SIM card based Security and Trust Management in Mobile Services

Lars Schnake, Carsten Rust, Rebekka Neumann

Abstract—**The paper describes an architecture for mobile services where the SIM card is integrated as a personal security token for providing basic services related to security, privacy, and trust. The presented work is part of a cooperative research initiative aiming at an open architecture for mobile services. The SIM is integrated using standard internet protocols. A web server on the card enables the exchange of data with the mobile device through HTTP. Moreover, a servlet architecture on the card allows for the provisioning of smart card based services with an interface similar to that of Web services. An important issue within the open and heterogeneous infrastructures for future mobile services is support for identification, evaluation, and rating of service offers. We therefore propose a SIM based trust management. The service is designed following the ideas of a web-of-trust infrastructure with an on-card key ring and trust value management. It uses digital signing for identification of services as well as for signatures by the user.**

Index Terms: **Mobile services, SIM, Trust management, Smart Card, Java Card**

I. Introduction

In order to repeat the success of the Web, mobile services of the future have to be simple to find, simple to use, simple to trust, and simple to set-up. The cooperative project “Simple Mobile Services” [1] has the strategic objective to define a new class of services w.r.t. these requirements, meeting the specific needs of mobile users. In this paper, we mainly address the specification and implementation of mobile services that are “simple to trust”.

Typical use cases for mobile services are travel scenarios. Consider for instance a traveller arriving at an airport. Assuming that he has an electronic note (e.g. a MEM like described in [2]) with all his travel details on the mobile phone, he could be offered a couple of services matching his personal situation: check-in for flight, guidance to the gate or other places of interest, alerts for important events like boarding, car rental for the flight destination, meetings with buddies, etc. In the described scenario, the requirement for secure and trustworthy services is obvious: for check-in, the flight ticket must be checked thoroughly. Alerts must be reliable as well as payment procedures. Finally, as service discovery and negotiation require access to personal user data, proper mechanisms for protecting the privacy of the user must be in place.

Security issues are important from both the end-user as well as from the service provider perspective. One requirement for repeating the success of the Web with mobile services is to enable individuals and small companies to become service providers (“simple to set-up”). This will cause a flood of heterogeneous service offerings to the user and complicates a trustable usage.

Nowadays, the SIM establishes the security of mobile networks by authenticating subscribers. Thereby, it provides an identity which is also used for further purposes as for instance micro-payment which is carried out through the mobile network operator.

As a security device for mobile networks, the SIM is also technologically capable of providing security, privacy, and trust functionality for mobile services. It includes secure tamper-proof storage capabilities as well as the required cryptographic modules. However, there is currently no established standard for application developers to access SIM functionalities as a service. Hence, a developer of a mobile service has no easy-to-use means to integrate SIM functionalities in order to secure the service.

In this paper, we present our approach for integrating the SIM as an enabler for security, privacy, and trust in mobile services. The integration is based on standard protocols supported by a web server and a servlet architecture on the card. Hence, the SIM can be used to provide basic security functionalities, which can be utilized by application programmers when realizing mobile services. We’ve realized the described approach with current smart card products. Nonetheless, the evolution to the Next Generation Java Card, called Java Card 3.0^{TM} (JC 3.0), will bring a major improvement. Both the current realization as well as the potential benefit from JC 3.0 are described in the paper. Moreover, an application example, a service for trust management, is considered.

II. State-of-the-art Mobile Service Security

A common environment for applications on mobile devices is the Mobile Information Device Profile (MIDP) for the Java 2 Micro Edition (J2ME). In the following subsection, we describe the basic security concepts of MIDP 2.0. In the second subsection, SIM-based security functions are described. Finally, the common concepts of secure internet connectivity are discussed.

A. .MIDP2.0 Security Domains Concept

MIDP 2.0 defines a set of APIs used by J2ME applications (MIDlets) on a mobile phone. From the security point of view, these APIs are disposed in function groups. The access to these groups follows generic rules. Explicit permissions must be granted especially to APIs which are security-sensitive, create costs for the user, or provide access to the user’s private data.

MIDP2.0 defines four different protection domains (Manufacturer Domain, Operator Domain, Trusted Third Party Domain, and Untrusted Domain), each granting a different level of access to the various function groups. The level of access depends on the source (origin) of a J2ME application and the level of trust this source can offer. Trust into J2ME applications is based on checking certificates which are stored either on the mobile device or on the SIM card, dependent on the protection domain of the application.

