Security Aspects of Smart Cards vs. Embedded Security in Machine-to-Machine (M2M) Advanced Mobile Network Applications

Mike Meyerstein*, Inhyok Cha, and Yogendra Shah

InterDigital Communications Corporation LLC,
King of Prussia, PA, USA
meyersmv@btinternet.com,
{Inhyok.Cha,Yogendra.Shah}@InterDigital.com

Abstract. The Third Generation Partnership Project (3GPP) standardisation group currently discusses advanced applications of mobile networks such as Machine-to-Machine (M2M) communication. Several security issues arise in these contexts which warrant a fresh look at mobile networks' security foundations, resting on smart cards. This paper contributes a security/efficiency analysis to this discussion and highlights the role of trusted platform technology to approach these issues.

Keywords: Smart card, UICC, machine-to-machine communication, embedded security, trusted environment.

1 Introduction

The idea of M2M is that un-manned terminals, e.g. traffic cameras, meters, cargo containers, can communicate with host servers using wireless global communications networks. This requires the usual secure authentication for network access.

The networks will not be specially M2M-enabled, so the authentication has to follow the standardised schemes currently in place for mobile (e.g. 3GPP) and fixed (e.g. WLAN) networks.

M2M security requirements [1] may make the conventional UICC (Universal Integrated Circuit Card) a less advantageous solution for secure authentication. It is necessary to look at the options for a non-personalised security module to which a network operator's MCIMs (Machine Communications Identity Modules) can be downloaded [1]. This may be accomplished using an embedded Trusted Environment (TRE) in a terminal. The TRE acts as a hardware root of trust for the storage and execution of secure applications and may also have protected software functions. The TRE may host downloaded software MCIMs that emulate the behavior of the USIM (Universal Subscriber Identity Module) [2] or ISIM (Internet Multimedia Services Identity Module) [3] applications.

* Mike Meyerstein is the proprietor of Meyerstein Consulting Ltd, currently providing consultancy services to InterDigital Communications Corporation.

A.U. Schmidt and S. Lian (Eds.): MobiSec 2009, LNICST 17, pp. 214 – 225, 2009.

2 M2M Requirements

The M2M market has some definitive characteristics [1]:

- Terminals may be in hard-to-reach locations (e.g. traffic cameras)
- Terminals may become geographically dispersed over time (e.g. cargo containers)
- Owners of large populations of terminals may want to change the network operator without visiting the terminals (e.g. to change the UICC).
- Terminals need to be protected against unauthorised removal of UICC
- Terminals may require over-network provisioning after sale or installation.

3 The Options for a TRE to Host Secure, Downloadable MCIMs

Client-side technologies for TREs could include

- UICC with download capability.
- An embedded TRE in the terminal, to provide a secure execution and storage environment. MCIMs would be downloaded to the TRE over public IP networks.
- Smart token such as the new multimedia card with on-card UICC (or "SMC" – Secure Multimedia memory Card)

A framework of standardised specifications is needed for the above solutions.

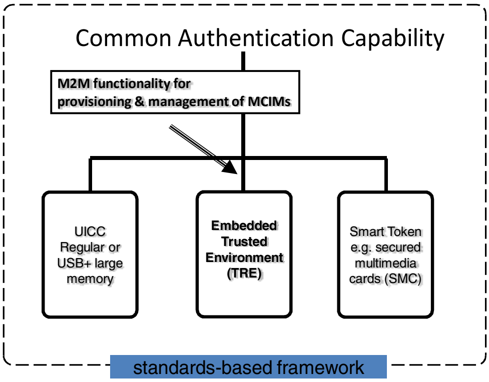
Common Authentication Capability
M2M functionality for provisioning & management of MCIMs
UICC Regular or USB+ large memory
Embedded Trusted Environment (TRE)
Smart Token e.g. secured multimedia cards (SMC)
standards-based framework

Fig. 1. Options for TRE to Host Downloaded MCIMs

The discussion within standardisation about the (dis-)advantages of the various candidate solutions is lively and far from concluded.

