we adopt the proposed secure co-processor variant of [BCG$^+$06] and describe a generic MTM with support for multiple stakeholder environments.

In an cost-efficient scenario, the trusted mobile platform is implementable based on a single generic MTM and several virtualized MTMs for each trusted engine. Hence, at least one dedicated MTM has to be available and additionaly a unique vMTM has to be instantiated in each trusted subsystem TS_σ. In such case, a physical-bounded MTM acts as a master trust-anchor and offers MRTM and MLTM functionality with respect to its specific use-case.

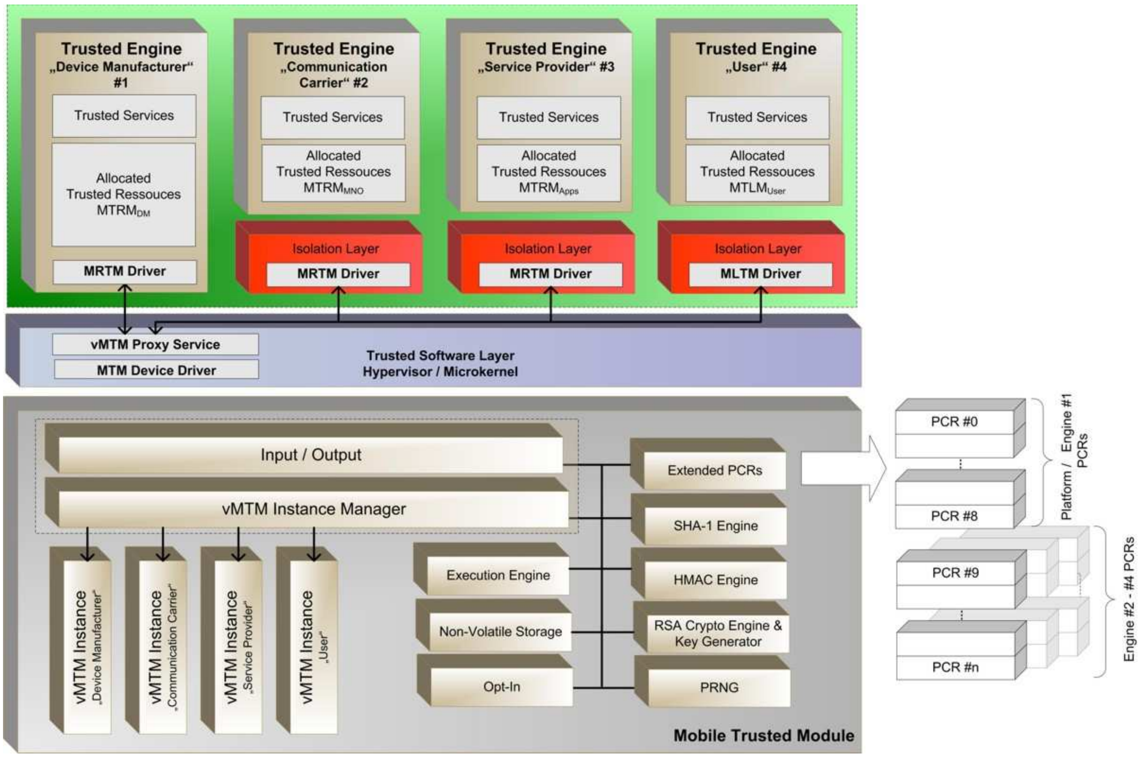


Figure 2.14: *MTM Architecture supporting Multiple-Stakeholder*

The Trusted Software Layer offers a *vMTM Proxy Service* to all embedded trusted engines TE_σ. The main task of this service is to route MTM commands from a TE_σ to its dedicated instance $vMTM_\sigma$. The advantage is that all security-critical MTM commands are tunneled to $vMTM_\sigma$ and are executed within the protected environment of the dedicated MTM.

Figure 2.14 illustrates the architecture of a generic MTM with isolated vMTM compartments. These architecture requires a slightly modified TPM. Mainly, we add a trusted component, the *vMTM Instance Manager*, which is responsible to separate vMTM instances from each other. This includes administration, isolated execution, memory management and access control for each stakeholder compartment. Thus, a vMTM instance is able to hold an autonomous and hardware-protected key-hierarchy to store its secrets and protect the execution of security-critical data (e.g. signature and encryption algorithms).

Chapter 3

Subscriber Authentication with virtual SIMs

In the first chapter, we have introduced different criteria and guidelines (see Section 1.2), which are essential for the examination and evaluation of the different architectures. Afterwards, the basics of mobile communication and subscriber authentication in mobile cellular networks, as well as Trusted Computing technology, were introduced in Chapter 2.

Based on this fundament, we systematically discuss an approach for an efficient and maintainable architecture for subscriber authentication, using a single trust-anchor and virtual SIMs. We first have a look on procedures and mechanisms of efficient administration, management and maintenance of subscriber credentials. Second, we will design three intergraded conceptual models for subscriber authentication in mobile cellular networks using trusted computing. In model "One" we present a authentication model that is straight-forward to actual GSM standard and enables subscriber access to mobile cellular networks based on Trusted Computing with compatibility to regular GSM authentication. Model "Two" is more comprehensive. Additionally to the precedent model, we integrate remote attestation for restricted network access. Beside the main task of SIM substitution, it provides remote attestion and mutual authentication between the MNO and the mobile station. The last considered protocol is a generalized proposal for user- and device authentication using vSIM credentials in generic network infrastructures. This model is based on Trusted Computing and like the predecessor it supports remote attestion and mutual authentication. In contrast to the previous models, we are using more generalized assumptions and specifications.

This vSIM architecture shows how a traditional SIM-Card could be replaced by a software emulation, which runs within an environment, protected and supported by a Mobile Trusted Module. It offers a suitable and sustainable solution with regard to the defined criteria (as we will see Chapter 4), that is able to compete with conventional SIM-based solutions.

This chapter is organized as follows. We will give an informal description of the scenario in Section 3.1 and identify the essential components of an idealized protocol using a vSIM credential for subscriber authentication. In Section 3.2, we give an overview of the vSIM architecture. Based on this analysis, a set of protocols for deployment and management of a vSIM Credential and a vSIM Container are discussed in Section 3.3. Therefor, we propose a model for take-ownership of a remote stakeholder in 3.3.1. In Subsection 3.3.2 a protocol for subscriber enrollment and vSIM Credential roll-out are discussed. Furthermore, a protocol for migration of a vSIM Container is presented in Subsection 3.3.3. An substantial part of this thesis holds Section 3.4. Here, we present three intergraded conceptual models for subscriber enrollment and authentication in mobile cellular networks using trusted computing. Finally, we introduce to a design for prototypical implementation of the specified vSIM services in Section 3.5.

3.1 Scenario

The use-case under consideration is illustrated in Figure 3.1 and involves four significant entities: the device owner / local user (U), the mobile trusted platform (MTP), the Mobile Network Operator (MNO), and the Point-of-Sale/Point-of-Presence (POS). In this scenario, U wants to establish a long-time relationship with the MNO (Step 1,2), in order to use the mobile network infrastructure and its

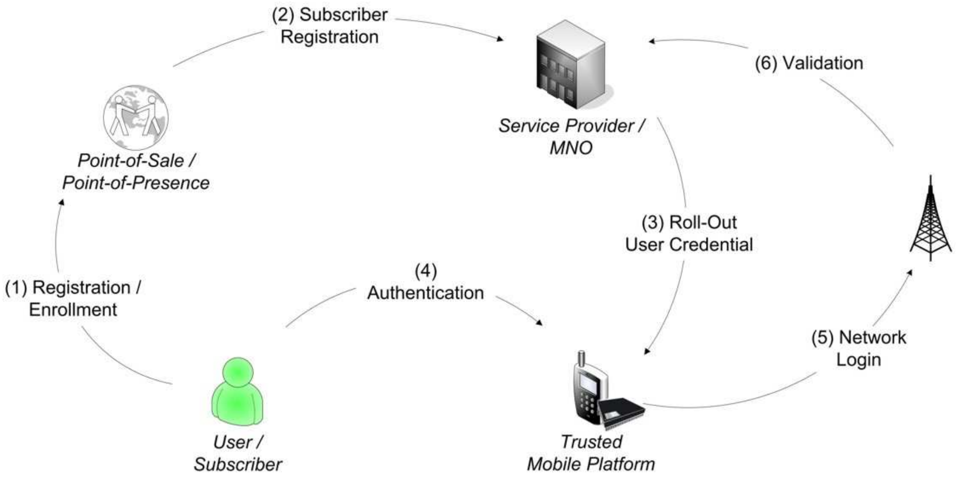

(2) Subscriber Registration
(6) Validation
Service Provider / MNO
Point-of-Sale / Point-of-Presence
(3) Roll-Out User Credential
(1) Registration / Enrollment
(4) Authentication
(5) Network Login
User / Subscriber
Trusted Mobile Platform

Figure 3.1: *Generic Trusted Mobile Scenario*

offered services (e.g. GSM, UMTS or Location Based Services).

Instead of purchasing a physical SIM card, the MNO supplies the $vSIM_{CORE}$ service inside TSS_{MNO} with a virtual software SIM credential (Step 3). Every time a user wants to access the mobile network, he/she authenticates to the vSIM service (Step 4), which uses the vSIM credential to perform network authentication (Step 5,6).

3.2 Architectural Overview of a vSIM Platform

The TCG MPWG has developed an architecture on a high level of abstraction for trusted mobile platforms. In this section, we detail significant components and services of a vSIM platform. Figure 3.2 schematically shows the layout of such a trusted platform. It holds a virtual software SIM service which substitutes the traditional smartcard and its functionality.

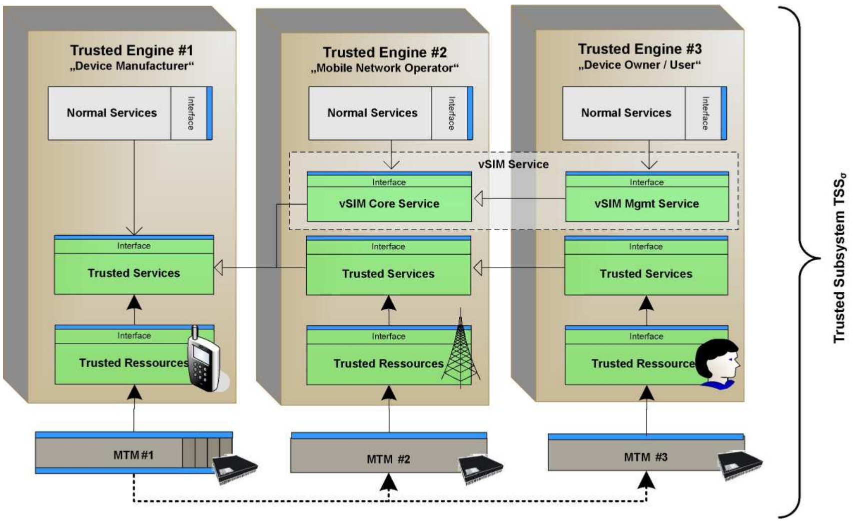


Figure 3.2: *TCG MPWG Architecture (vSIM)*

3.2.1 Platform Stakeholders and vSIM Container

In general, a MTP supports a set of trusted environments as described in Section 2.2.4. Each environment represents a protected domain associated with a specific stakeholder. In our purpose, we consider three different stakeholders: the Device Manufacturer (DM), the Mobile Network Operator (MNO), and the Device Owner / User (U).

As defined in Section 2.2.4, we specify a trusted subsystem TSS_{σ} as a logical unit of a trusted engine together with its interrelated hardware compartment of a stakeholder σ.

- *Device Manufacturer Subsystem:* The TSS_{DM} is responsible for the integrity and configuration of a device. It typically controls all internal and external communications and provides all security-critical hardware resources of a device. For this reason, all protocol messages of an embedded TSS_{σ} are routed through resources of TSS_{DM} to its destination.

- *Mobile Network Operator Subsystem:* All cellular services of a platform are assigned to TSS_{MNO}. It is responsible for administration and protection of the *vSIM Credential* ($Cred_{vSIM}$) and implements the network authentication mechanisms. Therefor, it provides a *vSIM Core Service* ($vSIM_{CORE}$) to the device owner, which implements the fundamental SIM functionality. We use the term *vSIM Container* as a synonym for a fully equipped TSS_{MNO}, holding the required $vSIM_{CORE}$ services.

- *Device Owner / User Subsystem:* In context of the vSIM service, the TSS_{U} protects all personal information and corresponding user credentials. Moreover, it holds a *vSIM Management Service* ($vSIM_{MGMT}$) and is responsible for administration and authentication of local users. In particular, $vSIM_{MGMT}$ offers an internal authentication oracle to the $vSIM_{CORE}$ service, in order to provide evidence of a local user's identity.

3.2.2 Generic Platform Assumptions

In this section we examine the platform assumptions in order to describe the idealized protocols:

1. A Mobile Station (MS) holds a small-scale embedded trust-anchor as described in Chapter 2.2. Contrary to Section 2.1.1.1, such a MS is not necessarily equipped with a traditional SIM. Therefore, we use term *Mobile Trusted Platform (MTP)* for a TC-supported mobile device enabling to distinguish these two different types of Mobile Stations.

2. The MTP is build upon a Trustworthy Operating System. The Trusted Software Layer of this OS supports virtual machines with several trusted engines and services compliant to the TCG requirements [Tru06a, Tru06b].

3. The MTP is at least running three trusted subsystems TSS_{DE}, TSS_{MNO} and TSS_U on behalf of the specified stakeholders DE, MNO and U, with an associated isolated execution environment and protected storage functionality. Each subsystem is used for critical security functions and consists of a trusted module TM_σ and associated trusted engine TE_σ.

4. For each TSS_σ, a take-ownership procedure has been carried out by the stakeholder σ. A TSS_σ implements a fully functional environment that is qualified to sign and encrypt arbitrary data.

5. TSS_U is able to generate asymmetric signature keys, which are only accessible by the dedicated platform user U.

6. A TSS_{MNO} holds a service, called $vSIM_{CORE}$. This service is responsible for the core functionality of a vSIM. In particular, this service implements the authentication mechanisms.

7. A TSS_U holds a $vSIM_{MGMT}$ service. This service is responsible for subscriber management of a vSIM. In particular, this service is responsible for administration and management of a local subscriber.

8. The MNO offers all required functionality of a Public Key Infrastructure, either directly or indirectly.

9. The MNO has a certificate $Cert_{MNO}$, issued by a certification authority CA. This certificate associates the identity of the MNO with the public key K_{MNO}^{pub} value for the verification of digital signatures. This certificate must be available to MTP and its embedded services.

3.2.3 Security Requirements

It is important, that our proposed vSIM architecture are at least as secure as traditional SIM-Cards. Therefore, the platform must satisfy some generic SIM security characteristics, namely *Protected Storage*, a tamper-resistant *Isolated Execution Environment* and *User Authentication*, as stipulated in 2.1.3.1. In addition, a vSIM platform has to guarantee that only authorized subjects can access or manipulate protected vSIM data, while

1. in transit to the vSIM Container, or other trusted services

2. in storage on the mobile platform
3. it is executed within the execution environment
4. it is transferred between environments of authorized subjects

This includes, that an adversary is not able to destroy or modify security-sensitive data or circumvent the access control mechanisms. It also must prevent leakage of sensitive information and has to guarantee that all required services are availability and work as expected. In particular, it guarantees that only entities authorized by the associated stakeholder are able to access the specific trusted vSIM services.

3.3 Protocols for Deployment and Management of vSIM Credential

In this section, we design a set of protocols for deployment and management of vSIM Credentials and and its associated services. These protocols describe a very important part of the vSIM architecture, since their abstract task is to offer comfortable usage, maintenace and administration characteristics to both, the local operator and the MNO.

The following generic methods for deployment and management are proposed for the vSIM architecture:

- Remote-Take-Ownership of a vSIM Container,
- Subscriber Enrollment and vSIM Credential Roll-Off, and
- Migration of a vSIM Container and vSIM Credential

These protocols are inherently required the meet the properties from Section 1.2. In particular the criteria *Portability*, *Usability* and last but not least *Acceptance* are affected.

3.3.1 Setup and Remote-Take-Ownership of a vSIM Container

A device owner of a mobile trusted platform must be able to buy a 'blank' mobile phone which is not pre-allocated and initialized by a specific MNO, so that he/she can choose an arbitrary mobile network operator without restrictions. This protocol to is used to

- perform a remote take-ownership of a TSS_{MNO}, and
- setup, customize and finalize the vSIM Container by the MNO.

