Subscriber Authentication in mobile cellular Networks with virtual software SIM Credentials using Trusted Computing

Michael Kasper, Nicolai Kuntze, Andreas U. Schmidt
Fraunhofer-Institute for Secure Information Technology SIT
Rheinstrasse 75, 64295 Darmstadt, Germany
Email: {michael.kasper,nicolai.kuntze,andreas.u.schmidt}@sit.fraunhofer.de

***Abstract*—The primary goal of this paper is to design a software replacement for a Subscriber Identity Module (SIM) based on the *TCG MPWG Reference Architecture* in order to access a mobile cellular network and its offered services. Therefor, we introduce a *virtual software SIM (vSIM)* with comparable usage and security characteristics like the traditional smartcard-based solution. Additionally, 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. 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.**

I. Introduction

In its forthcoming *TCG Mobile Reference Architecture*, the Mobile Phone Work Group of the Trusted Computing Group (TCG MPWG) specifies a new concept to enable trust into future mobile devices. It offers new potentials for implementing trust in mobile computing platforms by introducing multiple trusted engines on behalf of different stakeholders supported by a hardware-based trust anchor [10], [11].

In the context of this paper, we consider this specification from a slightly different point of view as it is envisaged by the TCG. Even though, SIM-based authentication is the established and proposed means for user authentication and securely accessing mobile cellular networks, an alternative is coming up with the advent of the TCG technology in mobile devices. Due to the capabilities of a mobile trusted platform to support multiple trusted engines with protected storage, strong isolation and secure communication within a defined security perimeter, the trusted platform is able to take over the SIM functionality.

This paper is organized as follows. In Section II, we give an overview of the vSIM architecture. The following Section III is the core of this paper. Here we present conceptual models for subscriber enrollment and authentication in mobile cellular networks using trusted computing. In Section IV we introduce to a prototypical implementation of the specified vSIM services. In Section V we summarize and conclude on our work and point out further research.

II. 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 familiarize the reader with significant components and services of a vSIM platform. For a better understanding, we recommend to [10] and [7]. In particular, the paper [7] of the authors introduce to essential parts of the *TCG MPWG Reference Architecture* and give an overview of significant platform components in terms of our objective. Figure 1 schematically shows the layout of such a trusted platform. It holds an (abstract) virtual software SIM service which substitutes the traditional smartcard and its functionality.

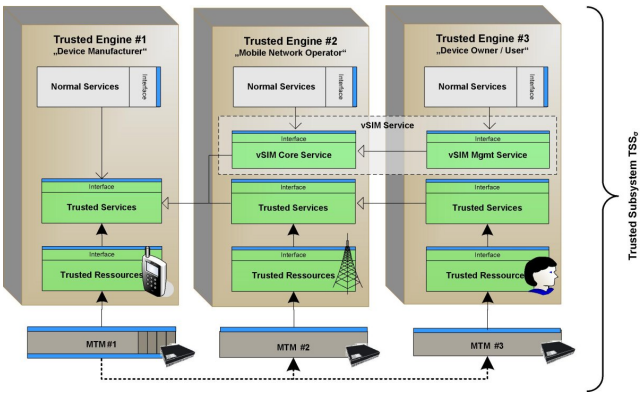

Fig. 1. TCG MPWG Architecture (reduced)

In general, a mobile trusted platform supports a set of trusted engines. Each engine 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 (DO).

For each environment, we define a trusted subsystem $\mathcal{TSS}_\sigma$ as a logical unit of a trusted engine together with its interrelated hardware compartment of a stakeholder σ. It is used for security-critical functionality and consists of a Mobile Trusted Module (MTM_σ) with its associated trusted engine TE_σ. All sensitive data required by the trusted subsystems is protected by their dedicated MTM_σ, either directly or indirectly.

Device Manufacturer Subsystem: The $\mathcal{TSS}_{DM}$ 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 $\mathcal{TSS}_{DM}$ to its destination.

Mobile Network Operator Subsystem: All cellular services of a platform are assigned to $\mathcal{TSS}_{MNO}$. 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.

Device Owner Subsystem: In context of the vSIM service, the $\mathcal{TSS}_{DO}$ protect all personal information and corresponding user credentials ($Cred_U$). Moreover, it holds a *vSIM User 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, to provide evidence of a local user's identity.