In addition to just granting or declining access to APIs or function groups respectively, access can also be granted with a distinct user permission interaction mode. Available modes are Oneshot, Session, and Blanket. For the definition of these modes and the default settings, we refer to [3]. The default settings provided by the mobile device can be altered by the user to a certain extent.

The "Recommended Security Policy for GSM/UMTS Compliant Devices", as described in chapter 15 of the MIDP2.0 specification defines, how operator trusted Protection Domain Root Certificates are placed in a WIM, SIM or USIM card.

Signed (i.e. trusted) MIDlet suites utilize a certificate profile based on the WAP Certificate Profile, WAP-211-WAPCert-20010522-a, which is based on RFC2459 Internet X.509 Public Key Infrastructure Certificate and CRL Profile. Version 3 of the X.509 standard is applied. Further details are described in chapter 4 "Trusted MIDlet Suites using X.509 PKI" and chapter 13 "javax.microedition.pki" of the MIDP2.0 specification.

Aiming at a secure platform for mobile services, the security concepts of MIDP 2.0 should be the basis. However, following MIDP 2.0, the SIM is only used as a secure storage device, which requires general trust into the platform. We prefer to integrate the trust management into the SIM instead, so that the decision about domain classifications is taken by components residing on a tamper-proof device.

B. SIM-based Security in Mobile Services

As outlined in the introduction, SIM cards are well-established as a tamper-proof security device in mobile telecommunication scenarios. Use cases which can be handled by the SIM are, for example, secure data storage, mutual authentication between the card and the external world, signature creation and verification, data encryption, user verification, access control, or secure payment.

"SIM" is the abbreviation for "Subscriber Identity Module". Its primary purpose is to identify a mobile phone user to the operator's network in a secure and consistent way. The three basic standards for smart cards (including SIM cards) are ISO/IEC 7816-4, ISO/IEC 7816-8, and ISO/IEC 7816-9. The main SIM standards derived from those are ETSI TS 102.221 and ETSI TS 102.222. There are several additional standards for SIM cards that support value-added applications (e.g. Short Message Information Service) and executable applications. In general, SIM-based applications ensure authenticity, integrity, confidentiality, and privacy. The majority of today’s SIM cards is Java enabled, i.e. conforms to the JC 2.2 specification from Sun Microsystems [7].

Aiming at using the SIM for basic security mechanisms in a mobile service architecture, one challenge is to interface the SIM by the mobile phone. One candidate technology to this end is the standard "Security and Trust Services API" (SATSA) specified as Java Specification Request 177 (JSR 177) [4]. It offers four optional Java packages to provide security and trust services to J2ME applications by integrating a so called "Security Element" (SE), which can be a smart card. It offers a communication mechanism between J2ME applications (MIDlets) on the mobile phone and JC applications (Applets) on the (U)SIM card. The access to the SIM card itself is protected with a certificate based access control policy.

Regarding interfacing and integration of the SIM, several approaches for realizing the functionality of a web server on smart cards have been proposed in recent years. Like a standard web server, an on-card web server is able to accept HTTP requests and return HTTP responses. The content replied by a web server may be either static or dynamic.

Future smart cards will even implement a full IP stack in order to provide web server functionality. For current smart cards, an on-card web server can be realized via a gateway running on the mobile host device for the smart card. The gateway translates the TCP/IP protocol into a protocol for communication between the host device and the smart card. The interfacing of the SIM based on a Smart Card Web Server will be described in more detail in Sections III and IV respectively.

C. Secure Protocols

SSL (Secure Sockets Layer) [5] is a standard for secure communication of applications using TCP/IP. TLS (Transport Layer Security) is the successor of SSL, standardised by the Internet Engineering Task Force (IETF). SSL/TLS supports privacy, authentication, integrity, and compression of the communication link. The primary function of SSL/TLS is to protect the transmission against eavesdroppers by encrypting all traffic with a common secret session key that is generated during initialization. In general, SSL/TLS is a proven technique for ensuring secure connections on transport layer.