The take-ownership operation builds the basic trust relationship between a stakeholder and trusted mobile platform. Currently, the *TCG MPWG Reference Architecture* does not define how a remote MNO can proceed this initial setup and take-ownership of its TSS_{MNO}.

In the next section, we give an approach to perform this operation. The main idea behind this procedure is to install and instantiate a 'blank' trusted subsystem TSS^*_{MNO} containing a pristine engine TE^*_{MNO} with a set of generic trusted services TS^*_{MNO}. This subsystem will be certified by a remote owner, if the platform is able to provide evidence of its pristine configuration and policy conformance respectively MNO. Figure 3.3 illustrates this process.

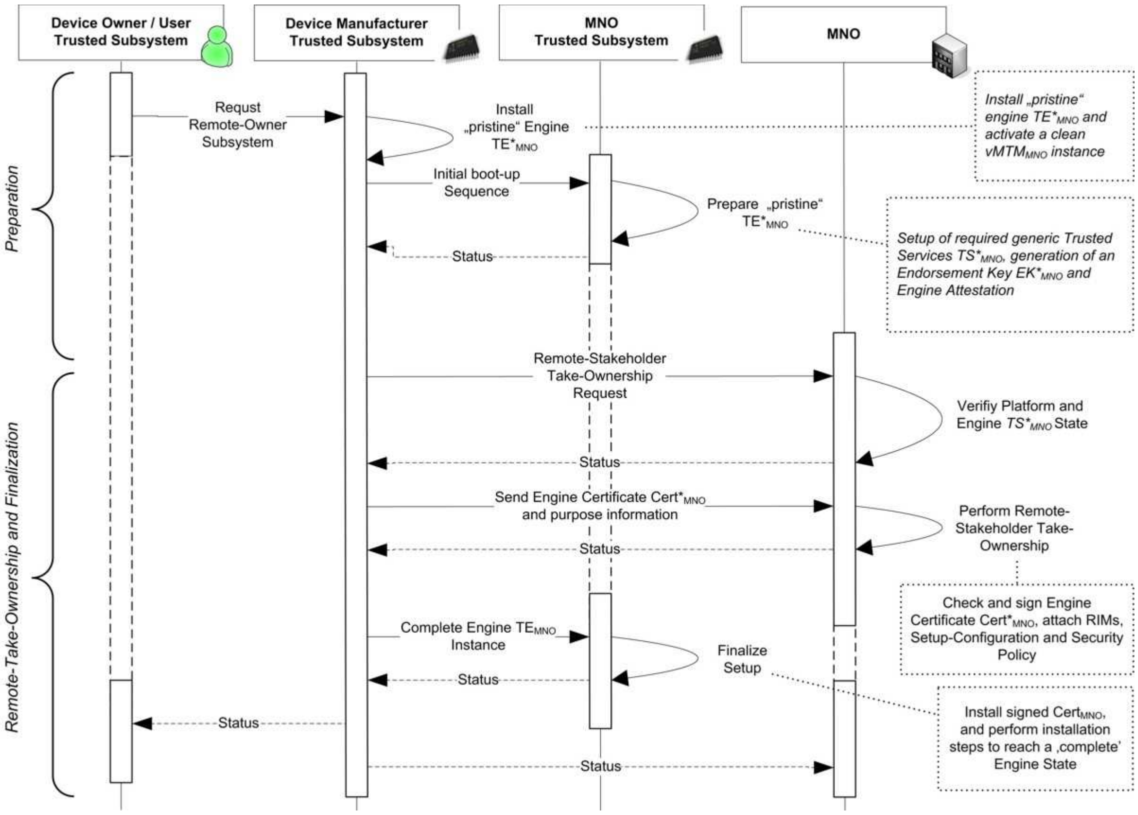

Device Owner / User Trusted Subsystem
Device Manufacturer Trusted Subsystem
MNO Trusted Subsystem
MNO
Preparation
Remote-Take-Ownership and Finalization
Requst Remote-Owner Subsystem
Install „pristine“ Engine TE*MNO
Install „pristine“ engine TE*MNO and activate a clean vMTMMNO instance
Initial boot-up Sequence
Prepare „pristine“ TE*MNO
Setup of required generic Trusted Services TS*MNO, generation of an Endorsement Key EK*MNO and Engine Attestation
Status
Remote-Stakeholder Take-Ownership Request
Verifiy Platform and Engine TS*MNO State
Status
Send Engine Certificate Cert*MNO and purpose information
Perform Remote-Stakeholder Take-Ownership
Status
Check and sign Engine Certificate Cert*MNO, attach RIMs, Setup-Configuration and Security Policy
Complete Engine TEMNO Instance
Finalize Setup
Status
Install signed CertMNO, and perform installation steps to reach a ‚complete' Engine State
Status
Status

Figure 3.3: *Remote Stakeholder Take-Ownership Protocol*

Platform and Protocol Precondition In a preliminary state, the trusted mobile platform has carried out the boot process and has loaded the trusted computing base and the engine TE_{DM} with its trusted services. The trusted platform has checked that the installed hardware and running software are in a trustworthy state and configuration. It is able to report and attest this state, if challenged by an authorized entity.

Protocol Scheme and informal Description In the first phase, the trusted engine TE_{DM} carries out a take-ownership preparation for the remote stakeholder. A 'blank' engine TE^*_{MNO} is installed and booted by the RTE_{DM}, and a clean $vMTM_{MNO}$ instance is activated inside the dedicated MTM. An initial setup prepares the pristine engine TE^*_{MNO}. A endorsement key-pair $EK^*_{TSS_{MNO}}$ is generated within $vMTM_{MNO}$ with an corresponding certificate $Cert_{TSS_{MNO}}$ [1].

Next, TE^*_{MNO} performs an attestation of its current state. The attestation can be done by the local verifier RTV_{DM} inside the TSS_{DM} using RIM certificates of the remote stakeholder RO. If no suitable RIM and corresponding RIM-certificate are available for an pristine engine, alternatively a remote attestation with an associated Privacy-CA is also possible.

$$TE_{MNO} \rightarrow MNO \quad : \quad ATTEST(S_i)$$

TE^*_{MNO} creates an symmetric session key K_S and encrypts the public part of the endorsement key $EK^*_{TSS_{MNO}}$, the corresponding certificate $Cert_{TSS_{MNO}}$, attestation and purpose information. Next, TE^*_{RO} encrypts K_S with a public key K^{pub}_{MNO} and sends both messages to the MNO. We assume that this key is either public available or pre-installed by the device manufacturer.

$$\begin{aligned} TE_{MNO} \rightarrow MNO \quad : \quad & ENC_{K_S}(\{EK^*_{MNO}, Cert_{TSS_{MNO}}, ...\}), \\ & ENC_{MNO}(K_S) \end{aligned}$$

After the messages are received by the MNO, the messages are decrypted using the private-part of key K^{pub}_{MNO}.

In a next step, the MNO verifies the attestation data and checks the intended purpose of TSS^*_{MNO}. If the engine and device attestation data is valid and the intended purpose is acceptable, the MNO generates an individual security policy SP_{MNO}. The MNO signs the $Cert_{TSS_{MNO}}$ and creates RIM certificates for local verifica-

[1] Typically, the key-generation needs an *Owner-Authentication*. Because it is problematic in a remote-owner scenario, authentication of command execution may enforced by challenge-response mechanisms between MNO and TSS_{MNO}.

tion of a 'complete' TSS_{MNO}. Furthermore, MNO creates a Setup-Configuration $SC_{TSS_{MNO}}$, which enforces the engine to individualize its services and complete its configuration with respect to the intended purpose and given security policy. In this step, MNO encrypts the messages with the public part of the $EK^{pub}_{TSS_{MNO}}$ and transfer this package to the engine TE_{MNO}.

$$\begin{aligned} MNO \rightarrow TE_{MNO} \quad : \quad & ENC_{TSS_{MNO}}(\{SP_{MNO}, SIGN_{MNO}(Cert_{TSS_{MNO}}), \\ & RIM_{MNO}, SC_{TSS_{MNO}}\}) \end{aligned}$$

Finally, the trusted engines TE^{*}_{RO} decrypts the received package and installs it inside the TSS_{RO} and completes its instance.

3.3.2 Subscriber Enrollment and vSIM Credential Roll-Off

A user of a mobile trusted platform wants to acquire an vSIM credential to use with the $vSIM_{CORE}$ service. The subscriber credentials are pre-generated by the MNO, derived from an initial secret, or generated by the MNO during the acquisition. This protocol is used to

- request a vSIM credential,
- authenticate the involved entities, and
- download and install the requested vSIM credential.

Because the vSIM services are completely implemented as a trusted software application, it implies that the respective vSIM credentials has to be transferred from the MNO to the vSIM service in a secure manner. In traditional SIM-based systems, the subscriber gets a security token after his/her enrollment. Contrary to a vSIM, this security token physically exists and can be pre-delivered with an included key, to the respective Point-of-Sale.

Platform and Protocol Precondition In a preliminary state of all protocols, the platform has carried out an authenticated boot process and has loaded the specific trusted software layer of the OS and its trusted compartments. This includes the trusted engines with its embedded services $vSIM_{CORE}$ and $vSIM_{MGMT}$. The trusted platform has checked that the installed hardware and running software are in a trustworthy state and configuration and it is able to report and attest this state, if challenged by an authorized entity.

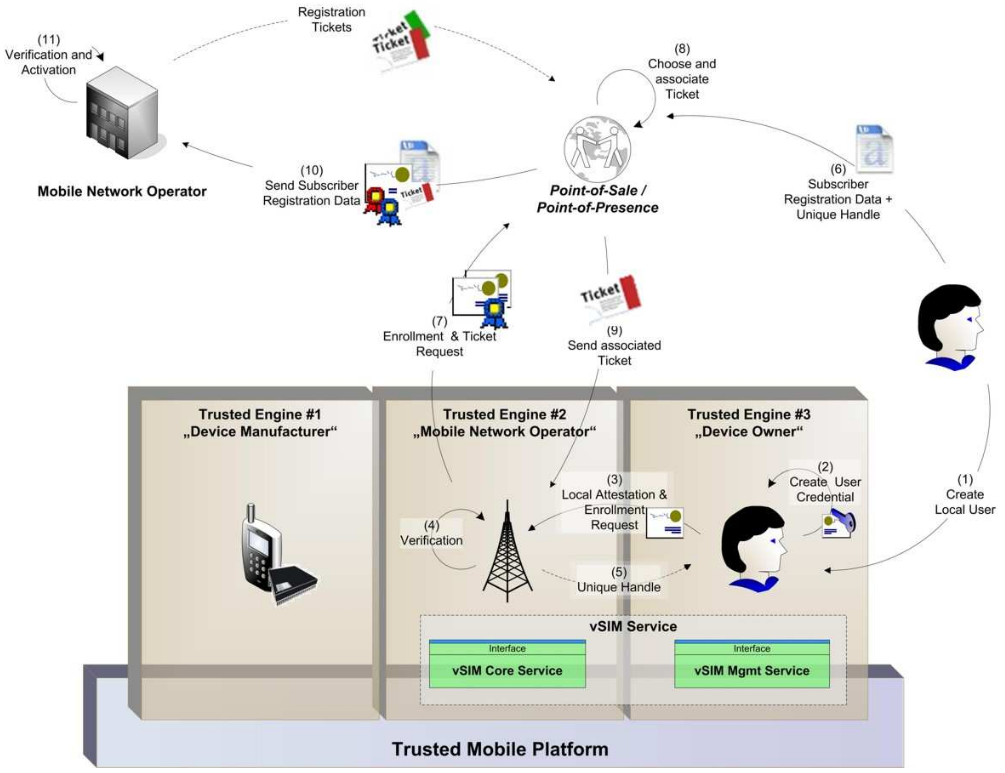

Registration Tickets
Ticket
Ticket
(11) Verification and Activation
(8) Choose and associate Ticket
Mobile Network Operator
(10) Send Subscriber Registration Data
Ticket
Point-of-Sale / Point-of-Presence
(6) Subscriber Registration Data + Unique Handle
(7) Enrollment & Ticket Request
Ticket
(9) Send associated Ticket
Trusted Engine #1 „Device Manufacturer“
Trusted Engine #2 „Mobile Network Operator“
Trusted Engine #3 „Device Owner“
(2) Create User Credential
(1) Create Local User
(3) Local Attestation & Enrollment Request
(4) Verification
(5) Unique Handle
vSIM Service
Interface
vSIM Core Service
Interface
vSIM Mgmt Service
Trusted Mobile Platform

Figure 3.4: *Model "Subscriber Registration and Enrollment"*

3.3.2.1 Protocol Scheme and informal Description

The Point-of-Sale POS orders a set of (pre-generated) registration tickets $Ticket_i$ from the MNO. A registration ticket consists of a triple:

$$Ticket_i := \{IMSI_i, RAND_i, AUTH_i\}.$$

The $IMSI_i$ identifies an *International Mobile Subscriber Identity* as described in Section 2.1.3. In other scenarios, it may be a unique credential ID that is assigned by the network operator. The term $RAND_i$ denotes a random value, which is needed to challenge TSS_{MNO} in the course of the protocol. Finally, with the $AUTH_i$ the trusted platform is able to check the integrity and authenticity of $Ticket_i$.

Phase 1 : "Subscriber Registration and Enrollment" The user enrollment and vSIM credential roll-out are separated into two phases. The following protocol sequence is depicted in Figure 3.4 describes the first phase. Here, we discuss the user enrollment and registration for services, offered by the MNO.

The user starts to request a new user credential for a local user, which is generated by TSS_U. For this, the local user enters a unique personal identifier ID_U, personal registration data $REGDATA_U$ and an authorization password CHV_U to the trusted service $vSIM_{MGMT}$ (Step 1).

$$U \to vSIM_{MGMT} \quad : \quad ID_U, CHV_U, REGDATA_U$$

Afterward, $vSIM_{MGMT}$ generates a asymmetric signature key-pair K_U and creates a certificate, which includes all relevant information, like the $REGDATA_U$ and the public portion of K_U (Step 2). The $vSIM_{MGMT}$ pass this certificate $CERT_U$ to the $vSIM_{CORE}$ service (Step 3).

$$\begin{aligned} vSIM_{MGMT} \to vSIM_{CORE} \quad : \quad & ATTEST(S_i), \\ & CERT_U \end{aligned}$$

Within this step, $vSIM_{MGMT}$ requests a enrollment procedure and reports its current state and configuration to the local verifier of $vSIM_{CORE}$. The TSS_{MNO} validates the given data (against Reference Integrity Metrics (RIM)) and checks, whether the present engine's state is in a acceptable condition (Step 4). Once the $vSIM_{CORE}$ is convinced about trustworthiness of the device, generates a unique handle PID of this process and sends this value to the $vSIM_{MGMT}$ (Step 5).

$$vSIM_{CORE} \to vSIM_{MGMT} \quad : \quad PID$$

Now, the user starts to communicate its registration data $REGDATA_U$ (e.g. name, address, accounting information, passport ID) and the PID to the Point-of-Sale (Step 6). $vSIM_{CORE}$ requests an enrollment procedure for U. Therefor, it signs the PID, its own certificate and the obtained user certificate and sends this package to the POS (Step 7).

$$\begin{aligned} U \to POS \quad &: \quad PID, REGDATA_U \\ vSIM_{CORE} \to POS \quad &: \quad SIGN_{TSS_{MNO}}(PID) \end{aligned}$$

After having received the request, POS chooses a $Ticket_i$, bind it to the key $K^{pub}_{TSS_{MNO}}$ (Step 8) and sends it to TSS_{MNO} (Step 9). In this case the POS could be an arbitrary point-of-sale or internet portal, which is accredited by the MNO.

$$POS \to TSS_{MNO} \quad : \quad BIND_{TSS_{MNO}}(Ticket_i)$$

Once the POS is convinced about trustworthiness of both, user and device, it attaches $CERT_U$ and the $IMSI_i$ (of the chosen ticket) to the given $REGDATA_U$, signs all gathered information with its private portion of its signature key K_{POS} and sends the signed data (online or offline) to the MNO (Step 10). Optionally, the POS encrypt the data with the public portion of K_{MNO}.

$$\begin{aligned} POS \rightarrow MNO \ &: \ IMSI_i, CERT_U, REGDATA_U \\ &: \ SIGN_{POS}(IMSI_i, CERT_U, REGDATA_U) \end{aligned}$$

The MNO verifies the data and generates the $Cred_{vSIM}$ with the $IMSI_i$, the shared key K_i and the certificate $CERT_U$ and signs this bundle with the private signature key K_{MNO}. Finally, the MNO activates the signed $Cred_{vSIM}$ and the corresponding nonces in its authentication center (Step 11). Now, the mobile device is able to access the registration service provided by MNO over some kind of channel. For instance, this service is implementable as a network teleservice or internet download service.