In Table 1 on the next page we collect the main arguments that have been advanced thus far in various forums and standardization committees (there is no reference document in which this information can be found).

Table 1. Comparison of Potential Solutions for TRE

REQUIRED FEATURES	Today's UICC	Embedded TRE	SMC
Currently standardised	good	medium (could be based partly on TCG specs)	medium
Currently available	good	medium (some limited versions available)	poor
Protection against unauthorised removal	poor (good, if M2M form factor UICC is soldered in)	good	poor
Provides secure API	good	good	not known
Does not require connector and interface chips	poor	good	poor
MCIMs can be downloaded for initial provisioning and for replacement of MIDs	poor. Can only install USIM or ISIM during manufacture	Good	poor. Can only install USIM or ISIM during manufacture
Key management suits M2M model	poor (pre-shared keys for authentication and download)	good (can use PKI (Public Key Infrastructure))	poor
Predictable costs	poor (for full-function, big memory, downloadable UICC)	Good (part of chipset)	poor
Secure channel to terminal	poor (standardised but seldom implemented)	good	not known
Remote change of operator	poor	good	poor
Open API for download	poor	good	poor
Track record of trust	good	poor	poor
Established infrastructure	good	Poor	Good (fits UICC infrastructure)
Built into operator's current trust models	good	Poor	Medium (partly fits UICC trust model)
Built into operator's current business models	good	poor	poor
Promotes operator's current brand image	good	poor	good

No solution comes out as perfect, but the embedded TRE shows promise if it can be standardised. The UICC shows promise if its current limitations (including that of lack of implementation of already-standardised features in cards and terminals) can be overcome. In the next sections, we look at issues surrounding the use of UICC as a TRE.

4 Smart Card Security in Mobile Networks: Why Is the Smart Card a Trusted Anchor?

Physical Tamper-Resistance:
An ISO -7816 [7], [8], [9] smart card has, in practice, a single-chip architecture with little possibility of monitoring communications between different chips on the card. Smart card ICs (Integrated Circuits) are designed and implemented to prevent probing and reverse-engineering. They are fabricated on a dedicated production line in a secure facility. Measures include scrambling of busses and of memory addresses, bonded passivation layers, permanently disabled test points, self-generated programming voltage.

Proprietary, Secure O/S (Operating System):
The Smart card's standardised API, e.g. [4], consists of a restricted command set that has no hidden commands or access methods. It is trusted because of its own built-in security mechanisms and because of those of the underlying hardware platform. It is non-updateable.

For applications such as USIM, the K (i.e. the pre-shared authentication key)and OTA (Over The Air) keys are stored and accessed by the O/S in proprietary ways. The O/S cannot be made to reveal the values or memory locations of those data.

Conventional GSM SIM (Subscriber Identity Module) cards did not allow adding applications to a live card. The advent of the Javacard [10] now allows applications on a multi-application UICC platform to be updated, deleted or added to an issued card, either remotely or locally. The potential of Javacard for Network Operators is currently restricted by

- Implementation by Network Operators of only SMS (Short Messaging Service) as the bearer (for OTA messages to the UICC), which has a very limited bandwidth
- OTA security standards [11][1] are not profiled for IP bearers
- Lack of a sufficiently rich terminal/UICC interface on nearly all MEs[2] (Mobile Equipments)
- General concerns about the security of multi-application Javacards.

In the world of telecoms, it is generally the buyer's task to perform due diligence tests on the smart card vendor to ensure that the O/S has been properly developed and evaluated.

Other Measures:
Smart cards include proprietary measures to prevent attacks such as slowing down the external clock and measures against power analysis attacks by the use of noise-free

[1] N.B. [11] has been recently split up and its former contents have been dispersed over [18], [19], TS 31.115 Secured Packet Structure for (U)SIM Toolkit applications and TS 31.116 Remote APDU Structure for (U)SIM Toolkit applications. [11] is still widely referred to in the telecoms sector.