III. Subscriber Authentication with vSIMs based on Trusted Computing

In this section, we will give an informal description of the scenario and identify the essential components of an idealized protocol. Based on this scenario, we design two intergraded conceptual models for subscriber authentication in mobile cellular networks using trusted computing. With these models we show how a traditional SIM-Card could be replaced by a software emulation, which runs within isolated environments, protected and supported by trusted engines. We call this software emulation a "vSIM — trusted virtual Subscriber Identity Module". Furthermore, we discuss how user enrollment and key delivery mechanisms are carried out efficiently.

A. Notation

To be prepared for the protocol descriptions we introduce the reader to some basic notations, which are used within the following protocol specification. The term $SIGN(x, \mathcal{K})$ denotes a digital signature message of data x computed with a private signature key $\mathcal{K}$. A blob message is denoted by $BIND(X, IM_{TP}, AIK_i)$ and is an encrypted message X bound to IM_{TP} and $AIK_{(X}$. $H(M)$ denotes a result of a one-way hash function on data M . Arbitrary data x is cryptographically bound to a certain platform configuration IM_{TP} by the function $SEAL(X, IM_{TP}, AIK_i)$. $X \| Y$ is a concatenation of data X and Y.

B. Scenario

The use-case under consideration is illustrated in Figure 2 and involves four significant entities: the local user (U), the trusted mobile platform (TP), 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

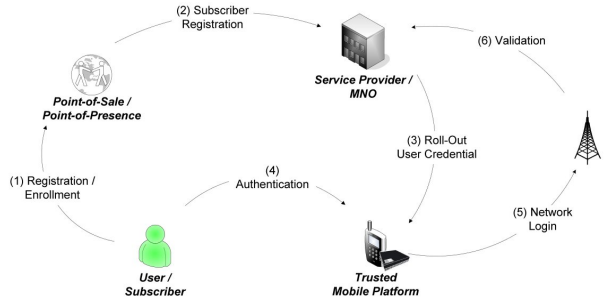

Fig. 2. Generic Trusted Mobile Scenario

with the MNO (Step 1), in order to use the mobile network infrastructure and its 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 $\mathcal{TSS}_{MNO}$ 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).

C. Security Requirements

It is important, that our proposed vSIM services 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*, *User Authentication* and *Secure Channel*. In particular, a vSIM platform has to guarantee that only authorized subjects can read or alter protected vSIM data, while

1) in transit to a vSIM services, or other trusted services
2) in storage on the mobile trusted platform
3) it is executed within the trusted environment
4) it is transferred between trusted services 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.

D. Platform and Protocol Precondition

In a preliminary state of all protocols, the platform has carried out a (remote-)takeownership procedure for all subsystems, and has installed an EK (or alternatively pre-generated AIK's) and a SRK within its isolated key-hierarchy. Furthermore, the platform has performed 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.

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

1) Protocol Scheme and informal Description: The Point-of-Sale orders a set of (pre-generated) registration tickets from the MNO. A registration ticket consists of a $IMSI_i$, $NONCE_{MNO}$, $NONCE_U$, and $AUTH_i$. The $IMSI_i$ identifies an *International Mobile Subscriber Identity*. In other scenarios, it may be a unique credential ID that is assigned by the network operator. The terms $NONCE_{MNO}$ and $NONCE_U$ denote random values, which are needed to challenge $\mathcal{TSS}_{MNO}$ and $\mathcal{TSS}_{DO}$ 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 describes the first phase. Here, we discuss the user enrollment and

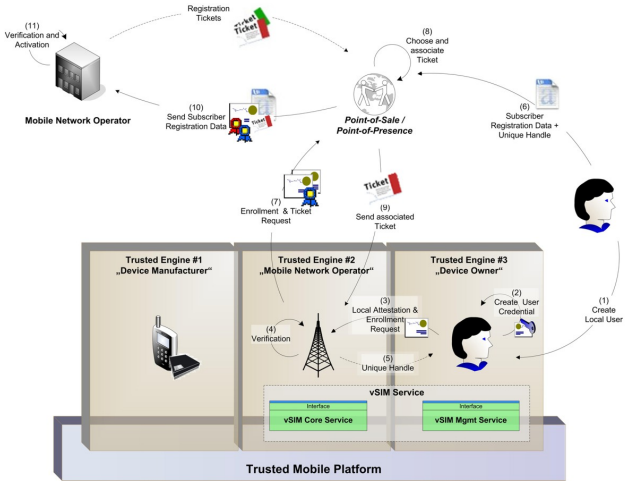

Fig. 3. Model "Subscriber Registration and Enrollment"

