Project Number	**:**	**AC095**	**:**	
Project Title	**:**	**ASPeCT**	**:**	**Advanced Security for Personal Communications Technologies**
Deliverable Type	**:**	**P (Public)**		

CEC Deliverable Number	**:**	**AC095/GD/W24/DS/P/04/1**
Contractual Date of Delivery to the CEC	**:**	**Y01 / M06 (August 1996)**
Actual Date of Delivery to the CEC	**:**	**Y01 / M06 (August 1996)**
Title of Deliverable	**:**	**Report on the use of UIMs for UMTS**
Work packages contributing to deliverable	**:**	**WP 2.4**
Nature of Deliverable	**:**	**R (Report)**
Authors	**:**	**Eric Johnson Achim Müller Jason Brown**

Abstract:

This report investigates the usage of UIMs in UMTS. Although it is intrinsically assumed that such a UIM will be an integrated circuit or smart card, parts of the discussion are appropriate whatever the UIM format.

Consideration is given to the gradual introduction of UMTS as a migration from second generation systems, in particular, the GSM SIM. The impact of this migration on the UIM and the existing mechanisms which can be used to achieve this migration are considered.

The new market opportunities which could become available if UIMs support both the UMTS application as well as additional applications are indicated. These additional applications can be from sectors other than the telecommunications sector. This is coupled with a discussion of the requirements and consequences of supporting multiple applications on a smart card.

Finally, the possibilities and advantages of using biometric techniques as a user identification mechanism in UMTS are discussed. These are compared to the baseline reference of a PIN based verification.

Keyword List:

ACTS, ASPeCT, card applications, SIM, GSM, migration, security, smart card, UIM, UMTS

Table of Contents

1 Introduction 5
1.1 Authors 5
1.2 Document History 5
2 References 6
3 Abbreviations 8
4 UIM Migration path 9
4.1 Introduction 9
4.2 Second to third generation migration scenarios 9
4.2.1 Second and third generation systems 9
4.2.2 GSM Phase 2+ 10
4.2.3 ASPeCT migration scenario 10
4.2.4 Impact on smart card 11
4.3 Enhanced service concepts 13
4.4 Over-the-air value added services 13
4.4.1 Introduction 13
4.4.2 Standard SMS 14
4.4.3 Interpreted SMS 14
4.4.4 Applications of Interpreted SMS 15
4.4.5 File management features 15
4.4.6 User interface 16
4.5 The SIM Application Toolkit 16
4.5.1 Introduction 16
4.5.2 Proactive SIM commands 16
4.5.3 SIM originated short messages 17
4.5.4 Data download to SIM 17
4.5.5 The more time mechanism 18
4.5.6 Call control by the SIM 18
4.5.7 The Toolkit as a platform for customised services 18
4.6 Requirements on the functionality of a UIM 18
4.6.1 Introduction 18
4.6.2 Security services 19
4.6.3 Operating system features 22
4.7 A preliminary view of UIM migration 25
4.7.1 Introduction 25
4.7.2 The SIM as a starting point 25
4.7.3 Migration of services and security 25
5 Multiple Applications on UIMs 28
5.1 Introduction 28
5.2 Structure of SMART CARD Applications 28
5.2.1 File Structure 29
5.2.2 Commands 30
5.2.3 Transport Protocol 30
5.3 ISO / ETSI / EMV Interoperability 31
5.3.1 Commands 32
5.3.2 File Structure 38
5.3.3 Data Elements 39
5.3.4 Security 40
5.3.5 Logical Channels 41
5.4 Problems with Multiple Applications 41
5.4.1 Selection 41
5.4.2 Independence of Applications 42
5.4.3 Mutiple PINs 42
5.4.4 Protocol 43
5.4.5 Physical & Electrical Compatibility 43
5.4.6 Blocking / Unblocking / Disable 44
5.5 Benefits of Multi-Application Cards 44

5.5.1 Shared Code & Data 44
5.5.2 Delayed Loading of Applications 44
5.6 Legal issues 46
5.7 User Acceptance 46
5.7.1 Ease of Use 46
5.7.2 Common PIN 46
5.7.3 Anonymity 46
5.8 Market Opportunities 47
5.9 Implementation Example using STARMAG 47
6 Biometric methods for user identification 48
6.1 Introduction 48
6.2 Cardholder Verification Methods 49
6.2.1 Objectives 49
6.2.2 Features 50
6.2.3 Performance 50
6.2.4 Threats 52
6.2.5 Survey of CVMs 52
6.3 CVM in Mobile Telephony 54
6.3.1 General considerations 54
6.3.2 An architecture for SIMs 55

Executive Summary

This document considers the usage of smart cards as a secure means to access telecommunication services. The interest lies in the transition from second generation to third generation telecommunication systems and in the consequences such a process will have on the design of User Identity Modules (UIMs). This study takes place in a changing context – the definition as well as the implications of third generation systems are currently undergoing an intensive discussion both in the public and in the relevant standardisation bodies.

The expectations of users, network operators and value added service providers must be considered in addition to the technological constraints when developing the specification of a UIM. Consideration of the UIM is imperative when developing a framework for the migration from GSM SIMs, as a well known example of a second generation system's card, towards the more enhanced UIM which is intended for use in the Universal Mobile Telecommunications Systems (UMTS). This migration path will be the starting point of this document. The selection of topics was done in order to emphasise the evolutionary character of migration which is not a one-step change but a smooth transition.

Major links between today and tomorrow are identified and it is shown that a transition can be based on available tools – especially the downloading facilities provided by GSM Phase 2+ features like the SIM Application Toolkit. This paves the way up to a dynamic allocation and adaptation of data and software which is stored in a smart card. Then a security framework is sketched for the Interpreted SMS that can be employed for secure application management, memory reallocation or operating system customisation. This security architecture is mainly based on cryptography and in particular mechanisms for access control, authentication, signature generation or verification and dynamic key management.

Large parts of the migration chapter are based on a card manufacturer's point of view. A different perspective underlies the discussion of multi-application cards and their security features. A key factor in such cards is the facilities a card should offer in order to allow a flexible and viable management of the different applications it supports. This reflects an application provider's, service provider's or a network operator's point of view and is intended to give a technology independent overview of things to come. Some more technical sections on interoperability between ISO, ETSI and EMV concepts for command structures, file management and security concepts follow. This shows that there are already different solutions for the same problems which are more or less compatible. The discussion then turns on to implementation oriented issues, for example which problems arise when multiple applications are implemented on the same card. These may include different requirements on the physical interface between the card and the terminal, different protocols for the transfer of data units or simply conflicting security concepts. The chapter is concluded with some general comments on legal issues and user acceptance of multi-application cards or opportunities of converging markets.