[2] A few terminals have implemented the JSR177 [12] terminal/UICC interface, but it's usually only the SIM toolkit part and not the general APDU (Application Protocols Data Unit) API. A few Windows Mobile MEs have allowed an open APDU API to the UICC, using the terminal's RIL (Radio Interface Layer) but it is not clear if those are still in production.

computational algorithms and/or injection of artificial noise and/or damping of noise on the power rail. There are also said to be a large number of detailed precautionary measures taken, some of which are described in the public domain (e.g. [14]).

Design and Development Process:

Security is designed into a smart card IC in the secure facilities of a semiconductor manufacturer. The computers that are used for this are isolated from the rest of the world. Undocumented counter-measures in the IC are supported by corresponding design criteria. Once the O/S development is finished, the entire source code may be checked by an independent evaluation.

Supply Chain:

There are only a handful of world-class vendors of smart cards, so it is feasible for Network Operators (who own and specify the UICCs) to perform the necessary security audits. There will be an agreed arrangement for transferring the K objects between network operators and UICC vendors. K values cannot be retained in the vendors' personalisation systems or be discovered by system operatives.

Security Evaluations:

Card vendors have their O/S independently evaluated and MNOs perform security evaluations of their card vendors' products and facilities. GSMA (GSM association) [15] provides non-public guidance to its members on how to do that. Common Criteria Protection Profiles have been published for smart cards. One of these [16] is aimed at the underlying IC platform but some are aimed at payment cards issued by financial institutions such as Visa and Mastercard. Cost can be an issue for wider adoption of these evaluation regimes. There are no standard specifications or protection profiles for the security evaluation of telecoms smart cards such as UICCs.

UICC-Terminal Interface Security:

The UICC employs some security measures in the interface with the terminal:

- User Authentication: PINs (Personal Identity Numbers) (called CHV1 (Card Holder Verification) and CHV2 in a SIM card), provide some level of protection with user authorisation on the interface. CHV1 can be disabled by the user, in which case there is no PIN-protection for making calls. The use of CHV1 and CHV2 poses a security vulnerability since the passwords get transported across the UICC-ME interface in clear. The UICC does not authenticate itself to the user, although 3G authentication [2] provides mutual authentication of the card and the network. In M2M, a remote user could rely on two possible methods of assuring himself of the authenticity of a UICC, i.e. (a) using a remote access protocol that exploits a pre-shared or private key on the UICC and/or (b) using e.g. Liberty Alliance [22] protocols to trigger the UICC issuer to perform an authentication of the UICC and possibly binding that to the remote access session,
- Secure channels: Commands to the UICC are not secured unless they are inside a 3GPP OTA envelope [11]. That is why ETSI and 3GPP have recently specified secure channels and their key establishment methods ([5], [6]) across the terminal-UICC interface. ISO (International Organisation for

Standardisation) 7816 [20], [21] and EN726 [17] define secure messaging between a terminal and a smart card but key distribution was not defined (it being assumed that it would be based on pre-installed keys). ISO7816 also defines a set of security-related commands. Neither the ISO nor CEN (Comité Européen de Normalisation) techniques are included in UICC specifications such as [4]. ETSI (European Telecommunications Standards Institute) and 3GPP secure channel specs [5] and [6] compliment the ISO and CEN standards by defining methods for key distribution between the terminal and UICC. There does not seem to be any reason to believe that normal terminals can be trusted to store the distributed keys. Protocols such as Global Platform [10], ETSI RAM (Remote Application Management)/ RFM (Remote File Management) [18], [19] and 3GPP OTA [11] provide end-to-end security from server to card for the purpose of loading and managing files and applications on the UICC. They do not require a secure ME/UICC interface.[3]