SSL/TLS supplies privacy protection by providing encryption to prevent eavesdropping. Also, server authentication can be used to avoid unintended disclosure of sensitive information to unknown third parties. Thus, SSL/TLS provides mechanisms for privacy. However, current implementations do not allow the user to assign trust values to certificates and servers, which would be necessary to evaluate if a certain piece of information with a given privacy level could be exposed to a given server.

A problem of TLS/SSL within our architecture might be that it is usually based on certificates and requires a global trusted certification authority. This would contradict with our objective of an open infrastructure, where even individuals can act as service providers. Furthermore, the signature of a trust center might not be significant enough regarding the credibility of the service provider, because this requires also an individual assessment. Other actual proposals (e.g. PERSPECTIVES [6]) also aim to overcome these problems but rarely attend the special circumstance of mobile user scenarios.

III. Leveraging Next Generation Java Card

JC 3.0, also called Next Generation Java Card, is the upcoming Java Card standard currently specified by Sun Microsystems (http://java.sun.com/javacard/3.0) and the Java Card Forum (http://www.javacardforum.org/). The main features and benefits of the platform are available. Besides the multithreading capability and the increased programming comfort of JC 3.0 compared to JC 2, one of the highlights will be the internet connectivity of smart cards. A web server and a servlet container will be available on JC 3.0 smart cards in order to serve multiple client requests received over HTTP via TCP/IP at a time by delivering static or dynamic content. Mobile services can be developed by service providers and deployed within servlet containers of multiple vendors in the form of standardized web applications.

A. Web applications as a means for providing mobile services

A web application is a collection of servlets, static content like HTML pages, and other classes and resources constituting a complete application or service running in a servlet container in order to serve client requests or other applications / services. A servlet is a Java class interacting with web clients via a request / response paradigm and generating dynamic content. A servlet obeys a specified life cycle and is managed by a servlet container.

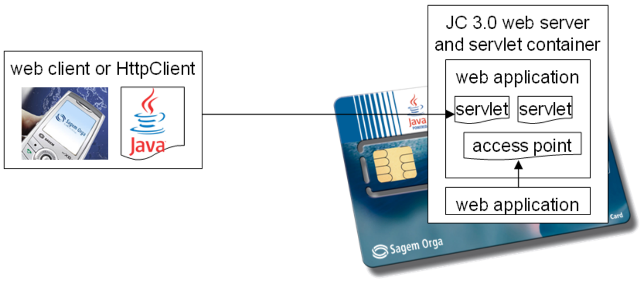


Fig. 1: Access to web applications from web clients and other applications

The JC 3.0 servlet specification will define the interface, the configuration, and the deployment format of web applications on the card (in the following for short web applications). Due to the standardized interface, configuration, and deployment format of web applications, service providers can develop their services independently and deploy their services within every servlet container satisfying the new architecture for mobile services.
The interface of a web application with respect to web clients requesting a static or a dynamic resource is specified by the service method of the web application's servlets and the URI mappings for these servlets used to make the servlets accessible by appropriately built client requests. The interface of a web application with respect to other web applications residing on the same smart card is specified by explicitly listed objects accessible from the other applications.
The configuration of a web application is specified declaratively by the web application's deployment descriptor, the JC platform specific application descriptor, and the runtime descriptor. The deployment descriptor of a web application:

- lists the elements of the web application,
- configures the URI mapping used to access the servlets of the web application,
- configures the session management available within the servlet container in order to track clients' information during several requests,
- configures security constraints of web application components with regard to the authentication used for web clients and with regard to the data transport,
- defines the security roles of users as used for authentication.

The optional JC platform specific application descriptor and the runtime descriptor can be used to specify the access control for other web applications e.g. the access points, among other things. Thus, besides configuring a web application's components, the three descriptors listed above can also be used to configure the web application's authentication mechanism and access control. The security concepts employed within web applications are provided by every servlet container satisfying the new architecture for mobile services.