Phase 2: "Secure vSIM Roll-Out and Installation" In the second phase of the protocol details the secure vSIM roll-out and installation as illustrated in Figure 3.5.

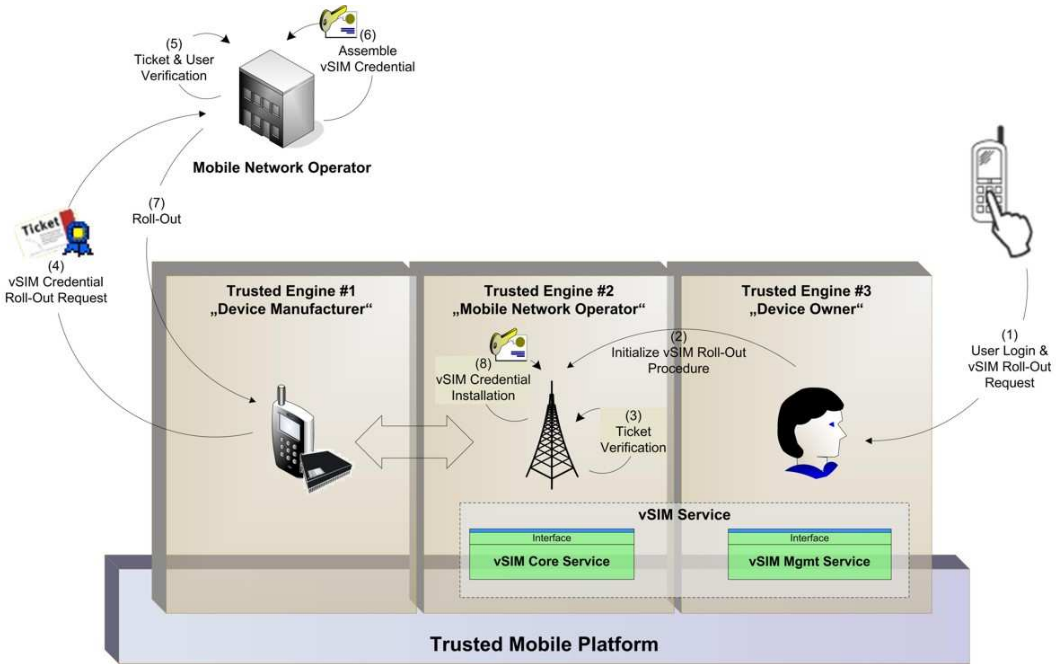


Figure 3.5: *Model "vSIM Credential Roll-Out"*

In order to obtain a $Cred_{vSIM}$, the user performs a log-in sequence and sends a unique id ID_U with a proper password CHV_U to the $vSIM_{MGMT}$ service, which loads the associated user key-pair K_U from protected storage (Step 1).

$$U \rightarrow vSIM_{MGMT} \quad : \quad ID_U, CHV_U$$

In a next step, the $vSIM_{MGMT}$ initializes a *vSIM Roll-Out Procedure* and sends a request the the $vSIM_{CORE}$ service (Step 2).

$$vSIM_{MGMT} \rightarrow vSIM_{CORE} \quad : \quad init\_rollout\_vsim$$

After having received this message, it unbinds the corresponding $Ticket_i$ and verifies the authenticity and integrity of the $Ticket_i$ (Step 3). Next, $vSIM_{CORE}$ extracts the $NONCE_U$ from the $Ticket_i$ and challenge U with this value.

$$vSIM_{CORE} \rightarrow vSIM_{MGMT} \quad : \quad NONCE_U$$

$vSIM_{MGMT}$ signs the $NONCE_U$ together with its ID_U in order to prove its identity to the MNO. This bundle is sent back to the $vSIM_{CORE}$.

$$vSIM_{MGMT} \rightarrow vSIM_{CORE} \quad : \quad SIGN_{TSS_U}(ID_U \parallel NONCE_U)$$

After the $vSIM_{CORE}$ has received the message, it composes a vSIM credential request and submits it to the assigned MNO registration service via some channel, mentioned above (Step 4). Therefor, $vSIM_{CORE}$ extracts $NONCE_{MNO}$ from the $Ticket_i$ and signs it together with the $IMSI_i$. Afterwards, the $vSIM_{CORE}$ sends its own signature and the obtained user signature to MNO.

$$\begin{aligned} vSIM_{CORE} \rightarrow MNO \quad : \quad & SIGN_{TSS_{MNO}}(IMSI_i \parallel NONCE_{MNO}) \\ & SIGN_{TSS_U}(ID_U \parallel NONCE_U) \end{aligned}$$

Having received $vSIM_{CORE}$'s request, MNO verifies the messages and obtain $CERT_U$ and $Cert_{TSS_{MNO}}$ (either from the request or from local storage) (Step 5). If revoked, it replies with an error message and halts the protocol. Otherwise the request is approved by the MNO. Next, MNO prepares $Cred_{vSIM}$ for transfer to $vSIM_{CORE}$, and generates a randomly chosen session key K_S. Afterwards, the key K_S is bound with to corresponding binding key of TSS_{MNO} to the destination platform (Step 6).

The MNO encrypt $Cred_{vSIM}$ with this session key and sends both to the TSS_{MNO} (Step 7).

$$\begin{aligned} MNO \rightarrow vSIM_{CORE} \quad : \quad & ENC_{K_S}(Cred_{vSIM}) \\ & BIND_{TSS_{MNO}}(K_S) \end{aligned}$$

Finally, TSS_{MNO} unbinds K_S. With this key it decrypts the vSIM credential and checks the enclosed signature (Step 8). If the decryption is correctly processed and the signature is verified, $vSIM_{CORE}$ seals the obtained $Cred_{vSIM}$ to valid platform configurations and finishes its installation.

Alternatively, the MNO could generate the shared key K_S in a preliminary stage, and include an encrypted vSIM credential $Cred_{vSIM,i}$ to the $Ticket_i$. In this case, the MNO only sends the bound key K_S to the $vSIM_{CORE}$ on the client platform. Another variation, is to bind the vSIM credential $Cred_{vSIM}$ directly, instead of using a symmetric session key K_S.

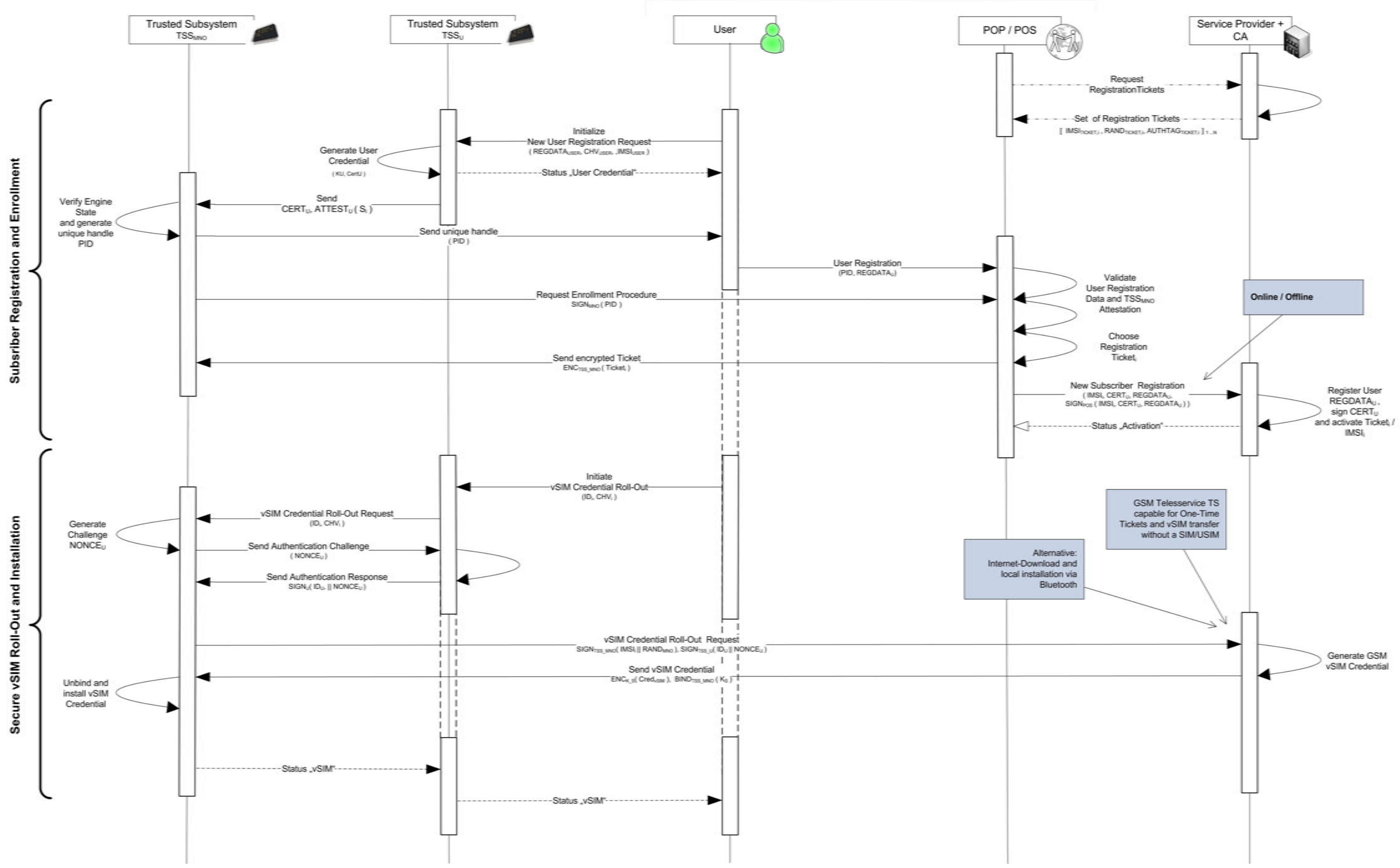

Trusted Subsystem TSS$_{MNO}$
Trusted Subsystem TSS$_{U}$
User
POP / POS
Service Provider + CA
Request RegistrationTickets
Set of Registration Tickets
[IMSI$_{TICKET,i}$, RAND$_{TICKET,i}$, AUTHTAG$_{TICKET,i}$] $_{1..N}$
Subsriber Registration and Enrollment
Initialize New User Registration Request
(REGDATA$_{USER}$, CHV$_{USER}$, IMSI$_{USER}$)
Generate User Credential
(KU, CertU)
Status „User Credential"
Verify Engine State and generate unique handle PID
Send CERT$_{U}$, ATTEST$_{U}$ (S$_{i}$)
Send unique handle (PID)
User Registration (PID, REGDATA$_{U}$)
Validate User Registration Data and TSS$_{MNO}$ Attestation
Choose Registration Ticket$_{i}$
Request Enrollment Procedure
SIGN$_{MNO}$ (PID)
Online / Offline
Send encrypted Ticket
ENC$_{TSS\_MNO}$ (Ticket$_{i}$)
New Subscriber Registration
(IMSI$_{i}$, CERT$_{U}$, REGDATA$_{U}$,
SIGN$_{POS}$ (IMSI$_{i}$, CERT$_{U}$, REGDATA$_{U}$))
Register User REGDATA$_{U}$, sign CERT$_{U}$ and activate Ticket$_{i}$ / IMSI$_{i}$
Status „Activation"
Secure vSIM Roll-Out and Installation
Initiate vSIM Credential Roll-Out
(ID$_{U}$ CHV$_{U}$)
Generate Challenge NONCE$_{U}$
vSIM Credential Roll-Out Request
(ID$_{U}$ CHV$_{U}$)
Send Authentication Challenge
(NONCE$_{U}$)
Send Authentication Response
SIGN$_{U}$(ID$_{U}$ || NONCE$_{U}$)
GSM Telesservice TS capable for One-Time Tickets and vSIM transfer without a SIM/USIM
Alternative: Internet-Download and local installation via Bluetooth
vSIM Credential Roll-Out Request
SIGN$_{TSS\_MNO}$(IMSI$_{i}$ || RAND$_{MNO}$), SIGN$_{TSS\_U}$(ID$_{U}$ || NONCE$_{U}$)
Generate GSM vSIM Credential
Send vSIM Credential
ENC$_{K\_S}$(Cred$_{vSIM}$), BIND$_{TSS\_MNO}$ (K$_{S}$)
Unbind and install vSIM Credential
Status „vSIM"
Status „vSIM"

Figure 3.6: *User Enrollment and Credential Transfer Protocol*

3.3.3 Migration of a vSIM Container and vSIM Credential

In a typical scenario, a local operator has to be enabled to move a vSIM Credential from a source $TSS_{MNO,S}$ to another MTM-enabled mobile device (e.g. after having bought a new mobile phone). Therefore, this migration protocol provides

- Mobility of a vSIM Credential, and
- Migration functionality of a vSIM Container.

All security-critical information including the Storage Root Key (SRK) has to be migrated to the target $TSS_{MNO,D}$. In our scenario, we assume the same remote owner on both subsystems $TSS_{MNO,S}$ and $TSS_{MNO,D}$.

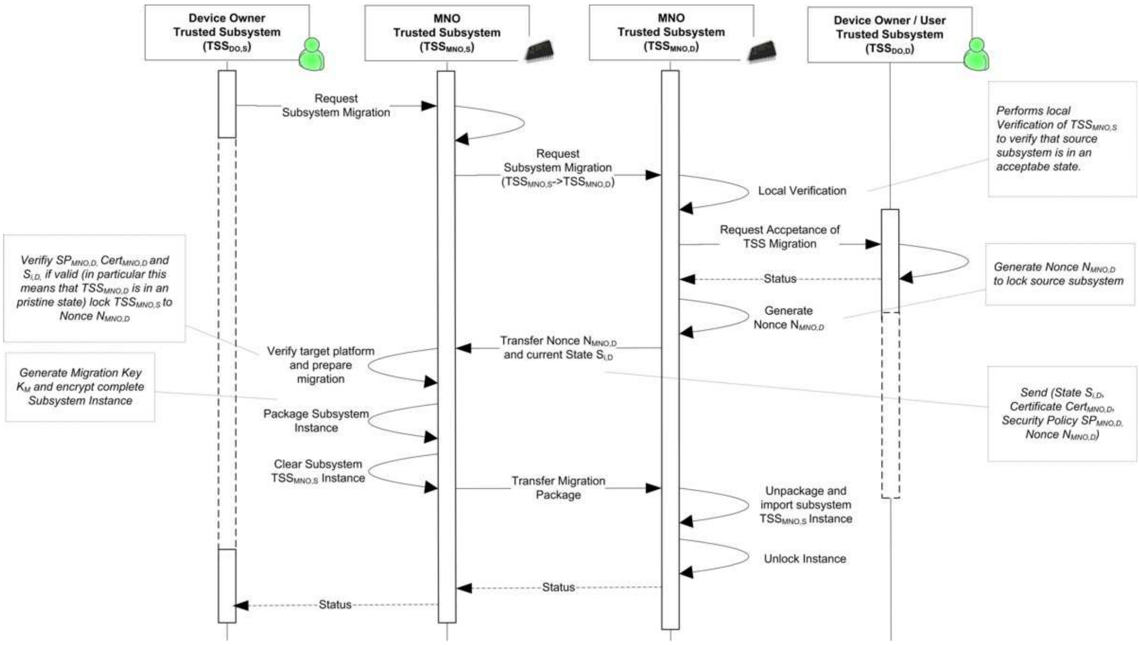

Device Owner Trusted Subsystem ($TSS_{DO,S}$)
MNO Trusted Subsystem ($TSS_{MNO,S}$)
MNO Trusted Subsystem ($TSS_{MNO,D}$)
Device Owner / User Trusted Subsystem ($TSS_{DO,D}$)
Request Subsystem Migration
Request Subsystem Migration ($TSS_{MNO,S}$->$TSS_{MNO,D}$)
Local Verification
Performs local Verification of $TSS_{MNO,S}$ to verify that source subsystem is in an acceptabe state.
Request Accpetance of TSS Migration
Generate Nonce $N_{MNO,D}$ to lock source subsystem
Status
Generate Nonce $N_{MNO,D}$
Verifiy $SP_{MNO,D}$, $Cert_{MNO,D}$ and $S_{i,D}$, if valid (in particular this means that $TSS_{MNO,D}$ is in an pristine state) lock $TSS_{MNO,S}$ to Nonce $N_{MNO,D}$
Transfer Nonce $N_{MNO,D}$ and current State $S_{i,D}$
Verify target platform and prepare migration
Generate Migration Key K_M and encrypt complete Subsystem Instance
Send (State $S_{i,D}$, Certificate $Cert_{MNO,D}$, Security Policy $SP_{MNO,D}$, Nonce $N_{MNO,D}$)
Package Subsystem Instance
Clear Subsystem $TSS_{MNO,S}$ Instance
Transfer Migration Package
Unpackage and import subsystem $TSS_{MNO,S}$ Instance
Unlock Instance
Status
Status

Figure 3.7: *Trusted Subsystem Migration Protocol*

To be able to securely migrate the SRK, we suggest a modification of the current TCG MPWG specification to allow *inter-stakeholder-migration* of a complete isolated key hierarchy. Thus, an isolated key hierarchy is (1) migratable between environments of identical stakeholders, (2) if and only if an entitling security policy on both platforms exists. The advantage of migration between identical stakeholder subsystems, is that the migration process doesn't require a trusted third party. We only involve the owner in combination with local verification mechanisms of the

TSS_{MNO} to migrate the trusted subsystem (including the SRK) to another platform. In the next step. we describe a complete, multilateral and secure migration protocol, which is illustrated in Figure 3.7.