- Protection of data across the interface: All standardised command-sets of a UICC are designed to be sent in the clear across the terminal/UICC interface. Protocols that may be subject to replay attacks must have counter-measures built in, e.g. the sequence numbers used in 3G authentication [2]. In future, the Smart Card Web Server (SCWS) [23], could use HTTPS (HyperText Transfer Protocol Secure)/TLS (Transport Layer Security) to establish a secure tunnel from card to server (or to terminal) via which usernames and passwords could be sent.
- Access control: In general, ACLs (access control lists) are not used in today's UICC O/Ss. Access to file operations relies on the principle that if the entity accessing the file can satisfy the access policy (embodied in the File Control Parameters [4]), then it must be an authorised entity.

5 Meeting M2M Requirements with UICCs

Advent of the "Big SIM" UICC

Recent innovations in smart card technology could go a long way to enabling the UICC to fulfill the M2M requirements. The new features described below, plus the ability to store downloaded applications and multimedia files, would need the large memory of "Big SIM". However, providing such features on UICCs could well be cost-prohibitive for M2M.

Large Memory: Recently, UICCs with flash memory of up to 2Gbytes have become available.

[3] Remote Application Management can theoretically be used to download any Javacard application to a UICC and to store it in a security domain. It does not currently apply to the U(I)SIM applications, as there is no standardised mechanism for the UICC to extract and store the Ki and algorithm customisation parameters. For the case of updating existing files either locally or remotely, this is possible only where the access conditions in the File Control Parameters can be satisfied. Remote (OTA) file update is possible on files in any application on the UICC, but only if the files were OTA-enabled at the time the file was created on the UICC.

High-Speed I/O: The conventional I/O speed of the UICC/terminal interface is only a half-duplex 9.6Kbit/s[4]. In order to be able to move data on and off the Big SIM in a meaningful timeframe, a USB (Universal Serial Bus) interface has been specified in [13] and [8].

Smart Card Web Server (SCWS): The advent of "Big SIM" with USB I/O enables the UICC to support an IP stack and web server. There are a number of advantages to this, e.g. use of (X)HTML ((eXtensible or) HyperText Markup Language) to communicate with the UICC. UICCs could even have their own IP addresses, which could introduce a whole set of security issues. ETSI SCP has standardised SCWS [23] and IP [23] on a UICC. Use of SMS for application download is limited in practice to about a 1kByte payload, i.e. 7 concatenated SMSs. Even then, this requires a dedicated SMS-C (SMS Centre) to achieve an effective success-rate. With ordinary SMS-Cs whose resources are shared with mainstream SMS, the practical limit may be as little as 2 concatenated SMSs, i.e. about 300 bytes. The size of a download using an IP bearer does not suffer from such limitations.

Internal Security Domains: Global Platform has specifications [10] that define security domains on a Javacard smart card. ETSI SCP (Smart Card Platform) are now expanding upon these in their specifications for "Confidential Applications." This allows the card issuer to set up domains for the use of third parties to load applets onto the card. The issuer cannot examine those applets. The UICC provides a sandbox environment in which the domains are isolated from each other – a feature that has been somewhat limited in Javacard implementations up to now.

Enhancements Required to UICCs for M2M Mass Market

A UICC to be used in mass-market M2M applications would have to support the requirements of long life-time with long maintenance intervals, non-removability, remote download of operator's authentication application and remote change of operator. Some very significant enhancements need to be considered as follows:

Security Domains: Support is required for security domains for the card issuer and for third parties, e.g. as per the SCP specifications "Confidential Applications" concept. But in the M2M scenario, network operators would be classed as third parties. In order for the card issuer to allow the M2M equipment owner to change to a new network operator, the card issuer (who is therefore not a network operator) assigns a domain to a new network operator and closes the domain of the old network operator.