B. Java Card 3.0 web server and servlet container

The JC 3.0 web server and servlet container (web server for short) provides the network services over which client requests and server responses are sent, and contains and manages servlets throughout their life cycle. The JC 3.0 web server allows concurrent handling of multiple client requests at a time due to the multi-threading capabilities of JC 3.0 virtual machines. Thus, mobile services can – in contrast to JC 2.2 - be accessed in parallel, e.g. a device user and a service provider can access a web application at the same time.
The web applications deployed within the servlet container are Standard Java like applications with high programming comfort due to the well-known and rich programming concepts of Standard Java e.g. the use of the Java String API, and more liberal use of objects compared with JC 2.2.
The JC 3.0 web server and servlet container implements the concurrent handling (see interface ThreadPool in the following figure) of multiple client requests (see classes HttpConnection, Request, and Response in the following figure) by dispatching of the requests to handlers for static resources, servlets, web applications, sessions, and security constraints (see all classes implementing the interface Handler). These handlers generate responses which are returned to the clients.

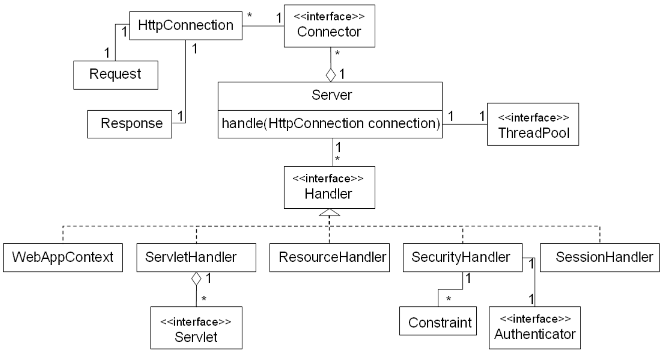


Fig. 2: High level overview of the Java Card 3.0 web server's architecture

The JC 3.0 API provides further security implementations besides the authentication and access control mechanisms mentioned in subsection A and shown in the figure above. These further security implementations e.g. protection domains and code isolation, can be used by service providers when developing mobile services and deploying them in the form of web applications within servlet containers.

IV. SIM Integration

Today, access to the SIM is rigidly regulated according to specifications of large international standardization bodies. For the future however, it can be expected that mobile systems will require fast and flexible deployment for all components, as these systems will evolve and expand steadily, even after first rollout. To accommodate this requirement for the SIM, we choose the new upcoming standards for smart card implemented web services, in order to extend the open architecture for the mobile device to the SIM and to exploit the common HTTP communication as a basis for integration into the mobile phone client (MOVE).

A. Smart Card Web Server

A Smart Card Web Server (SCWS) is an HTTP server that is implemented in a smart card, i.e. in the mobile context in a SIM card. The SCWS can provide static content like HTML pages or media files as well as dynamic content. The content can easily be accessed, for instance from mobile phone internet browsers.

For today's Java Card 2.2 platforms [7], the Open Mobile Alliance (OMA) has specified a SCWS including a proprietary interface for Web applications [9]. It will be adapted for upcoming techniques like high speed interface or TCP enabled smart cards. To deploy advanced web applications on the card with a standardized interface, is also one of the main objectives for JC 3.0 (cf. Section III).

B. Servlet Architecture

The SCWS provides an extension programming interface to deploy dynamic content similar to Java servlets from the J2EE world and other interactive modules that want to make use of HTTP communication.

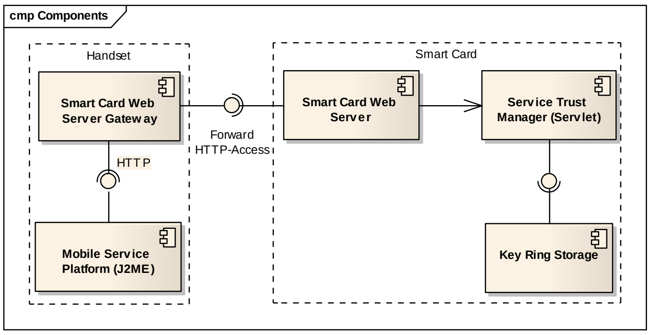


Fig. 3. Smart Card Web Server Architecture

Fig. 3 shows the basic components of the SIM service architecture to integrate the "Service Trust Manager" (cf. Section V) as a servlet of the SCWS on the SIM. Mobile Service Components can easily access the provided functions with the HTTP protocol via the "Smart Card Web Server Gateway" on a connection to the *localhost* address.

C. Service Integration

For service integration, a programming interface is provided by the SCWS and card applications can implement this and register to the SCWS. The SCWS component will then delegate incoming HTTP requests to the addressed application.