registration for services, offered by the MNO. The Device Owner/User starts to request a new user credential for a local user, which is generated by $\mathcal{TSS}_U$. For this, the local user enters a unique personal identifier ID_U and an authorization password PWD_U to the trusted service $vSIM_{MGMT}$. Afterward, $vSIM_{MGMT}$ generates a asymmetric signature key-pair $\mathcal{K}_U$ and creates a certificate, which includes all relevant information, like the $RegData_U$ and the public portion of $\mathcal{K}_U$. The $vSIM_{MGMT}$ pass this certificate $Cert_U$ to the $vSIM_{CORE}$ service.

Within this step, $vSIM_{MGMT}$ requests a enrollment procedure and reports its current state and configuration to the local verifier of $vSIM_{CORE}$. The $\mathcal{TSS}_{MNO}$ validates the given data (against Reference Integrity Metrics (RIM)) and checks, whether the present engine's state is in a acceptable condition. 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 user (resp. $vSIM_{MGMT}$).

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. $vSIM_{CORE}$ 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.

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

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 $\mathcal{K}_{POS}$ and sends the signed data (online or offline) to the MNO. Optionally, the POS encrypt the data with the public portion of $\mathcal{K}_{MNO}$.

The MNO verifies the data and generates the $Cred_{vSIM}$ with the $IMSI_i$, the shared key $\mathcal{K}_i$ and the certificate $Cert_U$ and signs this bundle with the private signature key $\mathcal{K}_{MNO}$. Finally, the MNO activates the signed $Cred_{vSIM}$ and the corresponding nonces in its authentication center. 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 order to obtain a $Cred_{vSIM}$, the user performs a log-in sequence and sends a unique id ID_U with a proper password PWD_U to the $vSIM_{MGMT}$ service, which loads the associated user key-pair K_U from protected storage.

In a next step, the $vSIM_{MGMT}$ initializes a *vSIM Roll-Out Procedure* and sends a request the the $vSIM_{CORE}$ service. After having received this message, it unbinds the corresponding $Ticket_i$ and verifies the authenticity and integrity of the $Ticket_i$. Next, $vSIM_{CORE}$ extracts the $NONCE_U$ from the $Ticket_i$ and challenge U with this value. $vSIM_{MGMT}$ 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 $vSIM_{CORE}$.

After the $vSIM_{CORE}$ has received the message, it composes a vSIM credential request and submits it to the as-

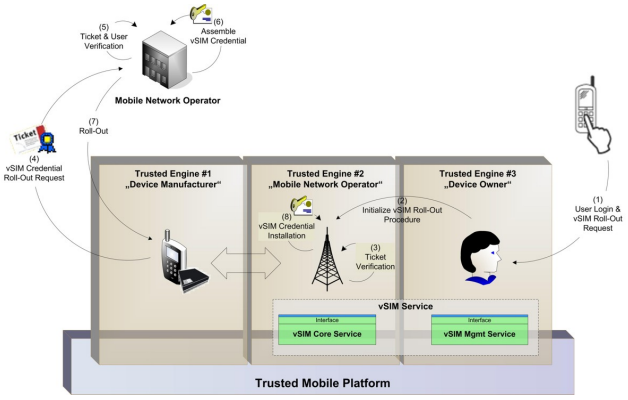

Fig. 4. Model "vSIM Credential Roll-Out"

signed MNO registration service via some channel, mentioned above. Therefor, $vSIM_{CORE}$ extracts $NONCE_{MNO}$ from the $Ticket_i$ and signs it together with the $IMSI_i$. Afterward, the $vSIM_{CORE}$ sends the its own signature and the obtained user signature to MNO.

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). 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 $\mathcal{K}_S$. Afterward, the key $\mathcal{K}_S$ is bound with the corresponding key of $\mathcal{TSS}_{MNO}$ and optionally to acceptable integrity metrics. The MNO encrypts $Cred_{vSIM}$ with this session key and sends both to the $\mathcal{TSS}_{MNO}$.