Platform and Protocol Precondition Similar to the preconditions from Section 3.3.1, the trusted mobile platform has carried out the same pre-steps as mentioned above. Furthermore, the MNO has performed an remote take-ownership procedure as described in Subsection 3.3.1.

Protocol Scheme and informal Description At the beginning of the migration protocol, the device owner DO_S of the source platform MTP_S initializes the migration procedure and requests an appropriate migration service of $TSS_{MNO,S}$. Next, the trusted platform MTP_S is instructed by TSS_{MNO} to establish a secure channel to the target platform MTP_D.

$$DO_S \rightarrow TSS_{MNO,S} \quad : \quad init\_migrate\_vsim$$

After the connection is available, $TSS_{MNO,S}$ activates the corresponding migration service of $TSS_{MNO,D}$ to perform the import procedure. Thereon, the target subsystem $TSS_{MNO,D}$ performs a local verification of $TSS_{MNO,S}$. If revoked, it replies with an error message and halts the protocol. Otherwise $TE_{MNO,D}$ requests an confirmation from the local owner DO_D.

$$TSS_{MNO,S} \rightarrow TSS_{MNO,D} \quad : \quad ATTEST_{TSS_{MNO,D}}(S_i)$$

Next, the target subsystem $TSS_{MNO,D}$ generates a random value $NONCE_{MNO,D}$. In order to provide evidence of its trustworthiness, $TSS_{MNO,D}$ sends all necessary information to the source subsystem $TSS_{MNO,S}$. This includes the current state $S_{i,D}$, a certificate of $TSS_{MNO,D}$, security policy $SP_{MNO,D}$ and the nonce $NONCE_{MNO,D}$.

$$\begin{aligned} TSS_{MNO,D} \rightarrow TSS_{MNO,S} \quad : \quad & S_{i,D}, Cert_{TSS_{MNO,D}}, \\ & SP_{MNO,D}, NONCE_{MNO,D} \end{aligned}$$

Having received the target subsystem's message, $TSS_{MNO,S}$ verifies the state of $TSS_{MNO,D}$. If the target system is in a trustworthy state and holds an acceptable security policy and system configuration, the current state of $TSS_{MNO,S}$ is locked to nonce $NONCE_{MNO,D}$.

The $TSS_{MNO,S}$ generates an symmetric migration key K_M, serializes its instance and encrypts it with the migration key, which is bound to an acceptable configuration of $TSS_{MNO,D}$. Next, the key-blob and the encrypted instance are sent to the destination $TSS_{MNO,D}$. In particular, this includes the whole isolated key-hierarchy $K_{MNO,S}$ with $SRK_{MNO,S}$, the security policy $SP_{MNO,S}$, and the required subsystem configuration $SC_{MNO,S}$.

$$\begin{aligned} TSS_{MNO,S} \rightarrow TSS_{MNO,D} \quad : \quad & BIND_{TSS_{MNO,D}}(K_M), \\ & ENC_{K_M}(K_{MNO,S}, SP_{MNO,S}, SC_{MNO,S}) \end{aligned}$$

Finally, the target subsystem $TSS_{MNO,D}$ decrypts the received blob and uses $SRK_{MNO,S}$ as its own SRK. The subsystem verifies the obtained security policy $SP_{MNO,S}$ and the subsystem configuration $SC_{MNO,S}$. With this information, $TSS_{MNO,D}$ rebuilds the internal structure of the source.

3.4 Conceptual Models for Subscriber Authentication

Now, as we are provided with the essential fundament, we are able to develop a model for a virtual software SIM with comparable usage and security characteristics like the traditional smartcard-based solution. In this section, we design the three intergraded conceptual models for authentication in mobile cellular networks using trusted computing.

- Model ”One”: Subscriber Access in mobile cellular Networks based on Trusted Computing with compatibility to GSM - Authentication

- Model ”Two”: GSM-Subscriber Authentication in mobile cellular Networks based on Trusted Computing with Remote Attestation for Basic-Network-Access

- Model ”Three”: Generalized Subscriber Authentication in IT Networks Infrastructures based Trusted Computing with Remote Attestation for Basic-Network-Access

3.4.1 Model "One" - Subscriber Authentication with compatibility to GSM - Authentication

Our proposal for model "One" is straight-forward to actual GSM standard and can be considered as a first step to TC-enhanced mobile network. It enables subscriber access in mobile cellular networks based on Trusted Computing with compatibility to regular GSM authentication. This model could be implemented in conventional GSM clients without any technological changes at the GSM infrastructure and at the GSM authentication protocol. Beside the major task is on its substitution of traditional SIM functionality, this model provides

- user-authenticated access to the subscriber domain,
- binding of $Cred_{vSIM}$ to a trustworthy platform configuration, and
- conformance to GSM authentication standard

The main task of the virtual SIM is to take over the functional range of the SIM card, with no additional duties and responsibilities regarding to GSM 11.11 SIM specification [rGPP97b]. The cryptographic algorithms A3 and A8, responsible for user authentication and key generation, are implemented within the trusted vSIM service.

This model is compatible to conventional subscriber authentication in mobile cellular GSM networks. Figure 3.9 illustrates the protocol sequence, which is described in detail as follows. At first we give information in this section of pre-requirements and conditions of the platform. Afterward we describe the generic protocol.

3.4.1.1 Platform and Protocol Precondition

The MTP has carried out the boot process and has loaded the specific OS and its trusted services. In particular, this also includes the vSIM services $vSIM_{CORE}$ and $vSIM_{MGMT}$. The trusted platform has checked, that the installed hardware and running software, are in a trustworthy state and configuration. It is able to report and attest this state, if challenged by an authorized entity.

3.4.1.2 Protocol Scheme and informal Description

At first we describe the generic protocol. We divide the detailed description into two phases. In phase 1, we construct the protocol for initialization of the services

$vSIM_{CORE}$ and $vSIM_{MGMT}$. Next, in phase 2, we consider subscriber authentication in GSM networks using the vSIM credential $Cred_{vSIM}$.

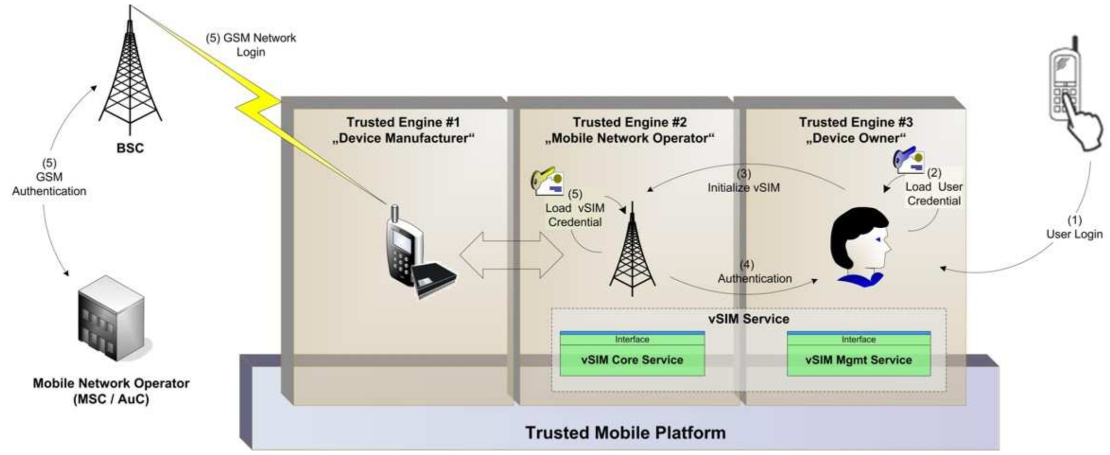


Figure 3.8: *Subscriber Authentication Figure - Model "One"*

Phase 1: “Initialization of vSIM Credentials” First, the user initialize the vSIM services and performs a log-in sequence. He/she sends a unique id ID_U with a proper password CHV_U to the $vSIM_{MGMT}$ service (Step 1), which loads the associated user credential from protected storage (Step 2).

$$U \rightarrow vSIM_{MGMT} \quad : \quad ID_U, CHV_U$$

Afterward, the $vSIM_{MGMT}$ service connects to the trusted interface layer of the $vSIM_{CORE}$ service and sends a vSIM credential initialization request to the $vSIM_{CORE}$ service (Step 3). After having received this request message, $vSIM_{CORE}$ generates a number $RAND_{AUTH}$, randomly chosen from a suitable range and sends this value as an authentication challenge to $vSIM_{MGMT}$, which holds the actual signature keys of U.

$$vSIM_{CORE} \rightarrow vSIM_{MGMT} \quad : \quad RAND_{AUTH}$$

Now, the $vSIM_{MGMT}$ takes the corresponding private portion of the user signature key, signs the challenge $RAND_U$ and sends this value back to the $vSIM_{CORE}$ service (Step 4).

$$vSIM_{MGMT} \rightarrow vSIM_{CORE} \quad : \quad SIGN_U(RAND_{AUTH})$$

Once, the $vSIM_{CORE}$ has received the signed message, it verifies its status. Finally, the $vSIM_{CORE}$ unseals $Cred_{vSIM}$ and initializes the SIM functionality using the $IMSI_i$ and K_i (Step 5).

Phase 2: "Subscriber Authentication" The GSM standard defines its own authentication protocol based on SIM credentials as described in Section 2.1.2. Since the SIM_{CORE} indirectly talks to the MNO, TSS_{DM} must provide a means to relay these messages between the $vSIM_{CORE}$ service and the MNO, this communication should be transparent to this protocol. All relevant communication mechanisms, like cryptographic algorithms A3 and A5, responsible for user authentication and key generation are implemented within the $vSIM_{CORE}$ module. [2]

Following protocol sequence outline the authentication process in GSM networks (Step 5, Step 6): First, the trusted platform initializes the authentication process and sends the $GSMAuthAlgorithm$ command to the $vSIM_{CORE}$ service of TE_{MNO}.

In the next step, the mobile device requests for authentication at the GSM network. Therefor, TSS_{DM} relays $IMSI_i$ (or $TMSI_i$) from $vSIM_{CORE}$ to MNO.

$$vSIM_{CORE} \rightarrow MNO \quad : \quad IMSI_i$$

The MNO generates internally a set of authentication triplets, as described in Section 2.1.2.1. It contains a authentication challenge $RAND_i$, a corresponding session key K_c and a $SRES$. The K_c and the $SRES$ are calculated with the GSM A38 algorithm. The MNO replies to TE_{MNO} by sending the challenge $RAND_i$.

$$MNO \rightarrow vSIM_{CORE} \quad : \quad RAND_i$$

This $RAND_i$ is passed to the trusted $vSIM_{CORE}$ service. Next, it also uses the A3 algorithm together with the key K_i. The output of the algorithm is the challenge response message $SRES^*$. The $vSIM_{CORE}$ sends this $SRES^*$ message to the MNO.

$$vSIM_{CORE} \rightarrow MNO \quad : \quad SRES^*$$

Finally, the MNO compares the $SRES$ with $SRES^*$. If they are equal, the subscriber is authenticated and $vSIM_{CORE}$ also derives the shared session key K_c

[2] Note that the specified GSM algorithms in phase 2 is substitutable by any other authentication algorithm, which requires provisioning of symmetric keys (e.g. one-time-password hashes or symmetric cryptographic keys) and associated attributes in form of a subscriber credential.

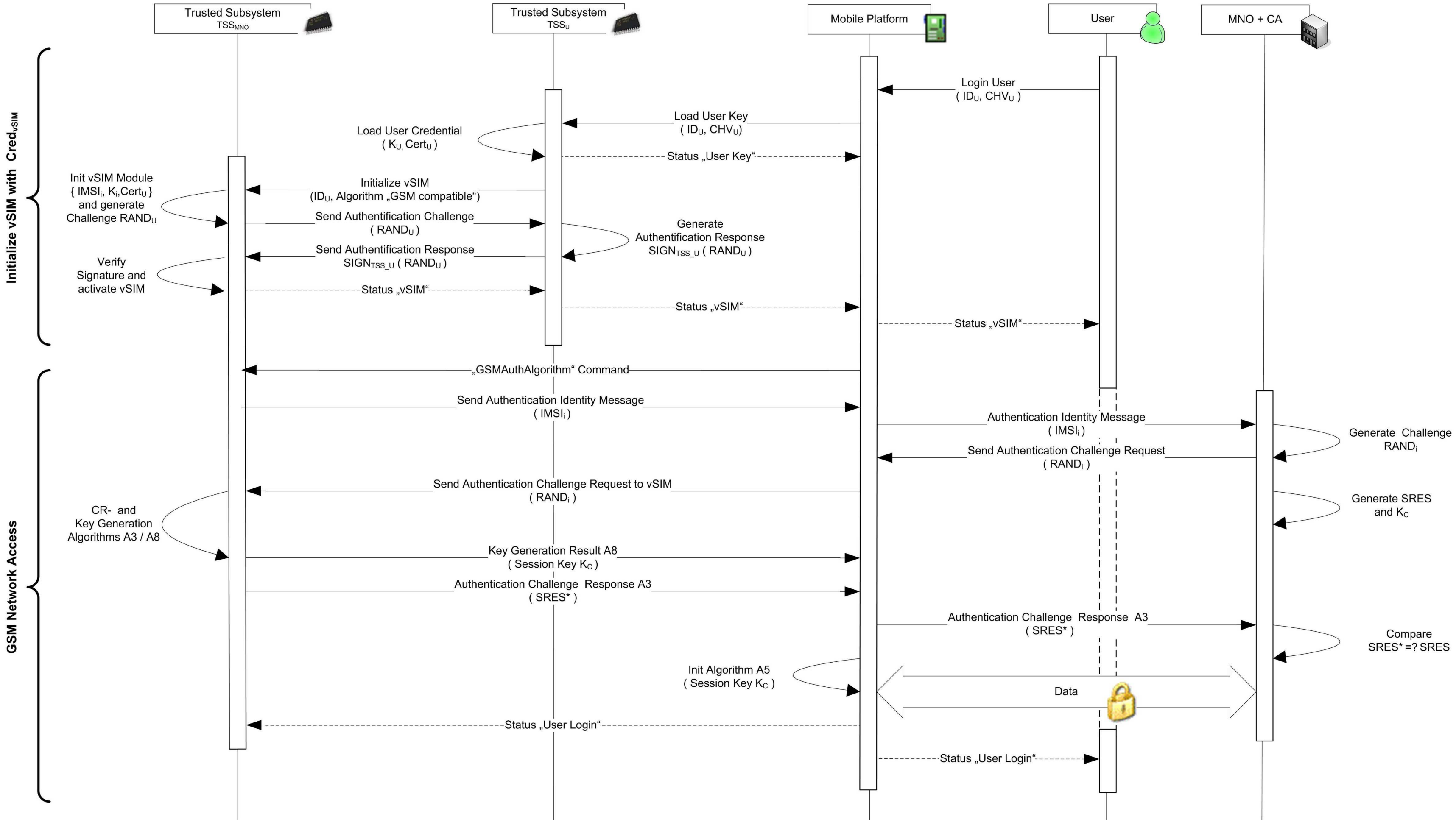

Trusted Subsystem TSS$_{MNO}$
Trusted Subsystem TSS$_U$
Mobile Platform
User
MNO + CA
Initialize vSIM with Cred$_{vSIM}$
GSM Network Access
Login User (ID$_U$, CHV$_U$)
Load User Key (ID$_U$, CHV$_U$)
Load User Credential (K$_U$, Cert$_U$)
Status „User Key"
Init vSIM Module { IMSI$_i$, K$_i$,Cert$_U$ } and generate Challenge RAND$_U$
Initialize vSIM (ID$_U$, Algorithm „GSM compatible")
Send Authentification Challenge (RAND$_U$)
Generate Authentification Response SIGN$_{TSS\_U}$ (RAND$_U$)
Send Authentification Response SIGN$_{TSS\_U}$ (RAND$_U$)
Verify Signature and activate vSIM
Status „vSIM"
Status „vSIM"
Status „vSIM"
„GSMAuthAlgorithm" Command
Send Authentication Identity Message (IMSI$_i$)
Authentication Identity Message (IMSI$_i$)
Generate Challenge RAND$_i$
Send Authentication Challenge Request (RAND$_i$)
Send Authentication Challenge Request to vSIM (RAND$_i$)
Generate SRES and K$_C$
CR- and Key Generation Algorithms A3 / A8
Key Generation Result A8 (Session Key K$_C$)
Authentication Challenge Response A3 (SRES*)
Authentication Challenge Response A3 (SRES*)
Compare SRES* =? SRES
Init Algorithm A5 (Session Key K$_C$)
Data
Status „User Login"
Status „User Login"