Last but not least, an overview of existing technologies for biometrical user identification completes this document and shows that there are alternatives to the dominant opinion that a smart card or an application within it must always be protected by a personal identification number. The most promising approach is voice recognition which perfectly fits with the properties of today's handsets for the GSM network – a speech processing capability is already present. Different concepts are studied, for example identifying voice patterns in combination with random phrases. The common prejudices against biometrical identification could be overcome if voice recognition is combined with security mechanisms which avoid replay attacks.

1 Introduction

This deliverable is the first technical one produced by Work Package 2.4. It is entitled *Report on the Use of UIMs for UMTS* and gives an overview of concepts which will be important for the role the UIM may play. It concentrates on more general properties of third generation telecommunication systems and UMTS in particular.
The use of UIMs is determined by different factors. One factor is that not everything that may be desirable can be supported by a smart card for technological reasons. The feasibility perspective underlies Chapter 5 which is concerned with smart card multi-functionality.
Chapter 4 on migration demonstrates that there are nevertheless powerful links between today's SIM and a future UIM. This is completed by a review of biometrical methods for user identification in Chapter 6.
This document was jointly produced by G&D and Vodafone. The split of work is detailed in the first subsection.

1.1 Authors

Eric Johnson	GIESECKE & DEVRIENT GMBH Prinzregentenstraße 159 D-81 607 München Germany	Phone: +4989 4119 944 Fax: +4989 4119 905 X.400: c=de;a=cwmail; p=g+d; s=johnson; g=eric
Achim Müller	GIESECKE & DEVRIENT GMBH Prinzregentenstraße 159 D-81 607 München Germany	Phone: +4989 4119 547 Fax: +4989 4119 540 X.400: c=de;a=cwmail; p=g+d; s=mueller; g=achim;
Jason Brown	VODAFONE Ltd. The Courtyard 2-4 London Road Newbury Berks RG14 1JX England	Phone: +441635506397 Fax: +44163531127 X.400: c=gb;a=gold 400; s=Brown;g=Jason; Jason.Brown@vf.vodafone.co.uk

1.2 Document History

Revision	*Date*	*Changes*
A	19/07/96	Preliminary Draft
B	22/08/96	Draft for review by ASPeCT (chapter on migration rewritten)
C	27/08/96	Final Review Version
D	30/08/96	Final Version for review by PMC
1	04/09/96	Issued Version

2 References

Chapter 4

[1] ETS 300 506 (GSM 02.09), September 1994 : Security aspects

[2] prETS 300 509 (GSM 02.17), May 1994 : Subscriber Identity Module (SIM) Functional Characteristics

[3] ETSI/TS/SMG-091111QR2 (GSM 11.11), July 1996 : Specification of the Subscriber Identity Module – Mobile Equipment (SIM-ME) interface

[4] ETSI/TS/SMG-091114Q (GSM 11.14), February 1996 : Specification of the SIM Application Toolkit for the Subscriber Identity Module – Mobile Equipment (SIM-ME) interface

[5] prETS 300 534 (GSM 03.20), March 1996 : Security related network functions

[6] prETS 300 536 (GSM 03.40), June 1996 : Technical Realization of the Short Message Service (SMS) Point-to-Point (PP)

[7] ETSI/PAC16(96) Part B (Main Report), July 1996 : Global Multimedia Mobility – A standardization framework for Multimedia Mobility in the Information Society

[8] IEEE Personal Communications, Feb 1995 : UMTS: Targets, System Concept, and Standardization in a Global Framework – Juha Rapeli

[9] ETSI/RTR/SMG-050001 (UMTS 00-01), May 1996 : Work programme for the standardization of the Universal Mobile Telecommunications System

[10] ETSI/DTR/SMG-050104 (UMTS 01-04), Sept 1995 : Scenarios and consideration for the introduction of the Universal Mobile Telecommunications System

[11] ETSI/DTR/SMG-050901 (UMTS 09-01), June 1995 : Security Principles for the Universal Mobile Telecommunications System

[12] ETSI/DTR/SMG-050902 (UMTS 09-02), May 1996 : Security Studies for the Universal Mobile Telecommunications System

[13] ETSI/DTR/SMG-050201 (UMTS 02-01), July 1995 : Framework for Services to be supported by the Universal Mobile Telecommunications System

[14] ETSI/DTR/SMG-0102201 (UMTS 22-01), June 1996 : Service aspects/principles for the Universal Mobile Telecommunications System

[15] DuD Fachbeiträge, Digital Signatur & Sicherheitssensitive Anwendungen, 1995 : Zurechenbarkeit, Verbindlichkeit, Nichtabstreitbarkeit – Siegfried Herda

[16] IEEE Press, Contemporary Cryptology, 1991 : Chapter 7, A Survey of Information Authentication – Gustavus J. Simmons

[17] John Wiley & Sons Inc.,1996 : Applied Cryptography (Second Edition) – Bruce Schneier

[18] ASPeCT Deliverable D05 (WP2.1), Migration Scenarios

[19] FIPS PUB 180-1 : 1995 – Secure Hash Standard

[20] RIPE MD, Ripe Integrity Primitives, Final Report of RACE Integrity Primitives Evaluation (R1040) June 1992

[21] ACTS AC095, ASPeCT Deliverable D05, Migration Scenarios

Chapter 5

[1] ISO/IEC 7816-1 : 1987– Identification cards – Integrated circuit(s) cards with contacts – Part 1: Physical characteristics.

[2] ISO/IEC 7816-2 : 1988– Identification cards – Integrated circuit(s) cards with contacts – Part 2: Dimensions and location of the contacts.

[3] ISO/IEC 7816-3 : 1989 – Identification cards – Integrated circuit(s) cards with contacts – Part 3: Electronic signals and transmission protocols.

[4] Ammendment 1: 1992 to ISO/IEC 7816-3 : 1989 Protocol type T=1, asynchronous half duplex block transmission protocol.

[5] Ammendment 2: 1994 to ISO/IEC 7816-4 : 1989 Revision of protocol type select.

[6] ISO/IEC 7816-4 : 1995 – Identification cards – Integrated circuit(s) cards with contacts – Part 4: Interindustry commands for interchange.

[7] ISO/IEC 7816-5 : 1994 – Identification cards – Integrated circuit(s) cards with contacts – Part 5: Numbering system and registration procedures for application identifiers.

[8] ISO/IEC 7816-6 : 1995 – Identification cards – Integrated circuit(s) cards with contacts – Part 6: Inter-industry data elements.

[9] ISO/IEC WD 7816-8 – Identification cards – Integrated circuit(s) cards with contacts – Part 8: Security architecture and related interindustry commands.

[10] prEN 726-5 : Terminal Equipment (TE); Requirements for IC cards and terminals for telecommunications use, Part 5 – Payment methods

[11] GSM 11.11 Version 5.30 July 1996: "Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module – Mobile Equipment (SIM-ME) interface"

[12] Europay, MasterCard, and Visa (EMV): June 30 1996 – Integrated Circuit Card Specification for Payment Systems

[13] Europay, MasterCard, and Visa (EMV): June 30 1996 – Integrated Circuit Card Application Specification for Payment Systems

[14] Europay, MasterCard, and Visa (EMV): June 30 1996 – Integrated Circuit Card Terminal Specification for Payment Systems

[15] ISO/IEC CD 9992-2 – Financial transaction cards – Messages between the integrated circuit card and card accepting device – Part 2 : Functions, messages (commands and responses), data elements and structures.