Removable UICC vs. Downloadable UICC: unauthorised removal of a traditional "removable" UICC must be made very difficult, while its replacement must be easy. A better alternative to this is to fix the UICC in the terminal and for it to support the ability to download MIDs. Such a UICC must be able to extract K objects and similarly sensitive data from messages from a remote server and lock them away in secure memory so that they cannot be revealed to entities outside the UICC. Network operators will demand that there be no reduction of security in UICCs that support these features.

Download Protocols: Support is required for secure download protocols other than standardised OTA, i.e. M2M requires protocols which do not require pre-shared keys

[4] Somewhat higher speeds can be negotiated between ME and card, but all must support the default speed for backward compatibility.

and which can be used over IP bearers. (In this respect, support for SCWS and IP stack could be an advantage).

Secure Interface to UICC: Support for a secure terminal-UICC interface [5], [6] may be a requirement, as discussed above.

6 Security Analysis and Comparison: Can an Embedded TRE Ever Be as Secure as a Smart Card?

Successful, Publicised Attacks Against Smart Cards
Smart Card technology had experienced some crises in the past, as follows:

Side Channel Attacks (specifically Power Analysis): This type of attack relies on the noise on the smart card's power contact being correlate-able with the processing that is going on in the UICC, especially when it is reading the values of a secret key into its internal registers. This can work well on an unprotected card that uses the DES (Data Encryption Standard) algorithm. RSA (Rivest, Shamir, Adleman) is not susceptible and neither is AES (Advanced Encryption Standard) (upon which 3GPP Milenage is based).[5] It is easy to prevent this type of attack at the design stage of a smart card. Likewise, an embedded TRE would be designed and implemented such that it would not leak this information. There would be no such interface that an attacker could monitor and the processing in the TRE would be such that any noise occurring on any power rail would not contain any recoverable information.

Probing of Broadcast TV Cards: Satellite TV smart cards have to contain global decryption keys, due to the nature of the service, i.e. a broadcast encrypted signal. If you crack one card, you have cracked the whole scheme. This is, of course, not true of networked telecoms systems. In the early 1990s, Cambridge University in the UK used physical probing to successfully recover secret keys from satellite TV cards. It is not necessary for embedded TREs (or UICCs, for that matter) to contain global keys and their embedded nature would make physical probing infeasible.

Cloning of SIM cards: The only verifiable instance of cloning of 2G SIM cards was a "known plaintext" (50,000 challenge-response pairs) attack against the COMP128 authentication algorithm and not against the card platform itself. This attack has been prevented by the use of better algorithms and is not at all possible with 3G. The relevance of this attack to TREs is that it has gone down in the annals of urban mythology as an attack on the SIM card, rather than an attack on the algorithm. "Mud sticks," as they say.

Attacks Against Disposable "Eurochip" Phonecards: Two successful attacks exploited (1) a weaknesses in the card's hard-wired logic and (2) the absence of a security module in some payphones. The relevance of this to embedded TRE is that many attackers seem able to acquire insider knowledge of the product. "Security by Obscurity" is not a viable counter-measure.

Lack of Security Specifications and Formal Evaluations for UICC: Tamper-Resistance: There are no standardised specifications as such that assure the degree of

[5] This attack was successfully perpetrated against the pilot of a well publicised smart card payment system in the UK in the early 1990s. Although the scheme was designed to use RSA with unique private keys, the pilot used DES with a global key on every card.

physical and logical tamper-resistance described above. It is up to the buyer to specify what he wants, although, in theory, protection profiles such as [16] could be adapted for UICCs and evaluations enforced by buyers. The same might also be true of embedded TREs, unless they use TCG-style technology and also conform to suitably-created and enforced protection profiles (which might need to be wider in scope than those specified so far by TCG for Trusted Modules (see http://www.bsi.de/zertifiz/zert/reporte/pp0030b.pdf).