To enable communication between mobile device and SIM, the servlet developer can make use of the Request and Response objects. For a flexible two-way communication we mainly make use of the HTTP POST command.

Based on the described mechanisms, it is possible for 3rd party developers to realize SIM-based services like the trust management outlined in Section V, which are integrated into the platform. Moreover, these SIM-based services can easily be deployed by application developers when realizing mobile services.

V. SIM Based Trust Management for Mobile Services

A couple of basic security services can be provided by the SIM. This includes management of sensitive user data, basic identity management, multi-id and multi-subscriber management, basic cryptographic services like signing data and verification of signatures, and many others. Based on the architectures described in the previous sections, these services can smoothly be integrated into composite services. The management of sensitive user data for instance should be only one component of the profile data management which also includes components for handling large non-sensitive data.

In the following, we present our proposal for on-card trust management in heterogeneous service infrastructures for mobile services as an example for provisioning security, privacy, and trust services on the SIM.

A. On-Card Key Management

One of the basic concepts of our service trust management approach is that the attributes of services (including ratings resulting from previous service utilizations) are stored individually for each user. Furthermore, we propose to securely store the data on the SIM preventing manipulation of service attributes by malicious software on the mobile phone.

For identifying services, their public keys can be used, which are stored in a kind of personal key-ring on the SIM as shown in Fig. 4. Beyond security considerations, an advantage of this solution is portability of the stored data.

In addition to the public keys of services and of the user, also a private unique identifier (technically a private key) will be stored on the card (and never leave the card). This enables usage of the SIM as a personal security token for digital signatures and signature verification. It assures the authenticity of the user's signature and verifies data from other parties.

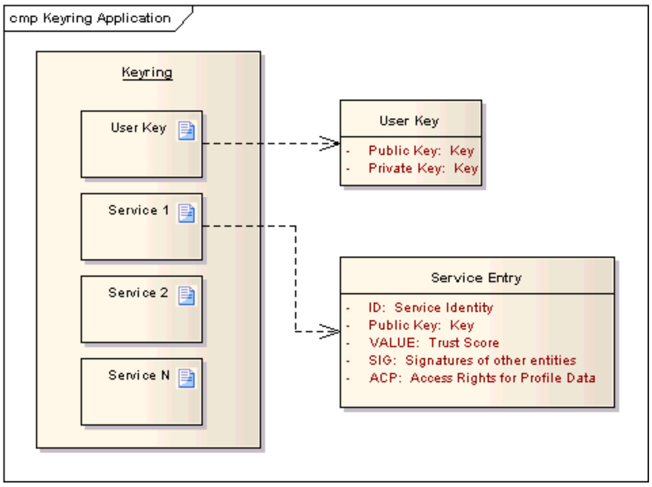
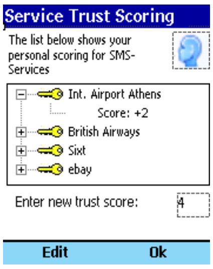


Fig. 4: Keyring on SIM card

The Trust Management Application on the SIM card provides an interface to add and delete public keys to the internal key-ring storage and to configure their attributes. The interface component is implemented as a servlet which can be used from other applications either on the SIM or on the mobile phone.

B. *Individual Trust Level Ranking*

Every key in the key-ring application can be assigned with a discrete trust value to rank the individual service trust for the user of the service, e.g. a value between 0 and 1 as outlined in the table in Fig. 6. We assume that this value will be adjusted automatically by dedicated system components, that observe each usage of a service, or based on recommendations from friends or neutral organisations. Moreover, it is conceivable that ranked public keys with trust values can be imported e.g. from an operator's database over the air or from local kiosk terminals. Finally, the user has the option to manually adjust his scoring value for a service, even though most users will probably not make use of this feature (Fig. 5).