[16] ISO/IEC 8825 : 1990 – Information technology – Open Systems Interconnection – Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1).

[17] ISO/ECT JTC1/SC17 N1020, ISO/IEC CD7816-3 – Identification circuit(s) card with contacts – Part 3: Electronic signals and transmission protocols.

Chapter 6

[1] SJB Services, England, 1996: The Biometrics Report - Emma Newham

[2] SJB Services, England, September 1994 – August 1996: Biometric Technology Today

[3] Fifth eurocheque SECURITY SYMPOSIUM, Brussels, 19th-21th December 1991: "Smart Cards, Authentication and Biometrics" - Dr.Klaus Vedder

[4] Electronic Times, December 1995: "Smartcard uses neural net to recognise faces" - Maureen Coulter

[5] Computers & Security, 14 (1995): "Biometrics, Is it a viable Proposition for Identity Authentication and Access Control?" - Hyun-Jung Kim

3 Abbreviations

ACCL	Authentication Capability Class
ADN	Abbreviated Dialling Number
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation 1
ATR	Answer to Reset
ASPeCT	Advanced Security for Personal Communications Technology
BER-TLV	Basic Encoding Rules - Tag Length Value, for ASN.1 notation
CA	Certification Authority
CAMEL	Customised Applications for Mobile network Enhanced Logic
CLA	Class Byte in an APDU
COS	Card Operating System
CVM	Cardholder Verification Method
DCS1800	Digital Cellular System on 1800 Mhz band
DECT	Digital Enhanced Cordless Telecommunications
DES	Data Encryption Standard
DF	Dedicated File
DL	Discrete Logarithm
DSA	Digital Signature Algorithm
EC	Elliptic Curve
EF	Elementary File
EMV	Europay, Mastercard and Visa
ETSI	European Telecommunications Standards Institute
F	Finite Field
FCI	File Control Information
FPLMTS	Future Public Land Mobile Telecommunications System
GF	Galois Field
GSM	Global System for Mobile Communication
IMSI	International Mobile Subscriber Identity
INS	Instruction Byte in an APDU
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITU	International Telecommunication Union
LAN	Local Area Network
MAC	Message Authentication Code
MMI	Man-Machine Interface
OTA	Over-the-Air
PIN	Personal Identification Number
PLMN	Public Land Mobile Network
PUK	Personal Unblocking Key
RNG	Random Number Generator
SC	Short Message Centre (Service Centre)
SCP	Services Control Point
SFI	Short File Identifier
SHA-1	Secure Hash Algorithm
SIM	Subscriber Identity Module
SME	Short Message Entity
SMS	Short Message Service
TPDU	Transfer Protocol Data Unit
UIM	User Identity Module
UMTS	Universal Mobile Telecommunication System
VAS	Value Added Service

4 UIM Migration path

4.1 Introduction

This chapter deals with migration from second to third generation telecommunication systems specifically with respect to the smart card. It starts with a short review of migration scenarios, both in general terms and as they are defined in the ASPeCT project, and outlines enhanced service concepts for third generation systems.
Although the functionality of a UIM is not yet specified, it is clear that some requirements on it can be identified today. For example, the Over-the-Air feature as it stands for the GSM SIM must in some way be supported by a UIM too. This is necessary to allow the provision of value added or customised services. Another important requirement is that a UIM offers a large suite of cryptographic modules and security services.
It is the goal of this chapter to show that the concepts for security and service provision as they are envisaged for UMTS can be supported by a smart card. It is important to identify general links between the different system generations and to consider how they reflect on card migration.
One such link is the SIM Application Toolkit. Together with the Short Message Service and appropriate security functionality, it is flexible enough to provide an OTA framework which is powerful and allows to adapt to changing demands from a lot of parties involved in migration. These ideas define the structure of this chapter that concludes with a preliminary view of UIM migration.

4.2 Second to third generation migration scenarios

4.2.1 Second and third generation systems

In this subsection, we discuss briefly the classical vision of second and third generation mobile telecommunication systems and then examine current trends in evolution from the second to the third generation.
From a traditional perspective, second generation mobile telecommunication systems generally relate to a single operating environment or application. For example, GSM is a wide area cellular standard, DECT a cordless standard and TETRA a Private Mobile Radio (PMR) standard. Services are almost exclusively circuit switched, relate to a single media (speech, data, short message etc.) and can be considered to afford relatively low user data rates (typically around 9.6kbits/s in a wide area cellular environment).
In contrast, again from a traditional perspective, third generation mobile telecommunication systems can be characterised as follows:

4.2.1.1 Scope

A third generation system is expected to encompass a number of operating environments/applications:

- Wide area cellular
- Small area cordless (including wireless PABX, residential, Telepoint)
- Wireless LANs
- Private mobile radio
- Mobile satellite
- Fixed access

Different air interfaces will undoubtedly be required for the different (wireless) environments / applications in order to optimise performance. In addition, for predominantly political reasons, different air interfaces will be required for the same (wireless) environments/applications in different regions of the world.

4.2.1.2 Services

Third generation systems are intended to support:

- Both circuit and packet switched services
- Services based upon very low and very high bit rates (up to 2Mbits/s in certain environments)
- Multi-media services
- Virtual home environment i.e. roaming support of network operator/service provider specific services

4.2.1.3 Forward Compatibility

Ideally, third generation systems will be forward compatible. In particular, terminals will be able to adapt to new or unfamiliar air interfaces and accept software downloads/updates from the network (with appropriate security!).

4.2.1.4 Service Provision

In a third generation system, it is expected that the activities of service provision and network transport will be independent. Thus a network operator will not necessarily offer services.
Standardisation of third generation mobile systems is being undertaken within the ITU in the form of the Future Public Land Mobile Communications System (FPLMTS) and within ETSI in the form of the Universal Mobile Telecommunication System (UMTS). The assumption is that UMTS will be a regional variation of FPLMTS.