Perceptions vs Real-World Implementation of UIC**C:** There is never a guarantee that, for a given advertised UICC product, all or any of the possible counter-measures have been implemented. It is a case of being an informed buyer. Network Operators can obtain un-published information about counter-measures from card vendors and large-volume buyers can specify their own counter-measures, within the constraints of the silicon manufacturing process. This should be borne in mind when some commentators argue that an embedded TRE cannot be as secure as a UICC.

Secure Terminal/UICC Channel**:** Even if the ETSI/3GPP specifications concerning key establishment [5], [6] are implemented, it is not specified how the terminal securely stores the local key and executes the algorithms. There have been failed attempts in the past (e.g. the MET – Mobile Electronic Transactions forum) to portray the terminal as a PTD (Personal Trusted Device). It would be fair to say that the telecoms industry has little confidence in the ability of terminal suppliers to implement a secure environment.

Arguments Against Trusting a Non-UICC TRE

Some network operators may have the following perceptions, as expressed in some draft versions of [1], about a non-UICC TRE in a terminal. Counter-arguments are in italics.

- Network operators can trust the UICC because they are in charge of the specifications and they buy them directly from their approved vendors. They will not be able to do this for TREs. *It will be necessary to have an international accreditation scheme for TREs.*
- The telecoms industry has little confidence in the ability of terminal suppliers to implement a secure environment (see above). This is not helped by the suppliers who are involved in standardising M2M but who keep trying to limit the functionality of the TRE. *Once again, a security accreditation scheme is needed.*
- Compared to the small number of UICC vendors, there are potentially many suppliers of M2M terminals and therefore of embedded TRE solutions. *As is the case with UICCs and secure semiconductor products, the number of companies that decide to position themselves as TRE manufacturers could turn out to be limited.*
- The UICC is a product with a great track record. The O/Ss have been evolved over many years. A non-UICC TRE is still at the conceptual stage. *No counter argument except that every technology is eventually superseded by progress. UICCs were new, once upon a time.*
- Network operators want to use OTA infrastructure for downloading MIDs. Current OTA is not secure enough and would need to be updated, but that

could cause backward-compatibility problems. Other protocols, which could be used over an IP bearer, could well be secure enough but would have to be evaluated. *M2M equipment vendors will have no problem in providing the required APIs to an embedded TRE to allow use of an IP bearer and appropriate download protocols. Network Operators could possibly be persuaded, since some of them were very active in Liberty Alliance [22]. Also, IP download could be sub-contracted.*

- Today's system for distributing K from UICC manufacturers to Network Operators is well understood and well tried. Network Operators take the view that the key distribution required in the proposed Candidate Solution 1 in [1] would involve passing the K around an increased number of entities, which could introduce vulnerabilities. Scale-ability of proposed solutions in [1] is also an issue. *The number of parties involved with the key distribution does not have to increase, since the required functions can be combined into a small number of real-world roles. Also, K could be end-to-end encrypted (with the TRE's public key) by the entity that generates it. The system solution for embedded TRE in [1] is intended to be scale-able, requiring normal roaming agreements for initial connectivity and standardized, PKI-based protocols for TRE validation/authentication and MCIM download.*

7 Conclusions

An embedded TRE, if properly specified, implemented and certified, could provide a much-needed alternative to the UICC in meeting the requirements of M2M.

An embedded TRE challenges the long-established infrastructure and practices of Mobile Network Operators and UICC suppliers, but it could be a key enabler for the potentially huge M2M market to take off, thereby becoming an important new source of revenue.

If an embedded TRE is to be a competitor to the UICC for M2M, it must have sufficient resistance to the threats that are described in the threat analysis in [1]. Ongoing co-operation between TCG and standardization bodes involved with communications aspects of M2M could produce appropriate security specifications.