Finally, $\mathcal{TSS}_{MNO}$ unbinds $\mathcal{K}_S$. With this key it decrypts the vSIM credential and checks the enclosed signature. 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 $\mathcal{K}_S$ 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 $\mathcal{K}_S$ 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 $\mathcal{K}_S$.

F. Model "One" - Subscriber Authentication with compatibility to GSM - Authentication

Our proposal for model "One" is straightforward to actual GSM standard. It is implementable in conventional GSM clients without any technological changes at the GSM infrastructure and at the GSM authentication protocol. Here, the main task of the vSIM service is to take over the functional range of the SIM card, with no additional duties and responsibilities regarding to the GSM 11.11 SIM specification [5]. The cryptographic algorithms A3 and A5, responsible for user authentication and key generation are implemented within the $vSIM_{CORE}$ service.

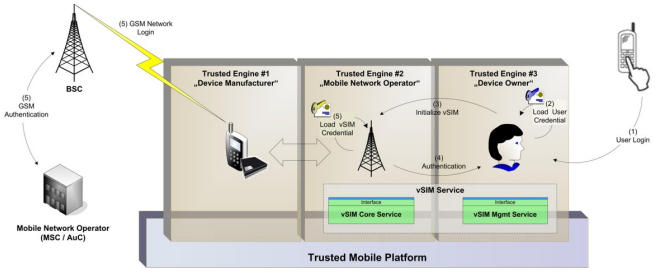

Fig. 5. Model "One"

1) Protocol Scheme and informal Description: 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 $Cred_{vSIM}$.

a) Phase 1: "Initialization of vSIM Credentials": First, the user initializes the vSIM services equal to the precedent protocol. Afterward, the $vSIM_{MGMT}$ service connects to the trusted interface layer of $vSIM_{CORE}$ and sends a vSIM credential initialization request to this service.

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. Now, the $vSIM_{MGMT}$ takes the corresponding private portion of the user signature key, signs the challenge $RAND_{AUTH}$ and sends this value back to the $vSIM_{CORE}$ service.

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 $\mathcal{K}_i$.

b) Phase 2: "Subscriber Authentication": The GSM standard defines its own authentication protocol based on SIM credentials. Since the SIM_{CORE} indirectly talks to the MNO, $\mathcal{TSS}_{DM}$ 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.[1]

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, $\mathcal{TSS}_{DM}$ relays $IMSI_i$ (or $TMSI_i$) from $vSIM_{CORE}$ to MNO. The MNO generates internally a set of triplets containing: a authentication challenge $RAND_{GSM}$, a corresponding session key $\mathcal{K}_{GSM}$ and a $SRES_{GSM}$. The $\mathcal{K}_{GSM}$ and the $SRES_{GSM}$ are cal-

[1]We 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.

culated with the GSM A3 and A8 algorithms, that implement a one-way function: $\mathcal{K}_{GSM} = A8(K_i, RAND_{GSM})$ and $SRES_{GSM} = A3(K_i, RAND_{GSM})$. The MNO sends a authentication challenge $RAND_{AUTH}$ back to TE_{MNO}.

This $RAND_{GSM}$ is passed to the trusted $vSIM_{CORE}$ service. Next, it also uses the A3 algorithm together with the key $\mathcal{K}_i$. The output of the algorithm is the challenge response message $SRES^*_{GSM}$. The $vSIM_{CORE}$ sends this $SRES^*_{GSM}$ message to the MNO.

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

G. 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 [8]. Beside the main task of SIM substitution, it provides (1)

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

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.

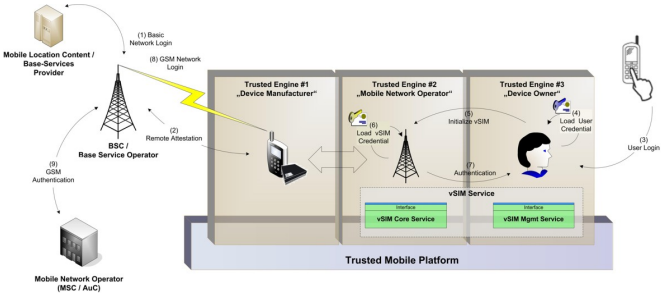

Fig. 6. Model "Two"

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

1) Protocol Scheme and informal Description: This protocol description is separated into three phases.

a) Phase 1: "Remote Attestation": At the beginning, the trusted platform initialize the remote attestation and device authentication process. TP 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}. Therefor, the $\mathcal{TSS}_{DM}$ generates a random value $RAND_{BASE}$ and signs its current integrity metric IM_{TP}. Next, the base authentication service of TSS_{DM} sends $RAND_{BASE}$, the signature (with the corresponding attestation information) and its certificate $Cert_{TSS_{DM}}$ to the network access point.