4.2.2 GSM Phase 2+

Without a doubt, the GSM second generation system for mobile communications is heading the evolution race towards the third generation. There are a number of distinct areas in which GSM as part of the Phase 2+ programme of developments is evolving towards the vision of a third generation system:

4.2.2.1 Scope

There are GSM Phase 2+ work items on:

- GSM900/DECT interworking
- GSM900/DCS1800 dual band operation
- GSM900/Satellite system interworking
- Support of PMR type services, officially known as **A**dvanced **S**peech **C**all **I**tems (ASCI):
  - **e**nhanced **M**ulti-**L**evel **P**recedence and **P**re-emption (eMLPP)
  - **V**oice **B**roadcast **S**ervice (VBS)
  - **V**oice **G**roup **C**all **S**ervice (VGCS)

4.2.2.2 Services

There are GSM Phase 2+ work items on:

- Services intended to offer the potential for end-to-end packet data transfer
  - **P**acket **D**ata on signalling channels **S**ervice (PDS)
  - **G**eneral **P**acket **R**adio **S**ervice (GPRS)
- Services and features intended to offer higher data rates over the air interface
  - **H**igh **S**peed **C**ircuit **S**witched **D**ata (HSCSD)
  - New channel coding schemes
  - Data compression schemes
- Virtual home environment type services
  - CAMEL

4.2.2.3 Forward Compatibility

The GSM Phase 2+ work item on the SIM Application Toolkit, which facilitates software download over the air interface to the SIM (among other features), can be thought of as a first step towards a forward compatible system.

4.2.3 ASPeCT migration scenario

A migration scenario in this context defines a specific path from one or more second generation systems to a single ubiquitous third generation system. The path is specified according to the order in which a number of evolutionary events occur e.g. the introduction of a new air interface, network

upgrades, the introduction of new services etc. However, identification of plausible migration scenarios involves consideration of a number of different aspects including triggering events, techno-economics related aspects, market demands and regulatory issues. These underlying aspects are considered in more detail in the ASPeCT deliverable D05 [18]. It is sufficient to state here that different network operators will follow different migration paths because of such factors as available resources, market priorities and the prevailing regulatory environment in the respective country.
It is clear from the previous discussion that a network based migration scenario can be described by a number of 'network levels' separated by 'transition phases'. In ASPeCT D05, a possible network based migration scenario is described consisting of four network levels. This is the ASPeCT migration scenario and is described briefly here.

4.2.3.1 ASPeCT level 1 : Initial Network Level

The initial network level relates to a network operator employing the same core network infrastructure to support GSM900, DCS1800 and DECT subscribers/users. This can be considered to be tantamount to a GSM Phase 2+ network. The justification for considering GSM Phase 2+ as the initial network level is that, although GSM Phase 2+ features are not yet implemented in any GSM network, they are reasonably well defined and provisional timescales for their rollout are already in place. Thus, the uncertainty regarding a migration scenario from GSM to UMTS only relates to the post GSM Phase 2+ period.

4.2.3.2 ASPeCT level 2 : Introduction of UMTS Users

UMTS users are introduced at this level. It may be that these UMTS users are only able to exploit the services and features which are available to the GSM900/DCS1800/DECT users. However, it may be possible in addition to emulate certain UMTS specific services and features for such users, even in the absence of a dedicated UMTS air interface and fixed network infrastructure. In any case, only a limited set of UMTS services and features can be offered at this level. In particular, high bit rate and/or multimedia services will not be available.

4.2.3.3 ASPeCT level 3 : Introduction of UMTS Air-interface(s)

The dedicated UMTS air interfaces are introduced at this level. When communicating via a UMTS air interface:

- UMTS users should be able to exploit some UMTS specific services, although it is unlikely that the full range of such services will be available because of limitations in the fixed network.
- GSM900/DCS1800/DECT users should be able to access their normal services, as emulated over the UMTS air interface.

It should be noted that the UMTS air interfaces (with the possible exception of the satellite component) are unlikely to be ubiquitous for a considerable time after their introduction, and hence UMTS users will still be accessing services via the GSM900/DCS1800/DECT air interfaces in a substantial proportion of cases.

4.2.3.4 ASPeCT level 4 : Full UMTS system, evolved from GSM (Phase 2+)

This is the target network level in which the UMTS air interfaces are ubiquitous and the fixed network infrastructure has been upgraded universally to Broadband ISDN. All UMTS services (including broadband services up to 2Mbits/s access in some environments) are available to UMTS users.
Each transition phase from one network level to another involves upgrade of one or more entities (the smart card, terminal equipment, the access network and the core network) as detailed in D05 [18]. We examine the effect upon the smart card in the next subsection.

4.2.4 Impact on smart card

In this subsection, we discuss some general considerations concerning smart cards in relation to the evolution from second to third generation systems. Following this, we list the types of smart cards, both single and multi-application, which are envisaged to be in circulation at each of the four levels of the ASPeCT migration scenario. Note that, in this subsection at least, we only consider telecomms applications when classifying smart cards as single or multi application; thus single application cards as defined here may contain other non-telecomms applications in addition to the telecomms application.

General considerations underlying the evolution of smart cards in this context include the following:

4.2.4.1 Ubiquitous Service Provision & Backward Compatibility

Until the full UMTS network is widely available, there is unlikely to be a single application smart card containing only a UMTS application. UMTS users will need other telecomms applications on their smart cards if they are to obtain ubiquitous service in areas where UMTS coverage is not available. In addition, even if UMTS coverage is available, the user may possess terminal equipment which does not contain UMTS functionality, and will need to access the network via a non-UMTS air interface. These closely linked concepts of ubiquitous service provision and backward compatibility are clearly central to the evolution of the smart card.

4.2.4.2 Development of UMTS application

As the third generation systems are progressively deployed, the service and feature capabilities that they afford to UMTS users will gradually increase. Thus, in addition to the initial evolutionary step of including a UMTS application on the smart card, it is likely that the UMTS application itself will evolve to match the prevailing service and feature capabilities of the underlying UMTS network. Hence, the UMTS applications of the first evolutionary smart cards are likely to be quite basic in comparison to those of the later evolutionary smart cards.
Bearing in mind the general comments made above, we now list the types of smart card which are envisaged to be in circulation at each of the four levels of the ASPeCT migration scenario [18].

4.2.4.3 ASPeCT level 1 : Initial Network Level

The envisaged smart card types are:

Single application:

- GSM900
- DCS1800
- DECT

Multi-application:

- GSM900/DCS1800/DECT or any multi-application subset thereof.

4.2.4.4 ASPeCT level 2 : Introduction of UMTS Users

The envisaged smart card types are:

Single application:

- GSM900
- DCS1800
- DECT

Multi-application:

- GSM900/DCS1800/DECT/UMTS or any multi-application subset thereof.

4.2.4.5 ASPeCT level 3 : Introduction of UMTS Air-interface(s)

The envisaged smart card types are:

Single application:

- GSM900
- DCS1800
- DECT

Multi-application:

- GSM900/DCS1800/DECT/UMTS or any multi-application subset thereof.

4.2.4.6 ASPeCT level 4 : Full UMTS system, evolved from GSM (Phase 2+)

The envisaged smart card types are:

Single application:

- GSM900
- DCS1800
- DECT
- UMTS

Multi-application:

- GSM900/DCS1800/DECT/UMTS or any multi-application subset thereof.

4.3 Enhanced service concepts

In subsection 4.2.2, it was stated that, for ASPeCT level 2 (Introduction of UMTS users), it may be possible to emulate certain UMTS specific services and features for UMTS users, even in the absence of a dedicated UMTS air interface and fixed network infrastructure. In this section, we describe briefly how such emulation may take place.