Figure 3.9: *Subscriber Authentication Protocol - Model "One"*

3.4.2 Model "Two" - Subscriber Authentication with Remote Attestation for Basic Network Access

In this section, we present a more comprehensive model compared with the precedent one. Additionally to model “One”, we integrate remote attestation for basic network access. A variant of this method has been described in [KS06]. Beside the main task of SIM substitution, it provides

- user-authenticated access to the subscriber subdomain,
- device-authenticated access to a generic domain,
- mutual authentication between the MNO and a trusted mobile device.
- finer-grained functional restriction (e.g. SIM-lock), and
- dynamic down-/upgrade of services

As illustrated in Figure 3.10, all devices inside a generic domain are able to use the generic services of the mobile communication network. A trusted platform which is located in the MNO domain has access to both specific subscriber-authenticated services and generic services. Such generic service, for instance are location-based information or WLAN-based internet services. In case of a mobile phone is located inside the generic domain, it uses a generic credential $Cred_{BASE}$ based on remote attestation mechanisms, to gain basic network access. The assignment to the subscriber domain of MNO is then done by performing a user-specific authentication process using vSIM credentials.

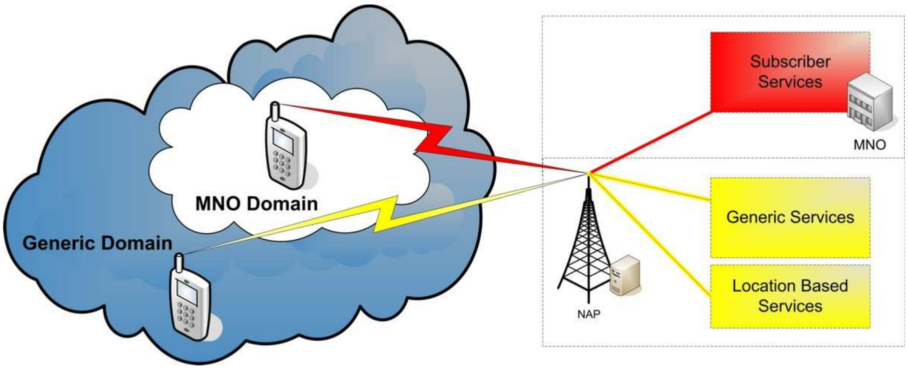


Figure 3.10: *Restricted Subdomain by Trust Credentials*

In model “Two”, we offer two different ways for subscriber authentication. In phase 3, a similar approach to model “One” is described. Alternatively, phase 3”

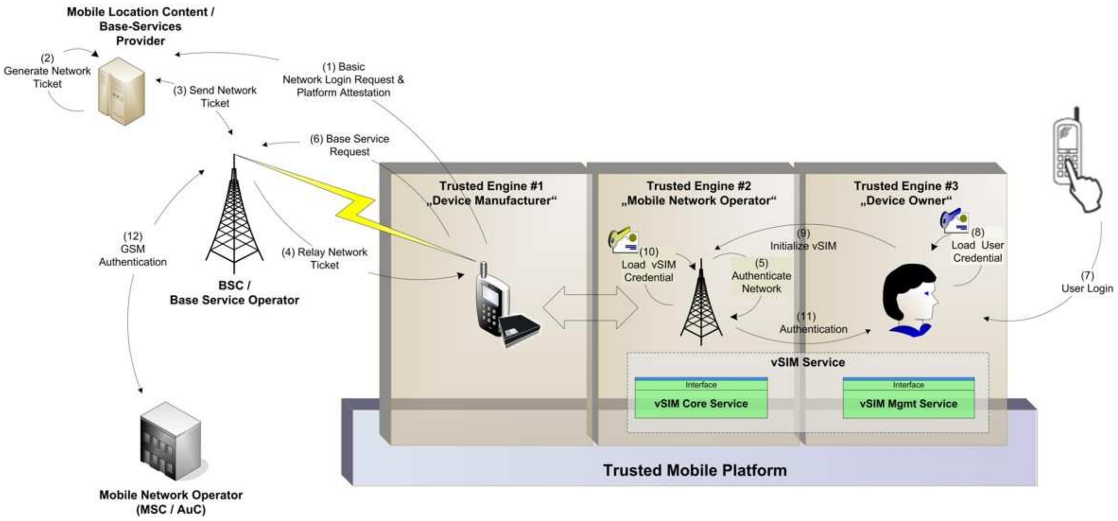


Figure 3.11: *Subscriber Authentication Figure - Model "Two"*

introduces another approach, which is build upon an established trust relationship of the generic domain.

3.4.2.1 Platform and Protocol Precondition

Similar to section 3.3.2, MTP has carried out the boot process and has loaded the specific OS and its trusted services. In particular, this also includes the vSIM services $vSIM_{CORE}$ and $vSIM_{MGMT}$. The trusted platform has checked, that the installed hardware and running software, are in a trustworthy state and configuration. It is able to report and attest this state, if challenged by an authorized entity.

3.4.2.2 Protocol Scheme and informal Description

This protocol description is separated into three phases as illustrated in Figure 3.11. The first phase describes the protocol for basic network access using remote attestation and ticketing, which is inspired by the Kerberos protocol [NYHR05]. In the second phase, the vSIM credential is initialized. Finally, the third phase holds the process for subscriber authentication.

Phase 1: “Basic Network Access” First, the trusted platform initialize the remote attestation and device authentication process. MTP requests the trusted engine TE_{DM} for a platform attestation and device authentication, addressed to the MNO. Then, the trusted engine TE_{DM} performs this request and connects to the corresponding network access point NAP_{MNO} (Step 1). Therefor, the TSS_{DM} generates a random value $RAND_{BASE}$ and and performs a platform attestation. Next, the base authentication service of TE_{DM} sends $RAND_{BASE}$, the attestation data and its certificate $Cert_{DM}$ to the network access point.

$$\begin{aligned} TE_{DM} \rightarrow NAP_{MNO} \quad : \quad & RAND_{BASE}, Cert_{TSS_{DM}} \\ & ATTEST(S_i) \end{aligned}$$

Having received this request, the NAP_{MNO} checks the state of the client machine. If the signed integrity metric of the client platform fails verification or no reference state is found, the NAP_{MNO} aborts the protocol and replies with an error message. Otherwise, the platform passed authentication and is considered as trustworthy.

Afterward, the NAP_{MNO} requests an accredited entity to generates a session key K_{BASE} and a network ticket (Step 2). Such an accredited entity may be an authentication center AUC_{MNO}, which belongs to the mobile network provider MNO. Substantially, the ticket contains the following information:

$$\begin{aligned} Ticket_{BASE} \quad := \quad & \{ID_{MTP}, ID_{NAP}, K_{BASE}, \\ & REALM_{BASE}, LIFETIME_{BASE}\}. \end{aligned}$$

Next, AUC_{MNO} encrypts $Ticket_{BASE}$ with the public (or shared) encryption key K_{NAP} and send both, $Ticket_{BASE}$ and K_{BASE} to the NAP_{MNO} (Step 3), which relays it to the client platform (Step 4). Therefor, the message is bound to the trusted subsystem TSS_{DM} with the corresponding public key $K_{TSS_{DM}}$ and a valid platform state.

$$\begin{aligned} AUC_{MNO} \rightarrow TE_{DM} \quad : \quad & BIND_{K_{TSS_{DM}}}(K_{BASE}), \\ & ENC_{K_{NAP}}(Ticket_{BASE}), \\ & SIGN_{AUC_{MNO}}(RAND_{BASE}) \end{aligned}$$

Once, TSS_{DM} has received the signed message, it verifies the status of the signed $RAND_{BASE}$ (Step 5). If revoked, the subsystem replies with an error message and halts the protocol. Otherwise the AUC_{MNO} is authenticated by the challenge response.

Next, TSS_{DM} decrypts the session key K_{BASE} and sends $ENC_{K_{NAP}}(Ticket_{BASE})$ together with an authenticator A_{MTP} to the NAP_{MNO}. The authenticator A_{MTP}

is composed of its platform identity ID_{MTP}, the current network address $ADDR$, and a timestamp $TIME$.

$$TSS_{DM} \rightarrow NAP_{MNO} \quad : \quad ENC_{K_{NAP}}(Ticket_{BASE}), A_{MTP}$$

After, NAP_{MNO} has received the encrypted ticket, it verifies the embedded information. If the status is valid, the trusted platform is authenticated and access to the generic services is granted.

Phase 2:"Initialization of vSIM Credentials" The initialization of a vSIM credential is performed in Steps 7 - 11 of Figure 3.11. This process is identical to model "One". For a detailed description of the protocol sequence, we refer to Section 3.4.1.2.

Phase 3: "Subscriber Authentication" (Variant 1) Similar to Section 3.4.1.2, this variant performs subscriber access with compatibility to regular GSM authentication. In an additional step, K_{BASE} is substituted by the session key K_c on both sides, the NAP_{MNO} and MTP (Step 12).

However, this approach is optimizable, by embedding the $RAND_i$ already into the encrypted key message from Step 4. In this case, $vSIM_{CORE}$ extracts the $RAND_i$ from this message, calculates the challenge response $SRES$ and sends both to the MNO. The MNO generates internally the expected $SRES$ and the corresponding session key K_c.

At this point a mutual authentication between the AUC_{MNO} and U has been performed. The AUC_{MNO} is authenticated by the signed challenge, obtained in step 3.1. On the other hand, the user has prooven its identity by $SRES$. The authentication between NAP and U is implicitly proven by a valid communication key K_c.

If an explicit authentication of these entities is required, some additional steps have to be carried out. The NAP authenticates itself to the platform by the following steps. First the NAP extracts the timestamp from the authenticator A_U. Next, NAP increments the value and encrypts it with the shared communication key K_c (or a derivation of it). Finally, it sends the message back to the trusted platform.

Phase 3": "Subscriber Authentication" (Variant 2) Alternatively to phase 3, the following protocol sequence describes the authentication process in variation to standard GSM authentication.

Here, we envisage a slightly modified authentication method, which offers significant security enhancements across the entire PLMN. In particular protocol flaws in *Signaling System 7 (SS7)* could be bypassed.

It takes advantage of the former negotiated information from the device authentication in phase 1. In conventional GSM Infrastructures an authentication triplet is sent over the SS7 network. This triplet contains of a challenge $RAND$, the correct response $SRES$, and the communication key K_c.

While initial access to the mobile cellular network with the communication key K_{BASE} is still established, a renewal of this key is discretionary. In particular, embedding a communication key K_c within this token is not necessary. However, a specific realm and accordingly other specific service information has to be sent to the network access point NAP_{MNO}.

We note that this approach avoids transmission of unprotected communication keys K_c across the PLMN infrastructure. The main idea behind this model is to use the still established communication channel between NAP_{MNO} and MTP, which is protected by K_{BASE}. Instead of performing a renewal of the communication key, the MNO only sends a service update message to the respective network access point NAP.

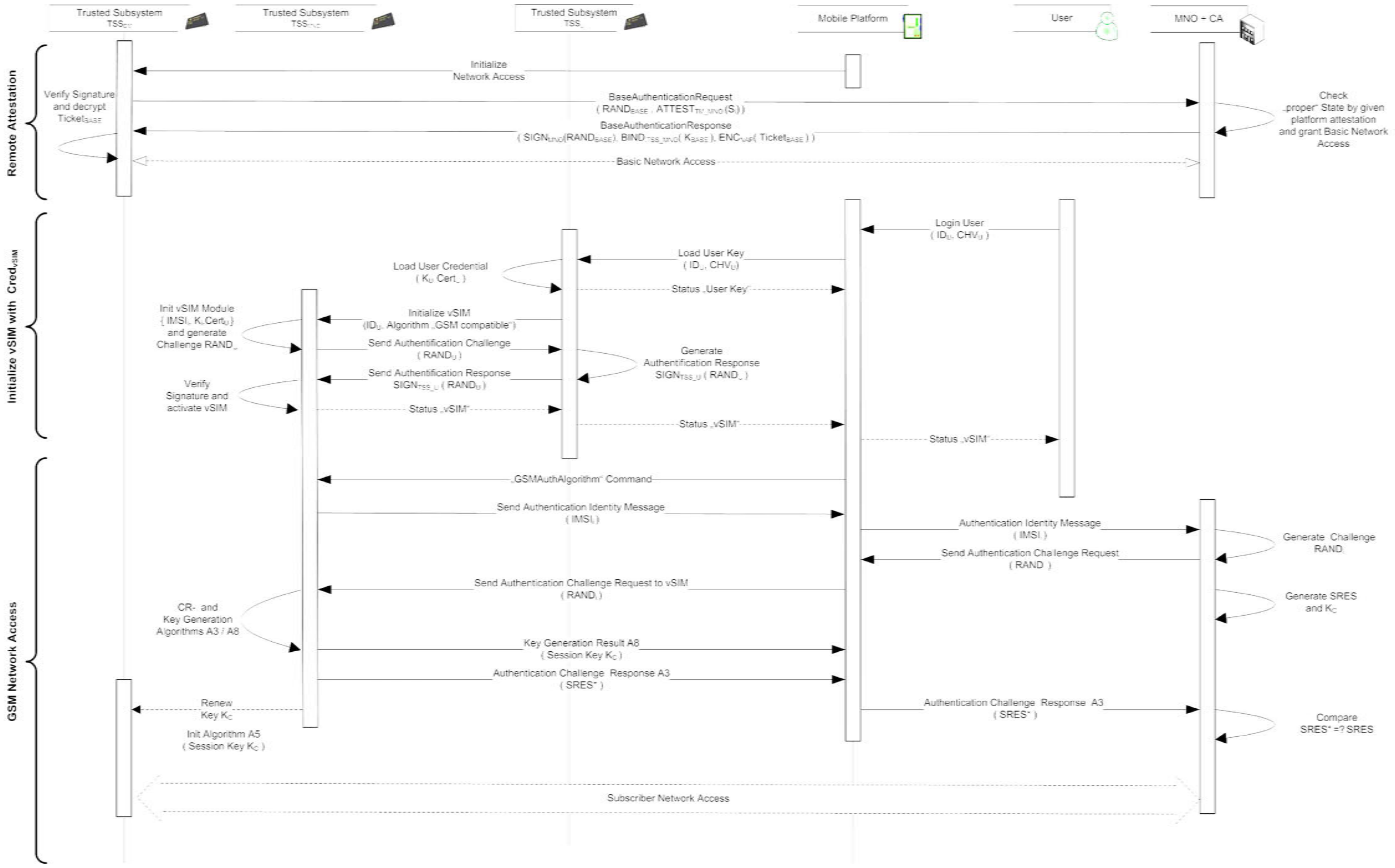

Trusted Subsystem TSS$_{DM}$
Trusted Subsystem TSS$_{vSIM}$
Trusted Subsystem TSS$_U$
Mobile Platform
User
MNO + CA
Remote Attestation
Initialize Network Access
Verify Signature and decrypt Ticket$_{BASE}$
BaseAuthenticationRequest (RAND$_{BASE}$, ATTEST$_{TM\_MNO}$(S$_i$))
Check „proper" State by given platform attestation and grant Basic Network Access
BaseAuthenticationResponse (SIGN$_{MNO}$(RAND$_{BASE}$), BIND$_{TSS\_MNO}$(K$_{BASE}$), ENC$_{UAP}$(Ticket$_{BASE}$))
Basic Network Access
Initialize vSIM with Cred$_{vSIM}$
Login User (ID$_U$, CHV$_U$)
Load User Key (ID$_U$, CHV$_U$)
Load User Credential (K$_U$ Cert$_U$)
Status „User Key"
Init vSIM Module { IMSI$_i$, K$_i$Cert$_U$} and generate Challenge RAND$_U$
Initialize vSIM (ID$_U$, Algorithm „GSM compatible")
Send Authentification Challenge (RAND$_U$)
Generate Authentification Response SIGN$_{TSS\_U}$ (RAND$_U$)
Send Authentification Response SIGN$_{TSS\_U}$ (RAND$_U$)
Verify Signature and activate vSIM
Status „vSIM"
Status „vSIM"
Status „vSIM"
GSM Network Access
„GSMAuthAlgorithm" Command
Send Authentication Identity Message (IMSI$_i$)
Authentication Identity Message (IMSI$_i$)
Generate Challenge RAND$_i$
Send Authentication Challenge Request (RAND)
Send Authentication Challenge Request to vSIM (RAND$_i$)
Generate SRES and K$_C$
CR- and Key Generation Algorithms A3 / A8
Key Generation Result A8 (Session Key K$_C$)
Authentication Challenge Response A3 (SRES*)
Authentication Challenge Response A3 (SRES*)
Compare SRES* =? SRES
Renew Key K$_C$
Init Algorithm A5 (Session Key K$_C$)
Subscriber Network Access

Figure 3.12: *Subscriber Authentication Protocol - Model "Two"*

3.4.3 Model "Three" - Generalized Subscriber Authentication in Network Infrastructures

The following section introduces an authentication model that captures the different aspects of the precedent models from a more abstract point of view. Here, we present a proposal for user- and device authentication using vSIM credentials in generic network infrastructures. This model is based on Trusted Computing and like the predecessor it supports remote attestion and mutual authentication.