Whether or not the previously successful attacks described above could be used to perpetrate such threats depends on the implementation of the embedded TRE, e.g. whether it is a discrete component, part of a single-chip CPU (Central Processing Unit), system ASIC (Application-Specific IC), or distributed across a chipset. It is not possible to provide an implementation-independent answer to the question: "How difficult would it be for an embedded TRE to have some of these measures?" because some of the measures might not be relevant. An ASIC makes things easier in terms of security evaluations and accreditations.

Specifiers, developers and implementers of embedded TREs must bear in mind the lessons which a long experience of smart cards has taught us:

- Side-channel attacks would be much more difficult to achieve against a TRE if the TRE is not a discrete component
- One can argue that direct probing of a TRE is not important, as an individual TRE does not contain any global secrets. Nevertheless, there are always

people who will take up the challenge of attacking the TRE, for example, in an attempt to extract a K. Such threats are faced by every security initiative and only provide further motivation to design it correctly.

- Attacks against the cryptographic algorithms in a TRE could be publicised as attacks against the TRE.
- Many attackers seem able to acquire insider knowledge of the product. “Security by Obscurity” is not a reliable counter-measure. Do not rely on attackers being unable to gather information on how a TRE works.
- Reputation is important. The UICC vendors have enormous reputations in the world of security, whereas the terminal suppliers have little or none. TREs should be manufactured and personalised by reputable chip manufacturers with a proven track record in the security industry.
- Perceptions are important. The perception of the UICC may, in some cases, exceed its actual specification or design. Such must not be allowed to become the case for TRE.

References

1. 3GPP TSG WG3 unapproved draft technical report TR33.812 (current version 8.0.0) Feasibility Study on Remote Management of USIM Application on M2M Equipment (this is a working title which can change at any time)
2. 3GPP TS 31.102; Characteristics of the USIM Application
3. 3GPP TS 31.103; Characteristics of the ISIM Application
4. ETSI TS 102 221: UICC-Terminal interface; Physical and logical characteristics
5. ETSI TS 102 484 Smart Cards; Secure Channel between a UICC and an end-point Terminal
6. ETSI TS 33.110 Key establishment between a UICC and a terminal
7. ISO 7816-1 Identification cards – Integrated Circuit Cards - physical characteristics
8. ISO 7816-2 Identification cards – Integrated Circuit Cards - dimensions and location of contacts. AMD1= assignment of C4 and C8 (2004)
9. ISO 7816-3 Identification cards – Integrated Circuit Cards - electrical interface & Tx protocols
10. Global Platform specifications, v 2.2, may be, downloaded from http://www.globalplatform.org
11. 3GPP TS 23.048 Security Mechanisms for SIM Toolkit Application; Stage 2. N.B. this has been recently split up and its former contents have been dispersed over [18], [19], TS 31.115 Secured Packet Structure for (U)SIM Toolkit applications and TS 31.116 Remote APDU Structure for (U)SIM Toolkit applications. TS23.048 is still widely referred to in the telecoms sector
12. Java community specification JSR177: Security And Trust Services API
13. ETSI TS 102 600: Characteristics of the USB Interface
14. Rankl, Effing: The Smart Card Handbook, 3rd edn. Wiley and Sons, Chichester
15. See, http://www.gsmworld.com
16. Eurosmart: Smart Card IC Protection Profile, PP-0002, first published (2001) (EAL4 augmented)
17. CEN standard EN726 Identification card systems. Telecommunications. Integrated circuit(s) cards and terminals. There are 7 parts to this standard

18. ETSI TS 102 225: Secured packet structure for UICC based applications
19. ETSI TS 102 226: Remote APDU structure for UICC based applications
20. ISO 7816-4 Identification cards – Integrated Circuit Cards -Organisation, security and commands for interchange
21. ISO 7816-8 Identification cards – Integrated Circuit Cards - Commands for security operations
22. Current specifications for ID-FF and ID-WSF, `http://www.projectliberty.org`
23. TS 102 483: UICC-Terminal interface; Internet Protocol connectivity between UICC and terminal