Having received this request, NAP_{MNO} checks the state of the client machine. If the signed integrity metric of the client platform fails verification, it 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 $\mathcal{K}_{BASE}$ and a network ticket. 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: ID_{TP}, ID_{NAP}, K_{BASE}, $REALM_{BASE}$, $LIFETIME_{BASE}$.

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}, which relays it to the client platform. Therefor, the message is bound to the trusted subsystem $\mathcal{TSS}_{DM}$ with the corresponding public key $\mathcal{K}_{TSS_{DM}}$ and a valid integrity metric. Once, $\mathcal{TSS}_{DM}$ has received the signed message, it verifies the status of the signed $RAND_{BASE}$. If revoked, the subsystem replies with an error message and halts the protocol. Otherwise the AUC_{MNO} is authenticated by the challenge response.

The $\mathcal{TSS}_{DM}$ decrypts the session key K_{BASE} and sends the encrypted ticket together with an authenticator A_{TP} to the NAP_{MNO}. The authenticator A_{TP} is composed of its platform identity ID_{TP}, the current network address $ADDR$, and a timestamp $TIME$.

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.

b) Phase 2:"Initialization of vSIM Credentials": The initialization of a vSIM credential is identical to model "One". For a detailed description of the protocol sequence, we refer to Section III-F1a.

c) Phase 3: "Subscriber Authentication" (Variant 1): Similar to Section III-F1a, this variant performs subscriber access with compatibility to regular GSM authentication. In an additional step, K_{BASE} is substituted by the session key $\mathcal{K}_{GSM}$ on both sides, the NAP_{MNO} and TP.

However, this approach is optimizable, by embedding the $RAND_{GSM}$ already in a preceding step. In this case, $vSIM_{CORE}$ extracts the $RAND_{GSM}$ from this message, cal-

culates the challenge response $SRES := A3(RAND_{GSM})$ and sends both to the MNO. The MNO generates internally the expected $SRES_{GSM}$ and the corresponding session key $\mathcal{K}_{GSM}$.

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 protocol step from phase 1. On the other hand, the user has prooven its identity by $SRES_{GSM}$. The authentication between NAP and U is implicitly proven by a valid communication key $\mathcal{K}_{GSM}$.

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. Therefor, it extracts the timestamp $TIME$ from the authenticator A_U and increments the value and encrypts it with the shared communication key $\mathcal{K}_{GSM}$ (or a derivation of it). Finally, it sends the message back to the trusted platform.

d) 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 (Public Land Mobile Network). In particular protocol flaws in SS7 (Signaling System 7) 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 unprotected over the SS7 network. This triplet contains of a challenge $RAND$, the correct response $SRES$, and the communication key $\mathcal{K}_{GSM}$.

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 $\mathcal{K}_{GSM}$ 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 $\mathcal{K}_{GSM}$ across the PLMN infrastructure.

IV. Prototypical Implementation on EMSCB/Turaya

The prototypical implementation of the trusted engines, and the specified vSIM services are realizable as an extension to the existing Turaya Computing Platform. Turaya is an implementation of the EMSCB security architecture [3] and 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. Figure 7 illustrates our model. A hypervisor/microkernel executes a legacy operating system in coexistence with a running instance of the EMSCB-based security architecture. The latter controls a virtual machine with several trusted engines and services compliant to the TCG requirements