For ASPeCT level 2, the introduction of UMTS users is achieved by upgrading the network SCP to include UMTS functionality and issuing UMTS uses with a multi application smart card containing a UMTS application. Between the user and SCP, the intervening network infrastructure, air interface(s) and terminal equipment remain unchanged i.e. they are still based on conventional GSM900/DCS1800/DECT technology.

Hence, emulation of UMTS specific services and features for UMTS users requires *direct transparent* communication between the UMTS SCP and the UMTS application of the smart card, as these are the only entities which possess UMTS functionality. The intervening network entities and the user's terminal equipment must then have the functionality to pass messages and/or data transparently. This is important because it is only with the advent of GSM Phase 2+ and in particular the SIM Application Toolkit that GSM terminal equipment will have the capability for transparent data transfer. Previous generation GSM terminal equipment does not have this capability i.e. the terminal equipment must attempt to interpret all messages received from the smart card or network as appropriate.

Let us suppose in the ASPeCT level 2 scenario that a UMTS user is using a piece of GSM900 terminal equipment which supports transparent data transfer. On registration (turn on of terminal equipment or insertion of smart card), the GSM application must be accessed on the smart card and the GSM identity of the subscriber (i.e. the IMSI) sent to the network. At this point the UMTS SCP realises that this identity corresponds to a UMTS subscriber can thereafter (in principle at least) communicate with the UMTS application of the card using transparent data transfer. No further correspondence with the GSM application is required. This implies that a multi application smart card for UMTS users in ASPeCT level 2 need contain a UMTS application and only 'dummy' GSM900, DCS1800 and/or DECT applications. However, in practice, 'full' GSM900, DCS1800 and/or DECT applications are likely to be included as these will be required if the terminal equipment used does not support transparent data transfer.

Examples of UMTS specific services and features which may possibly be emulated for UMTS users are:

- UMTS mutual authentication between network operator and smart card
- certain UMTS specific teleservices (e.g. the UMTS equivalent of the GSM short message service)

In any case, only a limited set of UMTS services and features can be offered to UMTS users at this level. In particular, high bit rate and/or multimedia services will not be available as these require a suitable UMTS air interface.

4.4 Over-the-air value added services

4.4.1 Introduction

In GSM the SIM card plays a key role in Value Added Service (VAS) systems which offer information services to the subscriber based on OTA. The inherent security and information storage features of a SIM make it an excellent partner for VAS Centres.

In GSM, in the absence of Phase 2+ SIM Application Toolkit functionality, OTA concepts are based on the Interpreted Short Message Service which is used to communicate *Over-the-air* between the VAS-Centre and the SIM. The OTA solution is ideally independent of the mobile network (GSM/DCS/PCS), the SMS-Centre and the handsets used – thus offering maximum flexibility. Unlike

the standardised mechanism for OTA defined within the SIM Application Toolkit, the details of this interim solution based upon I-SMS do depend on the card manufacturer. The VAS application need only be implemented on a VAS platform and on the SIM therefore allowing the mobile network to be treated as a transparent medium for messages.
A typical OTA solution supports SIM file management from several remote VAS platforms and includes a suite of security functions, making it ideal for a diverse range of VAS applications.

4.4.2 Standard SMS

Standard SMS (S-SMS) allows text messages to be sent to and from handsets. S-SMS has some inherent restrictions for information transfer:

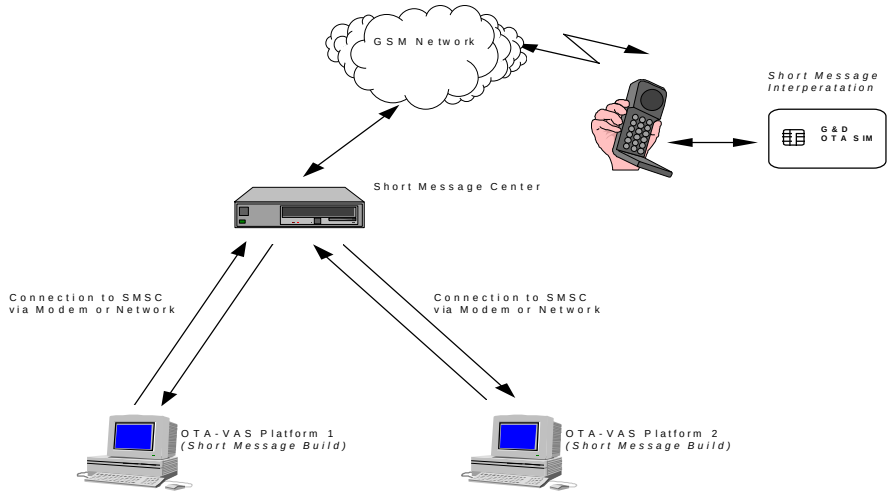
- No automatic verification of sender (someone could pretend to be the subscriber's bank and send him misleading or even dangerous information)
- SMS storage area blockage (every SMS has to be manually deleted)
- The user interface is often not very user-friendly

4.4.3 Interpreted SMS

Interpreted SMS (I-SMS) is based on S-SMS, but does not have the above listed restrictions. Security services for confidentiality and sender verification are included. No administration for subscriber messages (e.g. deletion) is required and a wider choice of user interfaces is available.
Basically, interpreted SMS provides the ground for a lot of concepts which are envisaged in third generation telecommunication systems. These are, for example, the following items:

- remote creation of new applications on SIM cards
- reallocation of prepaid memory areas
- subscriber specific adaptation of services
- dynamic management of keys and security related information

One of the most promising alternatives for implementing I-SMS is the Point-to-Point SMS. This service allows a Short-Message-Entity (SME) to send a message to any user via a short message centre (SC). The SME may be a value added service provider or an integral part of a GSM PLMN.



Different factors will influence the use of the I-SMS concept:

- the maximum length of messages
- the expressive power of the languages employed in I-SMS commands
- how much processing and storage power is available on the SIM
- the protocols which guarantee a timely delivery of messages in case of an absent mobile station

Some of these items are currently under study in ETSI, for example the possibility to concatenate standard short messages. This feature would allow to split up longer messages into smaller pieces, sent them in an arbitrary order to the SIM and reassemble them to the original one. Envisaged message lengths are 255*134 bytes – this reflects the fact that S-SMS limits the amount of user data

to 140 octets for one short message, that there is some overhead for managing the data and that today's SIMs can store at most 255 records in one elementary file at the same time.
How much complexity is necessary in the I-SMS depends on the concepts that will be supported. It must be kept at a minimum due to the negative effect on memory requirements. A very powerful interpreter is helpful from a theoretical point of view, but can hardly be implemented in software. Alternatives are hardwired solutions, but these increase the costs and reduce the flexibility of the chip.
Another important issue is the possibility of an absent mobile station. A user may turn of his handset to save his battery or remove the SIM. In such a case, the GSM network will inform the originating short message centre and alert it when the mobile station becomes alive sometime later. The SC will retry once again if the message content is still valid. Independently of this alert mechanisms, every short message centre may periodically repeat an unsuccessful transmission attempt.