Service Trust Scoring

The list below shows your personal scoring for SMS-Services

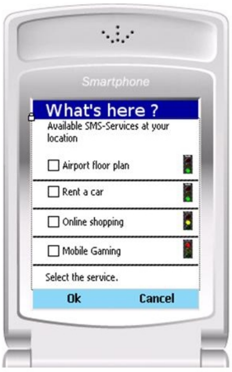
- Int. Airport Athens
  - Score: +2
- British Airways
- Sixt
- ebay

Enter new trust score: 4

Fig. 5 User managed Trust Scores

The components for on-card key management and trust level ranking are provided by the SIM based on the architecture outlined in Sections III and IV respectively. For preventing unauthorized access to these components, the enhanced security framework of JC 3.0 is leveraged, e.g. the security handler and the HTTP authentication mechanism.
The trust score stored in the SIM might not be the only parameter to determine the trust of a service. Other, especially non static parameters like the environment could also have impact on the trust representation. The final trust recommendation is calculated during service discovery and evaluated based on user specific policies. The trust ranking of a service can be indicated to user when displaying a list of available services as shown in Fig. 6. In the example the trustworthiness of each service is visualized by a small traffic light icon.

Trust Score	Meaning
0	Not trusted
0.5	Neutral trust level
1.0	Fully trusted



What's here ?

Available SMS-Services at your location

□ Airport floor plan
□ Rent a car
□ Online shopping
□ Mobile Gaming

Select the service.

Fig. 6 Indication of trust values

Furthermore, the service trust management could define specific thresholds to allow or deny critical actions.

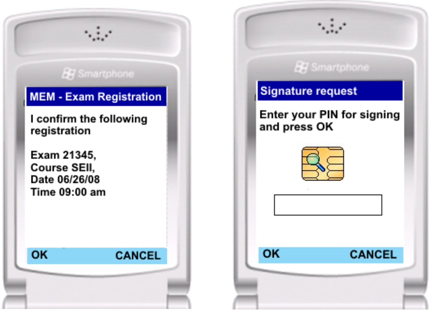
C. *Service Identity Signing*

Most currently deployed Public Key Infrastructures (PKIs) are hierarchically oriented and rely on a centralised design. But the approach of a strict and global Certification Authority would be in conflict with the open infrastructure of Simple Mobile Services. We aim at a web-of-trust-like infrastructure instead.
Service entities can sign each other. This can for example be used for chain stores. The signing party can also be a subordinate entity, e.g. an airport, which signs all airport related services. The signing information will also be stored in the keyring on card and can have impact on the trust level. Services unknown to the user could become more credible if they are signed by already trusted entities.
For the scenario of a traveller at the airport mentioned above, we assume that the traveller has stored the identity of the airport in his keyring. If now the so far unknown airport bookshop is being noticed, its trust level will be also identified as more trustable because its service identity is signed by the airport identity.
Hierarchical PKIs as well as Web-Of-Trust Infrastructures can become very complex. The suggested solution is to use a Web-Of-Trust, but with a limited signature chain. If we only attend direct signatures, instead of three levels like in OpenPGP [10, 11], we are able to simplify the problem.
Inspired by [12] we introduce a trust scoring calculation based on probabilistic argumentation for mobile service authentication where users are able to gradually rate the services. Trust values of signing entities together with the trust value of the entity itself can define an average service trust, based on a simple calculation algorithm: It must be further evaluated, if e.g. neutral trusted signatures are taken into consideration.

D. *Using digital signatures*

Above we described the use of digital signatures and the management of a key-ring application on a SIM to proof the identity of a service. This is important e.g. during service discovery to determine the individual trustworthiness of services. An example of a student exam registration is depicted in Fig. 7.

Additionally, we can find more aspects where digital signatures can be used: a lot of mobile services are commercial oriented and the user might want to conclude some kind of commercial contract. To secure the process, the use of digital signatures could be helpful in order to make contracts accountable. This implies an even higher level of security than the usage of secure connections which is common in B2C internet shops, as the signature outlasts the conclusion of the contract. It is important that this signature is based on a digital signature method that takes place on the presentation layer of the OSI model. The user must be able to see what he signs and the signature must not be lost in any transport layer or omitted in later storage.



I confirm the following registration

Exam 21345,
Course SEII,
Date 06/26/08
Time 09:00 am

Enter your PIN for signing and press OK

Fig. 7 User Centric Signing Process

A further use case is related to authorization. In general, authorisation is determined within an application once the identity of the other party is confirmed by authentication and the access privileges are assigned. However, further methods for authorisation are conceivable, allowing the user to perform certain actions or granting her certain rights that otherwise she would not have received, providing her with one-time (i.e. use only once, or time/event constrained) tokens. Several applications are possible. The user may for instance receive a time-constrained token during check-in at the airport, allowing him to buy in tax free shops until departure. Another example is a one-time token for entering the gate.