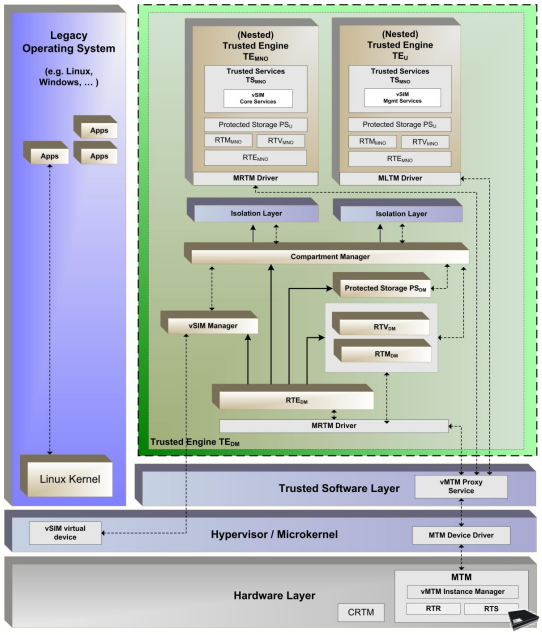

Fig. 7. vSIM Architecture on EMSCB/Turaya

[10], [11]. Each engine holds a set of trusted services on behalf of a specific stakeholder, which are executed within isolated execution environments [9], [1], [6] on top of the security kernel or nested inside TE_{DM}. In particular, TE_U is responsible for vSIM management and user authentication and TE_{MNO} holds the trusted vSIM core services (e.g. the GSM security algorithms A3/A8 [4]). The *vSIM Manager* component within the TSS_{DM} represents an interface to the $vSIM_{CORE}$ and $vSIM_{MGMT}$ services and is responsible for the communication with the underlying architecture and external entities. This component holds, for instance, the set of accepted grammar and commands of the vSIM service.

V. Conclusion and Further Work

In this contribution we have introduced to vSIM credentials as a means for subscriber authentication based on the TCG MPWG technology. It offers a real alternative to SIM-based solutions, if an equal degree of security and usage characteristics are reached. On a trustworthy operating platform, like the EMSCB platform, an adequate level may be achieved.

In our proposed solution, the requirements from Section III-C are fulfilled by fundamental security mechanisms of the *TCG MPWG Reference Architecture*. In particular, the need for confidentiality and integrity protection during execution and storage of a vSIM Credential is reached by our model. Only entities authorized by the associated stakeholder are able to access the specific vSIM services. Nevertheless, we note that our model is constraint by the provided security-level of the hardware trust-anchor.

We plan to integrate the implementation of the vSIM model into the generic domain. 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 [2]. The realization of (mobile) trust credentials in user-centric scenarios by vSIM credentials may be one thinkable approach. Therefore, we have already developed a generalization of the proposed subscriber authentication mechanisms. A prototypical implementation on EMSCB/Turaya is under development. 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.

REFERENCES

[1] R. Baumgartl, M. Borriss, C.-J. Hamann, M. Hohmuth, L. Reuther, S. Schönberg, and J. Wolter. Dresden realtime operating system (drops). In *Workshop of System-Designed Automation*, (SDA'98), 1998.

[2] J. Dashevsky, E. C. Epp, J. Puthenkulam, and M. Yelamanchi. Sim trust parameters. *Intel Developer Update Magazine*, 2003.

[3] EMSCB. European multilaterally secure computing base - towards trustworthy systems with open standards and trusted computing. http://www.emscb.com, 2006.

[4] ETSI. Digital cellular telecommunications system (phase 2); security related network functions (gsm 03.20 version 4.4.1). Technical report, European Telecommunications Standards Institute, 1997.

[5] ETSI. Gsm 11.11 - digital cellular telecommunications system (phase 2+); specification of the subscriber identity module - mobile equipment (sim - me) interface. Technical report, European Telecommunications Standards Institute, 1997.

[6] M. Hohmuth. *Linux-Emulation auf einem Mikrokern*. PhD thesis, TU Dresden, Fakultät Informatik, Lehrstuhl Betriebssysteme, 1996.

[7] M. Kasper, N. Kuntze, and A. U. Schmidt. On the deployment of mobile trusted modules, 2007.

[8] 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.

[9] PERSEUS. The perseus security framework. http://www.perseus-os.org. Applied Data Security Group, Ruhr-University Bochum.

[10] TCG. Tcg mpwg mobile reference architecture. Specification Version 1.0, Draft 27, Nov. 2006.

[11] TCG. Tcg mpwg mobile trusted module specification, version 0.9, 2006.