4.4.4 Applications of Interpreted SMS

The following are examples of possible applications based on OTA:

- Banking Transaction Information
- Share Price Information
- Service Telephone number download
- Advanced Directory Enquiries
- Virtual Prepaid SIM
- Loyalty Bonus Balance

Banking Transaction Information – OTA can be used to periodically transfer bank balances to reserved storage areas in the SIM, which can be displayed by the subscriber at any time. Notification of payment transactions may also be carried out.
Share Price Information – Share price information or currency exchange rates can be periodically sent to the SIM. Thus, the subscriber always has the current value at his fingertips.
Service Telephone Numbers – Telephone numbers can be downloaded to SIMs via OTA. Thus, service numbers of, for example, a Travel Agent or Florist can be downloaded to subscriber groups based on agreement between the network operator and the service agent.
Advanced Directory Enquiries – In addition to providing information via standard voice directory enquiry, a VAS centre can download the requested number as data to a predefined phone number location to facilitate dialling by the subscriber.
Virtual Prepaid SIM – To complement a network/billing system based SIM concept, the balance of the prepaid account can be downloaded to the SIM where it is constantly available to the subscriber. For security and system recovery reasons, the actual balance should be stored in the network and a copy of the balance sent to the subscriber as required.
Loyalty Bonus Balance – A cellular loyalty scheme in which preferred customers are credited with cellular minutes may be introduced. In this case, the balance, similar to the Virtual Prepaid SIM balance, may be sent to a fixed location in the Subscriber's SIM.

4.4.5 File management features

The following file management features will be possible through OTA:

- allocation of prepaid memory amounts to customers
- support tools for remote application development
- access from several independent VAS centres

Allocation of prepaid memory amounts to customers – From a service provider's point of view, memory cells are some kind of resource which is limited and therefore valuable. A VAS provider may offer a lot of applications or even decide to drop some of them he used to support but which are not profitable any more. One possibility to make such a dynamic application management possible is to sell bytes in a chip to VAS providers. A provider that bought a memory area on a SIM is granted the right to subsequently load data or software in this area without the help of a card manufacturer. Such a trading with memory areas requires support from hardware units which prevent VAS service providers from accessing memory cells they did not pay for.
Another possibility is to grant the right to buy memory sometime later. In this case, the manufacturer of the SIM or some substitute has to be asked for final allocation of cells. The winner is then who pays first for it.
Support tools for remote application development – The largest amount of memory is useless if a service provider cannot use it. Provided that a SIM supports I-SMS, descriptions of file structures or

any other way of organising data can be packed in a short message and be sent to the SIM which interprets the message. Depending on the expressive power of the command language, anything which is not even imaginable today may be remotely developed in the SIM. Such a process should be interactive and involve the user. Of course, this can only take place if the SIM is inserted in an active handset. The main advantage of this concept is that a service can be customised whenever a user instructs his VAS provider to do so.
Access from several independent VAS centres – In a growing service market, different providers will offer a wide variety of service products. This is desirable from a user's point of view, but makes things more complicated in the SIM. Security mechanisms must ensure that only those entities have access to data stored in a SIM which are entitled to do so. This requirement is much stronger than today's philosophy of PINs and PUKs. More enhanced methods for access control have to be used..

4.4.6 User interface

From a user's point of view, two different scenarios have to be considered:
Standard GSM Handsets – In this case, the value added service is to be offered to subscribers with standard handsets, with no special features. This offers maximum flexibility for distribution of the VAS to subscribers. However, the user interface must be an existing standardised one, such as ADN, The subscriber can thus scan the register for an appropriate entry such as "ACCOUNT:DM1000".
Customised GSM handsets – It may be required to have a specially developed handset, with a customised user interface. For example, a special button on the handset would call up the bank balance onto the screen.

4.5 The SIM Application Toolkit

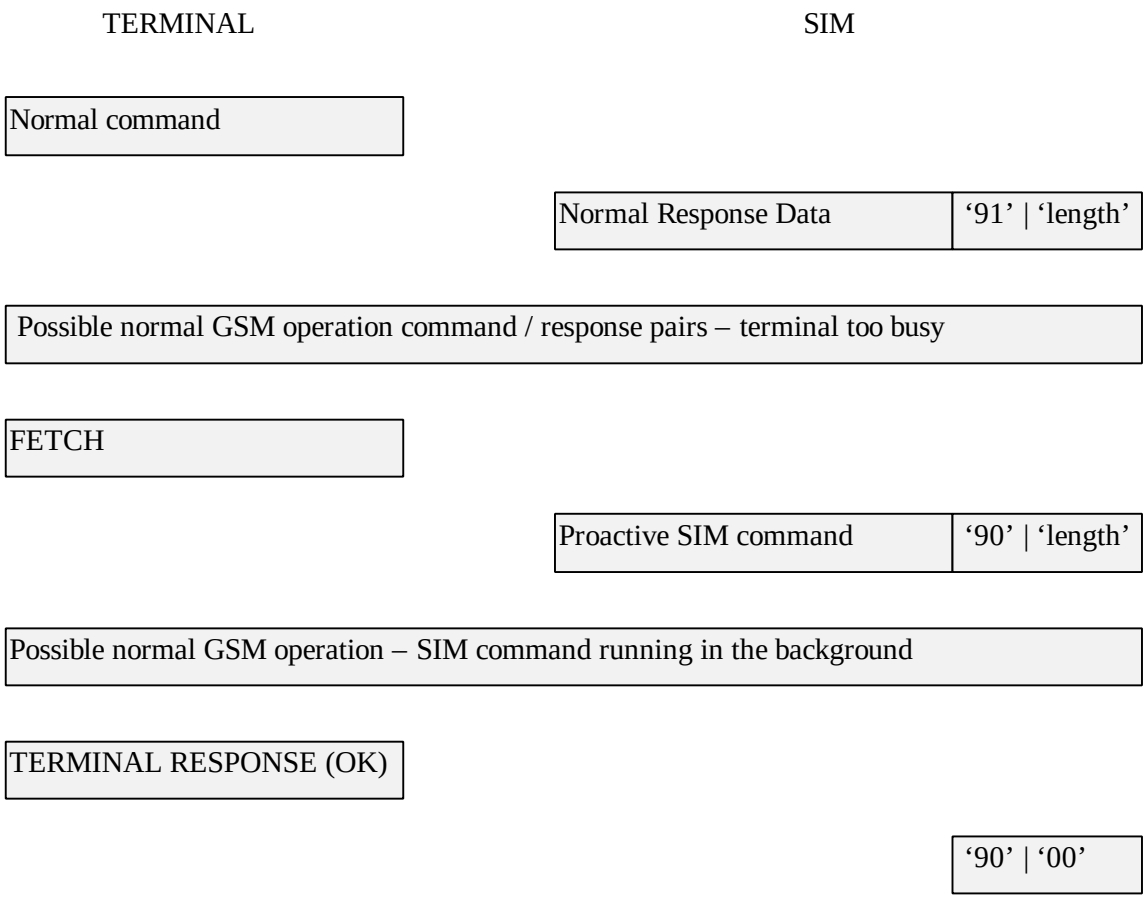
4.5.1 Introduction

In the previous chapter, the interpreted Short Message Service has been identified as a mechanism to implement the OTA concept. As it stands today, S-SMS may transport messages from an SME to an ME, the display of an ME or a SIM. However, short messages stored in the SIM can only be accessed for information by the user; they cannot be used for other more powerful tasks such as updating of records on the card. The SIM Application Toolkit facilitates such powerful functionality using a new transparent link between the network and SIM involving SMS.
In the following subsections, an overview of the most important concepts will be given. Anything therein should be seen in an OTA context.