In contrast to the previous models, we are using more generalized assumptions and specifications. This concerns particularly the structure and abilities of a vSIM credential $Cred_{vSIM}$ and the trusted $vSIM_{CORE}$ services.

A vSIM credential $Cred_{vSIM}$ is a identity-based identifier that can be used to authenticate a subscriber. It has a unique ID_U of a user U and at least one encryption-based (e.g. symmetric or asymmetric keys) or non-encryption-based (e.g. one-way hash chain) information. Only authorized subjects can create, read or modify to the content of a $Cred_{vSIM}$. A $Cred_{vSIM}$ may hold additional information of a device identity or a set of valid realms

A trusted platform holds a $vSIM_{CORE}$ service, running in an protected environment. $vSIM_{CORE}$ is responsible for the vSIM functionality. In particular, this service implements for the core authentication mechanisms. A specific implementation of mechanisms or protocols depend on the use-case. A $vSIM_{CORE}$ service is able to import (or use external) trusted functionality. Furthermore it holds at least one vSIM credential $Cred_{vSIM}$.

3.4.3.1 Platform and Protocol Precondition

MTP has carried out the boot process and has loaded the specific OS and its trusted services. In particular, this also includes the vSIM services $vSIM_{CORE}$ and $vSIM_{MGMT}$. The MTP has checked, that the installed hardware and running software, are in a trustworthy state and configuration. It is able to report and attest this state, if challenged by an authorized entity.

3.4.3.2 Protocol Scheme and informal Description

Similar to model "Two", the protocol description of this generalized approach is separated into three phases. This generalized setting for subscriber authentication is described by the following sequence of steps.

Phase 1: "Remote Attestation" In this phase, the remote attestation and device authentication are performed as described in Subsection 3.4.2.2. In this general case, the network entities of the MNO are substituted by adequate entities from the generic network infrastructure. Such *adequate entity* may be an authentication server AS within this network.

Phase 2: "Initialization of vSIM Credentials" Initialization of the vSIM services and a vSIM credential are performed as described in Subsection 3.4.1.2. However, this setting is based on the generalized assumption from 3.4.3. Thereby, it provides a wide basis for different types of authentication methods and protocols.

Phase 3: "Subscriber Authentication" The process of subscriber authentication aims to authenticate and authorize a given subscriber to a specified services. In the previous models, the subscriber authentication protocols are based upon a shared secret, which is embedded into the vSIM credential $Cred_{vSIM}$, and a challenge response procedures for authentication. In this generic approach these limitations are in-existent.

Figure 3.13 shows an example of a simple protocol for subscriber authentication with digital signatures based on this generalized model. Here, a random value $RAND_{SRV}$ is used to request an service upgrade at the AS. The TE_{OE} extracts $RAND_{SRV}$ from the $Ticket_{BASE}$, which was obtained in phase 1. Now, the TE_{OE} builds the authentication response $XRES^{*}_{SRV}$ and signs the $RAND_{SRV}$ with its private signature key $K^{priv}_{TM_{AS}}$. Together with a UID and a service identifier SRV, this signature $XRES^{*}_{SRV}$ is sent to the AS.

After having received this message, AS verifies signature $XRES^{*}_{SRV}$. If the signature is valid, the trusted platform is authenticated and a service upgrade will be performed.

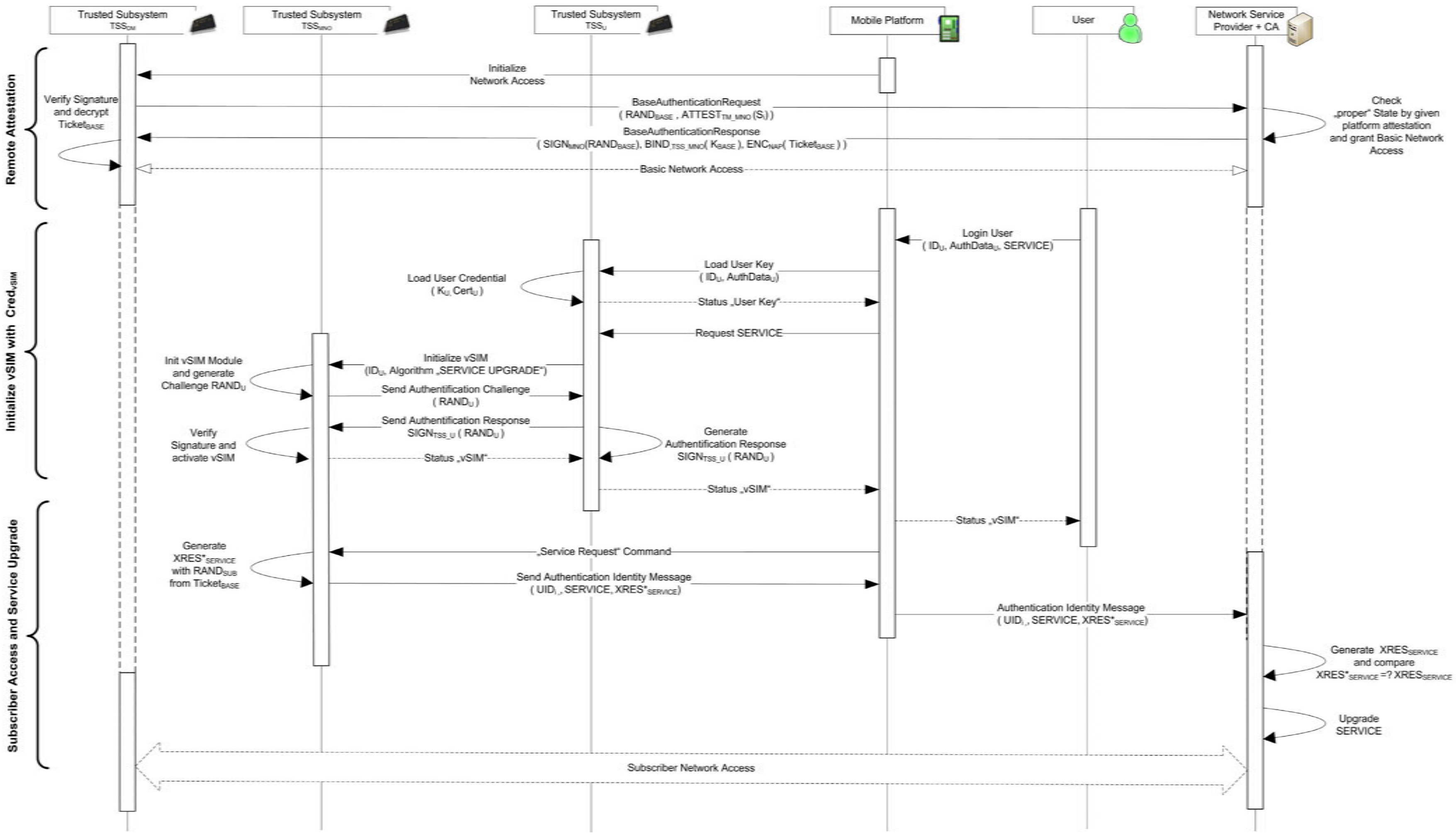

Trusted Subsystem TSS_{CM}
Trusted Subsystem TSS_{MNO}
Trusted Subsystem TSS_{U}
Mobile Platform
User
Network Service Provider + CA
Remote Attestation
Initialize vSIM with $Cred_{vSIM}$
Subscriber Access and Service Upgrade
Initialize Network Access
Verify Signature and decrypt $Ticket_{BASE}$
BaseAuthenticationRequest ($RAND_{BASE}$, $ATTEST_{TM\_MNO}(S_i)$)
BaseAuthenticationResponse ($SIGN_{MNO}(RAND_{BASE})$, $BIND_{TSS\_MNO}(K_{BASE})$, $ENC_{NAP}(Ticket_{BASE})$)
Check „proper" State by given platform attestation and grant Basic Network Access
Basic Network Access
Login User (ID_U, $AuthData_U$, SERVICE)
Load User Credential (K_U, $Cert_U$)
Load User Key (ID_U, $AuthData_U$)
Status „User Key"
Request SERVICE
Init vSIM Module and generate Challenge $RAND_U$
Initialize vSIM (ID_U, Algorithm „SERVICE UPGRADE")
Send Authentification Challenge ($RAND_U$)
Send Authentification Response $SIGN_{TSS\_U}$ ($RAND_U$)
Generate Authentification Response $SIGN_{TSS\_U}$ ($RAND_U$)
Verify Signature and activate vSIM
Status „vSIM"
Status „vSIM"
Status „vSIM"
Generate $XRES^*_{SERVICE}$ with $RAND_{SUB}$ from $Ticket_{BASE}$
„Service Request" Command
Send Authentication Identity Message (UID_i , SERVICE, $XRES^*_{SERVICE}$)
Authentication Identity Message (UID_i , SERVICE, $XRES^*_{SERVICE}$)
Generate $XRES_{SERVICE}$ and compare $XRES^*_{SERVICE}$ =? $XRES_{SERVICE}$
Upgrade SERVICE
Subscriber Network Access

Figure 3.13: *Subscriber Authentication Protocol - Model "Three"*

3.5 Prototypical Implementation of the vSIM Architecture

The prototypical implementation of the trusted engines TE_{DM}, TE_{MNO} and TE_U, and the specified vSIM services will be realized as an extension to an existing trustworthy operating platform. Therefor, we initially introduce to a high-level design of the vSIM architecture in Subsection 3.5.1. Then, in Subsection 3.5.2, we give a short analysis of the significant vSIM platform components.

3.5.1 Overview of the vSIM Platform Design

The high-level design of the proposed vSIM architecture is based on a trustworthy operating platform such as *EMSCB/Turaya* [EMS], or alternatively *IBM sHype* [SVJ+05]. The technical framework consists of four layers:

- *Hardware Layer*,
- *Virtualization Layer*,
- *Trusted Software Layer*, and finally
- *Compartment and Application Layer*.

The platform executes a legacy operating system in coexistence with a running instance of the security architecture. The latter controls a virtual machine with several trusted engines and services compliant to the TCG requirements [Tru06a, Tru06b]. Each trusted engine implements the vSIM services and applications on behalf of a specific stakeholder. The trusted engines are executed within isolated execution environments on top of the security kernel or nested inside TE_{DM}. Figure 3.14 illustrates the platform design of the vSIM architecture.

3.5.2 Analyis of the Platform Components

In the following subsection, we inspect these four platform layers of the vSIM architecture and describe its embedded components.

3.5.2.1 Hardware Layer

The *Hardware Layer* holds the generic hardware components of the computing platform and an additional generic MTM, as described in 2.2.5. The MTM acts as a dedicated master trust-anchor for the complete trusted mobile platform. Although,

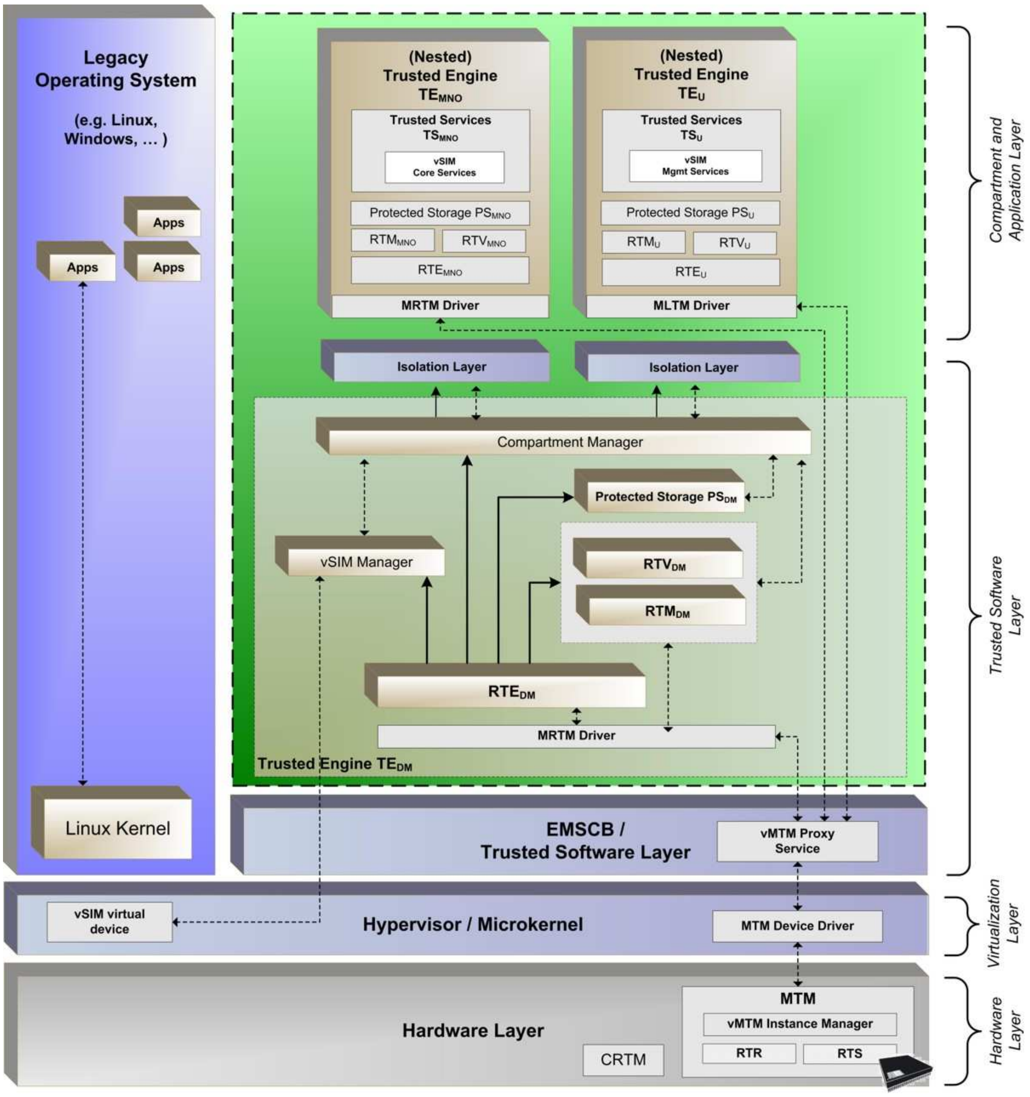


Figure 3.14: *vSIM Architecture on EMSCB/Turaya*

it is favorable to implement this architecture upon a slightly modified MTM, we focus on a (contemporary practicable) solution based on a software-based generic MTM emulation in conjunction with a standard TPM v1.2. The emulator implements a software-based generic MTM emulator based on our proposal in Section 2.2.5. It provides an interface to a standard TPM 1.2 and adapts the hardware-based TPM with MTM functionality compliant to the *TCG MPWG Reference Architecture*.