E. Exchange of keys

To enrich the trust management, different exchange mechanisms for key data are imaginable: Primary, all users could exchange keys and individual trust scoring peer to peer like personal recommendations, possibly supported by community platforms in the internet. In addition, general trustworthy institutions like the mobile operators could provide e.g. extended black and white lists to be imported in the personal keyring.

Furthermore, server backup mechanism could be implemented to support the user in case of loosing his SIM or whenever a migration to a different SIM becomes necessary.

F. Multplei Identities

Technically, the SIM card service is able to manage more than one identity including a keyring for each. The user could be enforced to deal with several identities. This could enable another level of privacy in the mobile service environment. The user is able to select the identity he is intended to use for the specific action. To simply retrieve information an anonymous identity can be used, only in case of important and/or cost intensive commercial contracts, the real identity including all personal information will be used.

VI. Conclusion

Users of future mobile services need assistance to deal with the manifold service offers and options without drawbacks for security, privacy, and trust. In order to cope with this requirement, an open architecture has been proposed which integrates the SIM as an enabler for security related services. Even though first smart card products featuring a web server and a servlet architecture are already available, the evolution to the Java Card 3.0 will bring major improvements. Cards implementing this standard will support the required protocols as well as necessary enabling technologies like multithreading. Nevertheless, our first prototypical implementations on today's Java Card platform 2.2 together with a Smart Card Web Server implementation based on [9] offers already an adequate solution.

During the remaining time of the Simple Mobile Services project, this approach will be evaluated in demonstrators, user studies, and live trials with students at the University of Rome 'Tor Vergata'. This evaluation will give indication to SIM manufacturers and mobile operators about the potential role of the SIM in future mobile services, in particular of user acceptance and business cases.

References

[1] Simple Mobile Services – Platform documentation, http://netgroup.uniroma2.it/SMSPlatformHome

[2] R. Walker, G. Bartolomeo, N. Blefari-Melazzi, S. Salsano: "MEMs - Mobile Electronic Memos: efficient information capture and sharing for mobile users", Wireless World Research Forum, Meeting 18, 13-15 June 2007, Espoo, Finland.

[3] Mobile Information Device Profile 2.0, JSR 118, http://jcp.org/en/jsr/detail?id=118

[4] "Security and Trust Services API", JSR 177, http://www.jcp.org/en/jsr/detail?id= 177).

[5] The IETF Transport Layer Security Workgroup: http://www.ietf.org/html.charters/tls-charter.html

[6] D. Wendlandt, D. G. Andersen, A. Perrig Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing, Carnegie Mellon University http://www.usenix.org/event/usenix08/tech/full_papers/wendlandt/wendlandt_html/index.html

[7] SUN Java Card specifications http://java.sun.com/javacard/

[8] R. Oeters, U. Kastens, C. Rust, L. Schnake: Benefits from realizing a location-based mobile hotel service with Java Card 3.0 Technical report, tr-ri-08-292-1, Universität Paderborn, 2008

[9] Open Mobile Alliance (OMA) Smartcard-Web-Server V1.0

[10] RFC 2440, OpenPGP Message Format, http://www.ietf.org/rfc/rfc2440.txt

[11] The IETF OpenPGP Workgroup: http://www.ietf.org/html.charters/openpgp-charter.html

[12] R. Haenni. Web of Trust: Applying Probabilistic Argumentation to Public-Key Cryptography. ECSQARU'03, 7th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty, Aalborg, Denmark, 2003

Lars Schnake – R&D Software Architect, Sagem Orga GmbH
Heinz-Nixdorf-Ring 1, 33106 Paderborn, Germany,
phone: +49 (0) 5251 889 1392
email: lars.schnake@sagem-orga.com
Carsten Rust – R&D Project Manager, Sagem Orga GmbH
Heinz-Nixdorf-Ring 1, 33106 Paderborn, Germany,
email: carsten.rust@sagem-orga.com
Rebekka Neumann - Researcher, University of Paderborn,
Software Quality Lab, Warburger Str. 100, 33098 Paderborn, Germany,
email: rneumann@s-lab.upb.de