4.5.2 Proactive SIM commands

The proactive SIM service provides a mechanism which allows the SIM to say to the terminal "*I have some information to send to you*". According to ISO/IEC 7816-3, this should be impossible because the terminal is always the master and initiates commands to the SIM. ETSI is currently developing a specification [4] to support this proactive SIM, this specification has not yet been ratified but is based on the following two ideas:

- *Proactive polling*: the SIM can negotiate the maximum time interval between STATUS commands issued by the terminal when in idle mode. This ensures that the SIM will have at least one chance to send a status response word to the terminal between any two subsequent points in time which differ more than the defined maximum polling interval.
- *Special status response words*: the proactive SIM service stays within the T=0 protocol, but adds a new status response word SW1 ('91'). This status response has the same meaning as the normal ending ('90 00') and can be used with most of the commands that allow a normal ending. The difference is that the terminal learns that the SIM has some information for it and uses the FETCH function (a new command) to find out what this information is. When the proactive command has been processed by the terminal, the SIM is informed through TERMINAL RESPONSE. The following figure illustrates an example:



The SIM can issue a variety of commands through this mechanism. The list encompasses

- dialogues between user/terminal and SIM via the display/keypad
- sending a short message/service request from the SIM to the network
- allowing the SIM to set up a call without user intervention

The next subsections will look at them in turn.

4.5.3 SIM originated short messages

A short message to be sent to the network can be generated by the SIM and be passed transparently to a Short Message Entity via the mobile terminal. The means to do this is the SMS-SUBMIT message type in the SMS Point-to-Point service.
Common mobile originated short messages have to be typed in by the user via the keypad of the terminal. This is sufficient for communication between human users, but it is not appropriate for OTA. A SIM must be able to sent its own mobile originated messages without user intervention.
With the SIM Application Toolkit, this becomes possible. First, the SIM generates the data it wants to sent. This data is then embedded into a SMS TPDU together with addressing information that identifies the recipient SME and the originating MSISDN. Finally, the whole message is transparently transported from the SIM to the mobile terminal through the proactive SEND SHORT MESSAGE command.
From now on, the SMS TPDU is treated as a user originated one. The terminal sends it to the network which delivers it to the destination SME. Depending on the implementation, the user of the mobile terminal may be noticed of the message by displaying it.
This proactive command is currently the only way for a SIM to send a message to a SME whenever it needs to do so without asking the user to type in the message manually.

4.5.4 Data download to SIM

A non-proactive mechanism which is embedded in an ordinary pair of command and response is SMS-PP data download. As usual, data that is to be sent from a SME to a SIM is packed in a SMS-DELIVER TPDU and transparently transported to the terminal through the SMS-PP service.

The SIM Toolkit command ENVELOPE allows the terminal to send the user data part of the message to the SIM without automatically storing it in the SMS table. The terminal may not even display such a message as long as it is appropriately marked. If the recipient mobile station should be absent, the network will retry later to deliver it.
This mechanism allows a SME to send command or control messages to a SIM which are not displayed or stored in the SMS table and are therefore not subject to memory capacity exceeded restriction in the SIM.

4.5.5 The more time mechanism

Another feature of the SIM Toolkit which looks quite interesting is the possibility of the SIM to prevent the terminal from stopping the clock when a predefined number (currently 1860) of clock cycles has elapsed after the last byte was received from the SIM. Such an automatic clock stop is dangerous for time consuming computations running in the SIM.
Applications which do not know in advance how long it will take to finish a computation that is running in the SIM can send the proactive MORE TIME command to the terminal. This prevents the normal stopping of the clock and allows the terminal to reset its clock cycle counter. When this counter has elapsed, the SIM clock will be stopped as usual. If the SIM needs even more time, it has to repeat the MORE TIME command whenever the terminal is active. If the card has to comply with a deadline date for the result of a computation, the SIM may prevent the terminal from changing its state to stand-by mode and enforce it to wait for the SIM-computation to finish.

4.5.6 Call control by the SIM

The idea which underlies the call control service is that every call attempt or supplementary service registration that includes a destination number is first passed to the SIM. The call set-up attempt may be declined if the SIM cannot accept international calls for example.
The advantage of this service against existing ones is that the SIM may also modify call related data. A user may type in the number of his family he is used to and expect that a call is set up. He is probably not interested in operator dependent modifications, for example country codes.
Before the call is set up, the SIM uses the standard number typed in by the user as a row index in a translation table which is stored in the card and replaces the standard number by a modified number which respects the current network context. In combination with OTA, a VAS provider can update the translation table in the SIM depending on the user's location.

4.5.7 The Toolkit as a platform for customised services

The Toolkit as it stands will close the gap between expectations of network operators, users and card manufacturers.

- A user wants his mobile station to be as simple as possible, but powerful enough to fulfil all the needs users may have. This requires the SIM to support the terminal in offering a simple MMI that nevertheless allows a user to define his personalised set of services.
- A network operator wants to have direct access to the SIM. The list of features covers file management, service adaptation, key management or even remote application development. This is not possible if the card has to be reprogrammed by the manufacturer every time a single bit changes.
- Card manufacturers are given the possibility to offer highly specialised system solutions which require more than standard operating systems for file management. The long term goal of a generic *chameleon* card which changes its appearance depending on the user's current desires is not only a challenge, but also an opportunity to enter new markets.

4.6 Requirements on the functionality of a UIM

4.6.1 Introduction

Independent of a particular migration path, some requirements on the functionality of a UIM can be identified today. These are in particular security services which are general enough to be used as the basis for later specifications. They do not anticipate things to be standardised later. It is the goal of this section to take a look at concepts which are not fully supported by today's SIMs but should be supported by future UIMs.