3.5.2.2 Virtualization Layer

Generic hardware abstraction between the physical hardware and the *Trusted Software Layer* is provided by the *Virtualization Layer*. This layer implements a fully functional *MTM Host Driver and Device Driver* for a dedicated generic MTM and a *virtual vSIM Device* for communication. Furthermore, it is responsible for instantiation of both, the trusted software layer and the legacy operating system.

- *Virtual vSIM Device* offers an exclusive vSIM communication interface to external entities (e.g. legacy OS). It is implemented by a char-device that provides an external accessible communication channel to the vSIM Manager.

- *MTM Host Driver and Device Driver* implements a full functional software driver that allows interaction with the MTM hardware device and/or the MTM Emulator. It holds the capability to channel the received communication messages to the respective MTM instance.

An excellent fundament is offered by both, the EMSCB project [EMS, PER] and alternatively the sHype project [SVJ$^+$05]. Currently, the EMSCB is based upon a microkernel of the L4-family [BBH$^+$98] that supports instantiation of L4 applications and L4Linux [Hoh96] compartments. On the other hand, sHype is based on the XEN hypervisor [DFH$^+$03], which is able to virtualize a legacy operating system (e.g. Windows, Linux). In general, each solution provides mechanisms for resource management, inter-process-communication, virtual machines, memory management and scheduling.

3.5.2.3 Trusted Software Layer

The *Trusted Software Layer* provides security functionality and is responsible for isolation of embedded applications and software compartments. It implements a set of security services (e.g. trust manager, compartment manager, protected storage manager), which are required by the RTR and RTV, *Protected Storage* and *Compartment Manager* of TE_{DM}. Therefore, it is reasonable to build the significant parts of the device manufacturer engine TE_{DM} within this layer. The components of this layer are:

- *vMTM Proxy Service* is a service that acts as a mediator between the *MTM Device Driver* and a *MTM Client Driver.* When an embedded trusted engine executes a MTM command, the message is routed through this proxy to the requested dedicated *MTM Host Driver.* The proxy will transmit and channel the data to an associated vMTM instance.

- *vMTM Client Driver* implements a restricted software driver that allows interaction with a *vMTM Proxy Service.* It either implements $MRTM$ or $MLTM$ capability. For this reason, a trusted engine is only able to execute its dedicated commands as specified by the TCG MPWG.

- *Roots of Trust (RTE, RTV and RTM* are a set of allocated trusted resources acting on behalf of the stakeholder DM as described in Section 2.2.4.2. In particular, these components are the *Root of Trust for Enforcement*, the *Root of Trust for Verification* and the *Root of Trust for Measurement.*

- *Protected Storage PS* provides storage mechanisms of TE_{DM} as described in Section 2.2.3.4. It acts as a storage manager which uses the protected capabilities and shielded locations of its dedicated vMTM instance $vMTM_{DM}$.

- *Trusted Engines Compartment Manager* controls the instantiation and update of trusted compartments. The main tasks of the compartment manager is to offer a minimal set of required TE_{DM} functionality to the vSIM services, and to route communication messages to their destination. Furthermore, it enforces the isolation of the dedicated trusted engines TE_{MNO} and TE_U that can only be accessed and manipulated by authorized entities.

- *vSIM Manager* represents an interface to the $vSIM_{CORE}$ and $vSIM_{MGMT}$ services and is responsible for the communication with the underlying architecture and external entities.

3.5.2.4 Compartment and Application Layer

The *Compartment and Application Layer* instantiates a set of isolated Trusted Engines TE_{σ}. Each engine embeds the required allocated trusted resources and protected storage mechanisms as well as the dedicated vSIM services.

- *Trusted Engine* TE_{σ} are implemented as parallel and isolated Linux ([DFH^{+}03, SVJ^{+}05]) or L4Linux ([EMS, BBH^{+}98, Hoh96]) compartments on behalf of different stakeholders. Each trusted engine is fully equipped with a *MTM Client Driver*, allocated *Roots of Trust* (*RTE*, *RTV* and *RTM*) and *Protected Storage* functionality as described above.

- *VSIM Core Service* $vSIM_{CORE}$ holds the trusted vSIM core services of TSS_{MNO} In particular, it implements the relevant protocol algorithms of MNO. This includes the subscriber authentication from Section 3.4, and the deployment and management protocols from Section 3.3.

- *VSIM Mgmt Service* $vSIM_{MGMT}$ The $vSIM_{CORE}$ holds the trusted vSIM management services of TSS_U. It implements the relevant protocol algorithms of U. This includes the user portion of the *Subscriber Authentication* protocols from 3.4 and the protocols for *Subscriber Enrollment and vSIM Credential Roll-Out* 3.3.2.

Chapter 4

Benchmark Analysis and Evaluation

The purpose of this *Benchmark Analysis and Evaluation* is to provide a review of the proposed vSIM architecture. A pre-screening of the aspired and existent architectures was conducted to identify the different possibilities for subscriber authentication in mobile cellular networks. We have identified four solutions:

- a single-trust anchor architecture using conventional SIMs,
- a dual trust anchor architecture using conventional SIMs,
- a single trust-anchor architecture using virtual SIMs, and finally
- a client-server architecture using remote SIMs.

It was agreed that the *Single Trust-Anchor Architecture using virtual SIMs* approach turns out as a suitable and sustainable solution, that is able to compete with the SIM-based solutions. This chapter substantiates this assumption.

The evaluation shows that our vSIM architecture can be used to substitute a SIM card. Therefore, we consider the following areas of interest derived from Section 1.2: (1) Security, (2) Cost-effectiveness, (3) Flexibility and Scalability, (4) Portability and Mobility, and (5) Usability, Compatibility and Acceptance.

This chapter is subdivided into three sections. In Section 4.1, we firstly provide a security analysis of the significant protocols of the vSIM architecture. Afterwards, we analyse the further criteria in Section 4.2. Finally, in Section 4.3, a comparison of this architecture with the other solutions is given.

4.1 Security Analysis

An important factor of the security analysis is the protection of the vSIM architecture against attacks. In this context, we have to inspect the following objectives: (1) protection of a MTM, (2) protection of the trusted compartments by the trusted operating system, and (3) the security analysis of the protocols.

4.1.1 Protection Mechanisms of a MTM

Since the MTM is the underlying trust-anchor which provides evidence of the trustworthiness of the vSIM architecture and the associated trusted subsystems TSS_σ, it is required that the MTM itself must be reasonably secured and protected from attacks.

Therefore, the *Common Criteria Protection Profile* makes (still relevant) assertions of the protection requirements of a TPM 1.1 against software and hardware attacks [Tru02a, Tru02b]. This *Protection Profile* stipulates only a limited physical protection of a TPM. However, it is reasonable that the protection against hardware attacks depends on the intended purpose. As a consequence, some TPM/MTM hardware implementations will have stronger physical protections than other [Hew07]. Thus, we assume that a vSIM architecture is built upon a MTM with equal security related characteristics and properties from Section 2.1.3.1, like a conventional SIM card.

4.1.2 Protection Mechanisms of a Trustworthy Operating system

In general, a *Trustworthy Operating System* provides fundamental capabilities that meets the security related requirements from Section 3.2.3 and *TCG MPWG Reference Architecture.* In particular, these operating systems support *Protected Storage*, a tamper-resistant *Isolated Execution Environment*, *Secure Channel* and *Access Control and Authentication.*

4.1.3 Security Analysis of the Protocols

The protocol analysis considers the different security requirements from Section 3.2.3 according to the proposed protocols. In this context, we inspect the provided security mechanism while a vSIM Credential is (1) transfered to the vSIM Container, (2) stored and executed on the mobile trusted platform or (3) transferred between environments of authorized subjects.

4.1.3.1 Protection of the vSIM Credential while on transit

The initial analysis focuses on the confidentiality and integrity protection of the vSIM credential while in transit from the MNO to the destination platform (MTP). This analysis concerns the protocol *Subscriber Enrollment and Credential Roll-Out*, which we have discussed in Section 3.3.2.

An adversary might be able to eavesdrop and modify the protocol messages between MTP and MNO. In order to circumvent resulting attacks, the $Cred_{vSIM}$ is encrypted with a session key K_S. This session key is bound to a specific state of the destination platform using the binding key K_{MNO}^{priv}. An adversary would have to extract K_{MNO}^{priv} from trusted subsystem TSS_{MNO} to recover K_S. Moreover, a HMAC is computed on the digital representation of $Cred_{vSIM}$ using this session key (or a derived/associated integrity key). Therefore, it offers a sufficient degree of confidentiality and integrity, because the decryption is only feasible by an authorized and trusted MTP. If the $Cred_{vSIM}$ was modified by an adversary while transmission to the MTP the computation of the HMAC will fail. However, K_S must also be securely managed and protected by MNO and TSS_{MNO}, at least to the same degree as $Cred_{vSIM}$ itself is protected.

4.1.3.2 Protection of the vSIM Credential while in Storage

The confidentiality and the integrity of the vSIM Credential $Cred_{vSIM}$ while in storage are protected by the mechanisms of protected storage specified in *TCG MPWG Reference Architecture.*

In order to prevent unauthorized access to the vSIM Credential and the associated binding keys, further measures are taken. First, the private portion of the binding key $BK_{TSS_{MNO}}$ is bound to a specific platform configuration of TSS_{MNO} such that this key not loadable until the current environment configuration matches to which the private key was bound. Second, the decryption of $Cred_{vSIM}$ is associated with a successfully challenge-response from $vSIM_{MGMT}$ using a valid signing key SK_{TSS_U}. In conjunction with this, 20 bytes of authorization (see CHV in 2.3) data must be stored with the private signing key SK_U^{priv}. This meets the *Strong Authentication* and *Portability and Mobility* requirements from 2.1.3.1.

4.1.3.3 Protection of the vSIM Credential during execution

The confidentiality and the integrity of the vSIM Credential $Cred_{vSIM}$ while in storage are protected by the mechanisms of protected execution environment. The vSIM services which are running within that environment of TSS_{MNO} and TSS_U can not be read or manipulated by an unauthoritzed entity.

Nevertheless, the vSIM Credential is only protected by software mechanisms after the object is loaded successfully into the execution environment. In terms of security, a dedicated hardware-based protection of security-sensitive data is stronger than the software-based solution. And consequently, if the protection of the protected execution environment fails, the vSIM Credential may be accessible to an adversary. In order to circumvent this security flaw, a dedicated MTM could also be equipped with A3/A8 GSM computation engine, as shown in Figure 4.1. Consequently, the secret individual key K_i would never leave the hardware protected environment of the MTM.

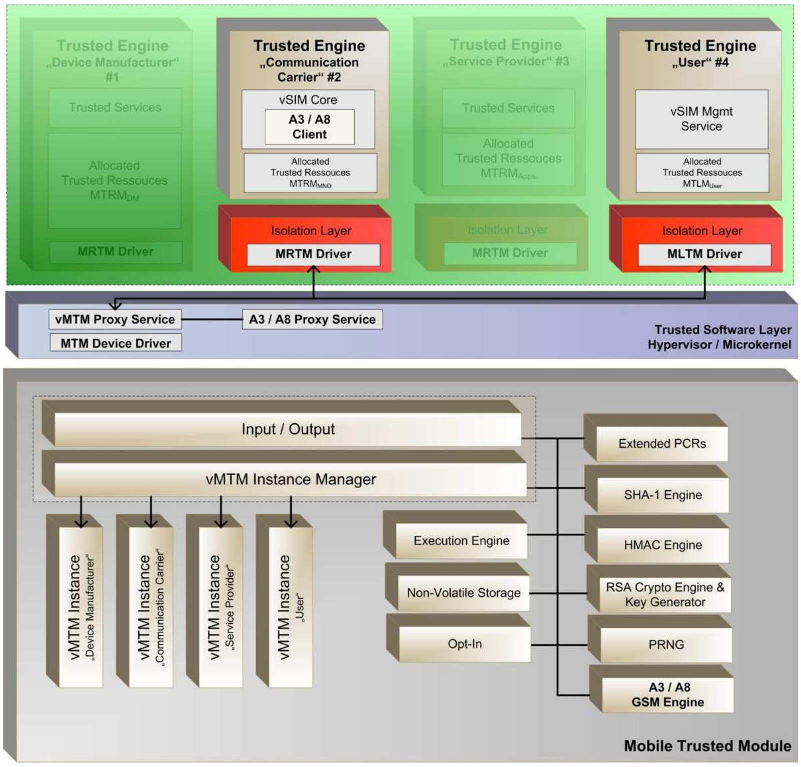


Figure 4.1: *Generic MTM architecture with an additional A38 engine*

A $vSIM_{CORE}$ service sends its GSM authentication request to a *A3/A8 Proxy Service*, which is a part of the *vMTM Proxy Service* within the *Trusted Software*

Layer. This service forwards the messages from the $vSIM_{CORE}$ to the dedicated MTM instance.

4.2 Benchmark Analysis

The analysis of the other benchmark criteria and conditions of Section 1.2 is another important issue for evaluation of the vSIM architecture. For this reason, we have to inspect the following objects: (1) Cost-effectiveness, (2) Flexibility and Scalability, (3) Portability and Mobility, (4) Usability, (5) Compatibility and (6) Acceptance.

4.2.1 Cost-effectiveness

The major aspect of our solution concerns cost effectiveness. Our approach reduces the production and logistic costs, regarding to the two-trust-anchor solution while keeping an adequate level of security and trustworthiness. Unfortunately, it is difficult to determine an exact and sustainable cost analysis, since these costs are treated strictly confidential by device manufacturers.

- *Manufacturing and Production Costs:* The main component that affects the costs of a security chip is naturally the micro controller. With the proposed solution, the MNO is able to use the already installed MTM for its purpose of subscriber authentication and protected storage. The mobile device is not necessarily equipped with a conventional SIM card.

- *Logistic Costs:* As the vSIM Credentials are completely implemented as a software object, it implies that a $Cred_{vSIM}$ can transferred from the MNO to the vSIM Container via an arbitrary network connection. For this reason, a MNO can reduce and minimize effective costs of the logistic process.

4.2.2 Flexibility and Scalability

The vSIM service architecture for subscriber authentication offers flexibility and scalability to both, the user and the MNO. The next paragraphs shortly describes the benefits concerning this issue.

Parallel vSIMs in a single Mobile Device The proposed vSIM architecture supports the instantiation and usage of many parallel vSIM services. It conveniently allows a subscriber to make and receive calls on different $MSISDN$ numbers with different subscriber identities, from one single handset without switching a physical SIM card.

Online Registration and Roll-Out of vSIM Credentials A device owner initiates and controls the subscriber registration, vSIM credential roll-out and installation process as described in Subsection 3.3.2. This is potentially done by using a web-based interface giving subscribers full control to decide how they would like to register themself and receive their vSIM Credential on their mobile devices.

Network-based Migration of a vSIM Container and vSIM Credential The vSIM Container is migrateable between different devices holding the same stakeholder and a suitable platform configuration as described in Section 3.3.3. A device owner/user can migrate a vSIM Container and the vSIM Credential using an arbitrary network connection. This implies migration over an existing PAN (e.g. Bluetooth) as well as an established connection to the destination device over the internet.

Remote Update of Services, Firmware and Applications Since the MNO is able to access its dedicated compartment TSS_{MNO}, arbitrary data object can be installed or modified by an authorized entity. Thus data object, for instance are either vSIM service- and algorithm updates, or new firmware and applications for the mobile device.

Dynamic Up- and Downgrade of Network Services The vSIM architecture offers the potential for dynamic up-/downgrade of network services and finer-grained functional restriction. These functionalities provide mobile communication services as well as location-based services, like content services or mobile payment services. Since some of these services require a higher level on platfrom integrity and conformity (e.g. online-banking transaction) the vSIM architecture is able to provide evidence of the platform state. In particular, the approach for subscriber authentication in Model "Two" (see 3.4.2) allows to add or remove functionality and services easily.

4.2.3 Portability and Mobility

The vSIM architecture allows subscribers to use a $Cred_{vSIM}$ with arbitrary trusted mobile platform. With the identified protocols for deployment and management from Section 3.3, a vSIM credential is removeable and portable to other devices. Hence, it enables a subscriber to use its credential with other devices, and vice versa.

Using two different vSIM compartments on behalf of the stakeholder MNO and U, the vSIM architecture enables explicitly to differentiate between device and subscriber identity. A MTP and a subscriber have their own identity with different intended usage characteristics.

4.2.4 Usability, Compatibility and Acceptance

Determining usability and compatibility of a system is an important part, since it finally leads to the acceptance of the proposed vSIM architecture. For this reason, we have designed the subscriber authentication protocols as well as the protocols for deployment and management with a high level of compatibility and usage characteristics to current GSM standard. This was reached by using algorithms, commands and grammar compliant to GSM 11.11 and GSM 11.14 standard from Section 2.1.3. The proposed model “One” for subscriber authentication, in particular is compliant to this GSM standard.

4.3 Comparison and Evaluation

A comparison of the different architecture as well as the proposed models for subscriber authentication from Section 3.4 has been done using the set of benchmarks from Section 1.2. Table 4.2 illustrates the results.

In the present thesis, we have substantiated our choice and have proven that a sufficient level of each criteria is reached. In this context, it is reasonable to decide in favor of the proposed *Single Trust-Anchor Architecture using virtual SIMs for Subscriber Authentication* in conjunction with “Model One” based on a generic MTM emulation. The tradeoff and fundamental design decisions are primarily based on the cost-value ratio by reaching an adequate level of security and better deployment and management functionality in comparison with the other competing solutions.

Although, the proposed Models "Two" and "Three" offer more flexibility and security by embedding platform attestation. The decision is resulted by the compatibility to current GSM standard. Thus, a higher level of acceptance from both, the mobile network operator and the device manufacturer is reachable.

			vSIM Architecture			
Architecture / Criteria	SIM	SIM / MTM	Model "One"	Model "Two"	Model "Three"	C/S SIM
Security						
Protected Storage	X	X	X	X	X	X
Isolated Execution Environment	X	X	X	X	X	X
Secure Channel	X	X	X	X	X	X
Access Control and Authentication	X	X	X	X	X	X
Remote- / Platform Attestation				X	X	X
Trustworthy Operating Client Platform		X	X	X	X	
Resistant against Software Attacks	X	X	X	X	X	X
Resistant against Hardware Attacks	X	X	X	X	X	X
Mutual Authentication	X	X		X	X	
Hardware Protected K_i	X	X	-/X	X	-/X	X
Cost Effectiveness						
Reduction of Manufacturing Costs			X	X	X	
Reduction of Logistic Costs			X	X	X	
Portability and Mobility						
Device Mobility	X		X	X		X
User Mobility	X		X	X		X
Diversity of User and Device Identity	X		X	X	X	X

			vSIM Architecture			
Architecture Criteria	SIM	SIM / MTM	Model "One"	Model "Two"	Model "Three"	C/S SIM
Flexibility and Scalability						
Multiple SIMs in a single Device			X	X	X	X
Remote Service Update	-/X	-/X	X	X	X	
Dynamic Service Up- and Downgrade				X	X	
Remote Take-Ownership vSIM Container			X	X	X	
Network-based Migration			X	X	X	X
Usability						
Conventional Usage Characteristics	X	X	X	X	X	
Online-Registration of a Subscriber	X	X	X	X	X	
Online-Roll-Out of SIM Credentials			X	X	X	
Compatibility						
Compatibility to GSM 01.02	X	X	X			X
Compatibility to GSM 11.11	X	X	X	X		X
Compatibility to GSM 11.14	X	X	X	X		X
Acceptance						
Device Owner / User	X	X	X	X		
Mobile Network Operator (MNO)	X	X	X	-/X		

Table 4.2: *Comparison vSIM/SIM Architectures*

Chapter 5

Conclusions and further work

In this thesis, we have examined both, theoretical and practical aspects of how subscriber authentication in mobile cellular networks could be implemented to the next generation of mobile phones and devices. This chapter summarizes the results and discusses possible further work.

5.1 Conclusion

The present thesis demonstrates the substitutability of a SIM card with an adequate trusted software module, supported and protected by a trustworthy operating system. In this regard, we have introduced vSIM Credentials as a means for subscriber authentication based on the TCG MPWG technology. It offers a real alternative to the other SIM-based solutions under consideration, while an sufficient degree of security and usage characteristics are reached.

The first contribution of this thesis has examinated several architectural directions, including our aspired architecture of subscriber authentication with vSIMs. Furthermore, we have discussed a set of benchmarks, which have been seen crucial in terms of our objective.

The required theoretical fundament and background of GSM and Trusted Computing was given in Chapter 2. In particular, we have discussed the forthcoming *TCG MPWG Reference Architecture* and have detailed theoretical and practical aspects of this specification.

In Chapter 3, we have identified and developed a comprehensive framework for subscriber authentication in Chapter 3. This framework includes the fundamental vSIM architecture including essential methods for deployment and management as well as the conceptual models for subscriber authentication in mobile cellular networks. Therefor, we have systematically discussed the procedures and mechanisms of efficient administration, management and maintenance of subscriber credentials. The following protocols were detailed:

- Remote-Take-Ownership of a vSIM Container
- Subscriber Enrollment and vSIM Credential Roll-Off, and
- Migration of a vSIM Container and vSIM Credential

Based on this fundament, we have developed three intergraded models for subscriber authentication using trusted computing.

- Model ”One”: Subscriber Access in mobile cellular Networks based on Trusted Computing with compatibility to GSM - Authentication
- Model ”Two”: GSM-Subscriber Authentication in mobile cellular Networks based on Trusted Computing with Remote Attestation for Restricted-Network-Access
- Model ”Three”: Generalized Subscriber Authentication in IT Networks Infrastructures based Trusted Computing with Remote Attestation for Restricted-Network-Access

This vSIM architecture was analysed and evaluated in Chapter 4. As a result, it shows that a traditional SIM-Card could be replaced by a virtual SIM which is base on this framework. In particular, Table 4.2 reveals that the proposed *Single Trust-Anchor Architecture using virtual SIMs for Subscriber Authentication* in conjunction with “Model One” based on a generic MTM emulation is a suitable and sustainable approach with regard to the SIM-based solutions. A prototypical implementation on a trustworthy operating platform is under development.

5.2 Outlook and further Research

From a general point of view it seems like Trusted Computing will play a significant role in future computing. Using a vSIM as a trusted and protected software allows expansion to a much wider field of authentication and identification management systems on standard PC platforms [DEPY03]. The realization of (mobile) trust credentials in user-centric scenarios by vSIM credentials, as shown in Figure 5.1, or the support of online transactions by vSIM authentication are thinkable approaches.

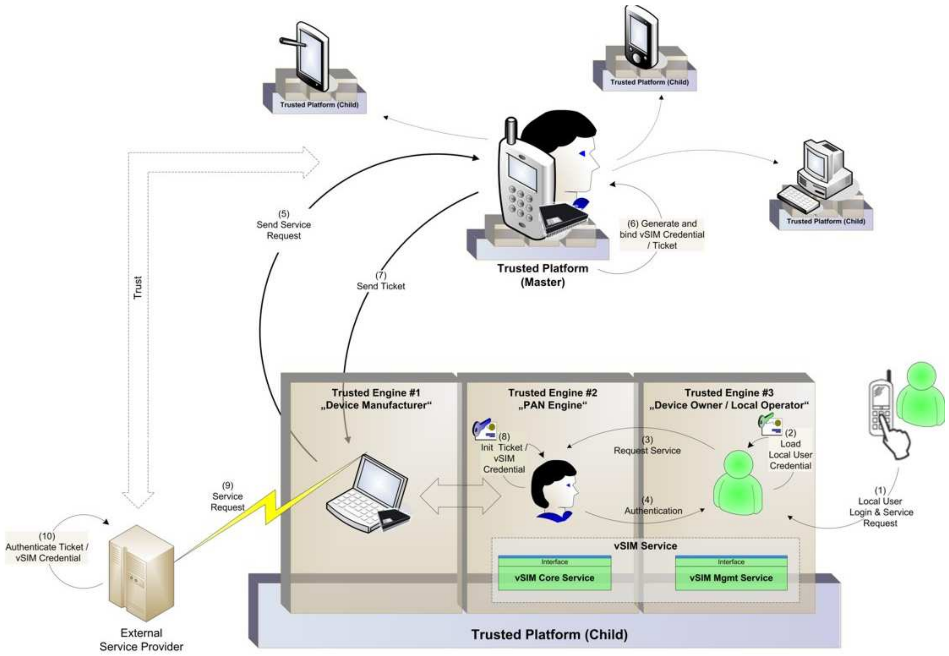


Figure 5.1: *Personal-Area-Networks and vSIMs*

For this reason, we plan to integrate the vSIM model into the generic domain. However, there are some privacy and security challenges associated with this implementation on a desktop computer using an unmodified TPM, which needs a further research.

Bibliography

[BBH^{+}98] R. Baumgartl, M. Borriss, Cl.-J. Hamann, M. Hohmuth, L. Reuther, S. Schönberg, and J. Wolter. *Dresden Realtime Operating System (DROPS).* In *Workshop of System-Designed Automation*, 1998. (SDA'98).

[BCG^{+}06] S. Berger, R. Caceres, K. A. Goldman, R. Perez, R. Sailer, and L. van Doorn. vTPM: Virtualizing the Trusted Platform Module. Technical report, IBM T. J. Watson Research Center, Yorktown Heights, 2006.

[DEPY03] J. Dashevsky, E. C. Epp, J. Puthenkulam, and M. Yelamanchi. SIM Trust Parameters. *Intel Developer Update Magazine*, 2003.

[DFH^{+}03] B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, I. Pratt, A. Warfield, P. Barham, and R. Neugebauer. *Xen and the Art of Virtualization.* In *Proceedings of the ACM Symposium on Operating Systems Principles*, 2003.

[Eck04] C. Eckert. *IT-Sicherheit - Konzepte, Verfahren und Protokolle*, volume 2, chapter 13.1 (GSM), pages 641–643. Oldenbourg Verlag, Munich, 2004.

[Eck06] C. Eckert. *IT-Sicherheit - Konzepte, Verfahren und Protokolle*, volume 4, chapter 11.10 (Trusted Computing), pages 617–649. Oldenbourg Verlag, Munich, 2006.

[EMS] EMSCB - Towards Trustworthy Systems with Open Standards and Trusted Computing. Offical Website. http://www.emscb.com.

[Hew07] Hewlett-Packard Development Company, L.P. Business PC Security Solutions - Questions and Answers. http://h20331.www2.hp.com/Hpsub/cache/292232-0-0-225-121.html, 2007.

[Hoh96] M. Hohmuth. *Linux-Emulation auf einem Mikrokern.* PhD thesis, Technical University Dresden, Dresden, 1996.

[iGR06] iGR. *Worldwide Wireless and Mobile Market Forecast, 2005-2010.* Technical report, iGillott Research Inc., 2006.

[Imp] Implementa SIM Server. Offical Website. http://www.implementa.com.

[JK03] J. Jonsson and B. Kaliski. *Public-Key Cryptography Standards PKCS-1.* Technical Report 2.1, RSA Laboratories, Bedford, 2003.

[KBC97] H. Krawczyk, M. Bellare, and R. Canetti. *HMAC: Keyed-Hashing for Message Authentication.* Technical Report RFC-2104, Internet Engineering Task Force (IETF), 1997.

[KS06] N. Kuntze and A. U. Schmidt. *Trusted Computing in Mobile Action.* In *Peer reviewed Proceedings of the ISSA 2006 From Insight to Foresight Conference.* Information Security South Africa (ISSA), 2006.

[Mit05] C. Mitchell. *Trusted Computing.* The IEE, 2005. IEE Professional Applications of Computing Series 6.

[Nat02] National Institute of Standards and Technology. *FIPS PUB 180-2; Specifications for the Secure Hash Standard.* Technical report, NIST, 2002. Federal Information Processing Standards (FIPS).

[NYHR05] C. Neuman, T. Yu, S. Hartman, and K. Raeburn. *The Kerberos Network Authentication Service (V5).* Technical Report RFC-4120, Internet Engineering Task Force (IETF), 2005.

[PER] Perseus Security Framework. Offical Website. http://www.perseus-os.org.

[rGPP91] 3rd Generation Partnership Project. *3GPP TS 03.20; Security-related Network Functions.* Technical Report 3.3.2, 3GPP, 1991. Technical Specification Group Services and System Aspects.

[rGPP97a] 3rd Generation Partnership Project. *3GPP TS 02.09; Security Aspects.* Technical Report 6.1.0, 3GPP, 1997. Technical Specification Group Services and System Aspects.

[rGPP97b] 3rd Generation Partnership Project. *3GPP TS 11.11; Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface.* Technical Report 3.3.2, 3GPP, 1997. Technical Specification Group Services and System Aspects.

[rGPP01] 3rd Generation Partnership Project. *3GPP TS 01.02; General description of a GSM Public Land Mobile Network (PLMN).* Technical Report 6.1.0, 3GPP, 2001. Technical Specification Group Services and System Aspects.

[rGPP02] 3rd Generation Partnership Project. *3GPP TS 55.205; Specification of the GSM-MILLENAGE Algorithms; An example algorithm set for the GSAuthentication and Key Generation functions A3 and A8.* Technical Report 6.0.0, 3GPP, 2002. Technical Specification Group Services and System Aspects.

[rGPP04] 3rd Generation Partnership Project. *3GPP TS 11.14; Specification of the SIM Application Toolkit (SAT) for the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface.* Technical Report 8.17.0, 3GPP, 2004. Technical Specification Group Services and System Aspects.

[rGPP06] 3rd Generation Partnership Project. *3GPP TS 55.216; Specification of the A5/3 Encryption Algorithm for GSM, and the GEA Encryption Algorithm for GPRS.* Technical Report 6.1.0, 3GPP, 2006. Technical Specification Group Services and System Aspects.

[rGPP07] 3rd Generation Partnership Project. *3GPP TS 21.101: Technical Specifications and Technical Reports for a UTRAN-based 3GPP system.* Technical Report 6.6.0, 3GPP, 2007. Technical Specification Group Services and System Aspects.

[Sma01] Smart Card Securtiy User Group. *SCSUG-SCPP; Smart Card Protection Profile.* Technical report, Common Criteria for Information Technology Security Evaluation, 2001.

[Str05] M. Strasser. *Software-based TPM Emulator for Linux.* http://tpm-emulator.berlios.de/, 2005.

[SVJ$^+$05] R. Sailer, E. Valdez, T. Jaeger, R. Perez, L. van Doorn, J. L. Griffin, and S. Berger. *sHype: Secure Hypervisor Approach to Trusted Virtualized Systems.* Technical Report RC23511, IBM Research Division, 2005.

[TCG] TCG. Offical Website. https://www.trustedcomputinggroup.org.

[Tru02a] Trusted Computing Group. *Trusted Computing Platform Alliance (TCPA) Main Specification.* Technical Report 1.1b, Trusted Computing Group, 2002.

[Tru02b] Trusted Computing Group. *Trusted Platform Module Protection Profile.* Technical Report 1.9.7, Trusted Computing Platform Alliance (TCPA), 2002.

[Tru05] Trusted Computing Group. *TPM Main Part 1 Design Principles.* Technical Report Version 1.2 Revision 94, TCG, 2005.

[Tru06a] Trusted Computing Group. *TCG MPWG Mobile Reference Architecture.* Version 1.0 Draft 28, currently unpublished (30.04.2007), 2006.

[Tru06b] Trusted Computing Group. *TCG MPWG Mobile Trusted Module Specification.* Technical Report Version 0.9 Revision 1, TCG, 2006.

[Tru06c] Trusted Computing Group. *TCG Software Stack (TSS).* Technical Report Version 1.2, Level 1, Errata A, TCG, 2006.

[Tru07] Trusted Computing Group. *TCG Specification Architecture Overview.* Technical Report Revision 1.3, TCG, 2007.

[Wal00a] B. Walke. *Mobilfunknetze und ihre Protokolle. Grundlagen, GMS, UMTS und andere zellulare Mobilfunknetze*, volume 2, chapter 5, pages 367–458. B.G. Teubner, Stuttgart, 2000.

[Wal00b] B. Walke. *Mobilfunknetze und ihre Protokolle. Grundlagen, GMS, UMTS und andere zellulare Mobilfunknetze*, volume 2, chapter 3, pages 135–342. B.G. Teubner, Stuttgart, 2000.

[Wir06] Wireless Inteligence. *GSM subscriber statistics - GSMA Q2-2006.* Technical report, GSMA, 2006.

Erklärung

Hiermit versichere ich, die vorliegende Diplomarbeit selbstständig und unter ausschliesslicher Verwendung der angegebenen Quellen und Hilfsmittel angefertigt zu haben. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mannheim, 30.04.2007 Michael Kasper