4.6.2 Security services

4.6.2.1 Introduction

In the section on Over-the-Air value added services, a framework was given which defines the communication structure and the entities involved. Depending on the point of view, the security services a UIM has to support are different. A user may have different expectations than a network operator and vice versa. Nevertheless, the techniques used to implement security stay the same – they are independent of the goals in mind.
Due to the dynamic nature of the field and the changing landscape of telecommunication systems, it is necessary to identify basic *requirements* which reflect expectations of the parties involved and which are inherently stable. This is the major goal of this subsection on security services.
First, mechanisms for access control to data or software stored in a UIM will be considered. In the subsequent paragraph on authentication capabilities, the need for a management of authentication algorithms is demonstrated and a practical solution based on Authentication Capability Classes is proposed.
Another important requirement is the ability of a UIM to offer non-repudiation services to all the parties involved. This covers logging mechanisms and signatures for user originated requests. The last paragraph on key management classifies the different types of keys which have to be stored in a UIM and identifies some mechanisms for their generation and distribution.

4.6.2.2 Mechanisms for access control

Today's philosophy for access control in the SIM world is known as *Personal Identification number*. The presentation of this magic number is enough to open the door to any data element which is classified as PIN-protected. To introduce different classes of protection, several PINs are used. After three consecutive false attempts, the PIN is invalidated and the *Personal Unblocking Key* plays the role of the PIN. This method may be iterated.
From a user's point of view, this is nice as long as the PIN can be changed by the user and only one application resides on the SIM. With a multi-application card, the situation changes.
A SIM that supports VISA and GSM is radically different. For example, a network operator who updates a service on the user's behalf may change a bank balance stored on the card without being authorised to do so. This is a new quality of threat which makes PINs and PUKs obsolete – much stronger mechanisms to enforce access control are in place here.
Basically, the trust relationships in a secure system are reflected by the location and usage of keys. If a user shares a long term key with a VAS provider and can influence the generation of session keys, he can be more confident in the security of the system.
The key concept for access control is *message authentication*. A command message sent from a VAS provider to the SIM has to be authenticated. How this is done depends on the computation power of the SIM as well as on the framework for key management. Two different mechanisms are in place here:

1. A service provider holds a master key MK which is common for all his subscribers and is stored in the SIM. Together with the ICC number of the SIM, this key is used to generate an individual key for each user : IK(U). If this VAS provider wants to send a message, he authenticates himself to the user and employs IK(U) to set up a session key SK(U). This session key is distributed to the user during the authentication procedure. Together with SK(U) and a symmetric encipherment algorithm, it is easy to enforce integrity for messages sent to the SIM: the provider computes a MAC for the data part of the message and appends it at the end. The SIM can detect any change during the transmission through this MAC and rely on its authenticity.
2. A different approach employs asymmetric cryptosystems: the SIM stores the public part P of a service provider's signature key. This public key is semi-static – it need not be redistributed for each authentication procedure. If a message M is received by the SIM that claims to be have been sent by a VAS provider, the SIM uses a cryptographic hash-function h to compute h(M). If M is authentic, it has the signed value SIGN(h(M)) as an appendix. The SIM can verify the appendix with the public key P and indeed be sure that M was signed and therefore sent by the service provider it claims to come from.

Both schemes have in common that the originator of the message appends some value that establishes authenticity. Nevertheless, if the keys used to generate this appendix are compromised, the security of the whole system is lost.

From a more general point of view, access to data stored in the SIM is granted to everyone who can prove that he knows some keys. This shows that there is a strong relationship with the key management framework.
Currently, an access condition is defined for each file. The way the access levels are handled in a GSM SIM is illustrated by the following example: ADM refers to administrative management.

ACCESS CONDITIONS	EF_{IMSI}
Operation	Level
READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	PIN

The major disadvantage of this scheme is that it mixes user identification and access control. The user has to type in the correct PIN manually. In case of a success, the SIM is lifted to the PIN-level and the terminal may perform a READ command on the EF_{IMSI} without further authentication.
A more detailed access control mechanism should define which keys have to be used for *over-the-air* access commands. This concept is entirely independent from user identification and may coexist with the current PIN-based solution.
Example: the file EF_{TRANS} contains a translation table for phone numbers similar to the abbreviated dialling numbers service. This table is periodically updated over-the-air and maps short numbers to international numbers. Depending on the user's location, different updates will take place. To keep things simple, only the VAS provider has the right to update entries in this table. The operator of the visited network may read them, but is not allowed to change anything. Provided that the user already shares keys $K_{U,NO}$ and $K_{U,SP}$ with the network operator and service provider respectively, an access structure for EF_{TRANS} looks like follows:

ACCESS STRUCTURE	EF_{TRANS}	
Command	Entity	Key
READ	Network Operator	$K_{U,NO}$
	Service Provider	$K_{U,SP}$
UPDATE	Network Operator	NOT ALLOWED
	Service Provider	$K_{U,SP}$

It should even be possible to change access structures for selected files. This can be done with the aid of the card operating system. Any *over-the-air* access attempt has to pass an OTA-access-module that compares commands and their respective MACs or signed hash appendices with the expected ones. The access structures are a part of this module. Nevertheless, the OTA-access-module is subject to its own access structure which defines who has the right to change which access condition tables. This scheme may be iterated, but at some point, it has to stop with a static table that cannot be changed.

4.6.2.3 Authentication capabilities

In contrast to today's authentication algorithm A3 which is proprietary and remains unchanged for a card's life, future UIMs have to cope with a lot of different algorithms. Network operators may agree on *authentication capability classes* which are standardised and encompass existing ones.
At the beginning of a registration attempt, a user sends an initial authentication request to the operator of the visited network. This includes the authentication capability class of the UIM. Based on information from the user's service provider, the operator sends the description of a prescribed authentication mechanism back to the user that lies in this class. For the current registration, both sides adhere to the selected mechanism.
This leads to the question what such authentication capability classes (ACCL) may look like. Basically, they will consist of a set of commands and their responses – from a card's point of view. The number and complexity of such pairs will depend on the class. As it stands, the GSM authentication mechanisms works as follows: the terminal sends some random number RAND to the SIM through the RUN GSM command and receives a value SRES from the SIM. This number is sent back to the network and compared with the one expected. The sequence of actions between terminal and SIM is a one-step process.

New authentication mechanisms will require two or three-step interactions. The following example illustrates how these new concepts can be integrated into existing standards and has to be seen in the context of the ASPeCT authentication framework, as described in the ASPeCT deliverable D05 [22].
Example : *The authentication starts with the UIM sending some data to the network that interprets this data depending on the history (current or new registration). Really new about this is the fact that the CARD initiates the authentication procedure. There may be different reasons for this action:*

- *the UIM wants to establish a new temporary identity with the network operator or wants to register for a new network*
- *the UIM received a request for authentication from the network operator or the service provider*

In the first case, it is up to the UIM to inform the terminal. The means to do this is probably some proactive SIM command that requests the terminal to fetch the data from the card and sent it to the network for interpretation. In the second case, an ordinary pair of command and response is sufficient.
After this first step, the details of the ACCL will be determined by the network operator who informs the UIM. This requires some data download mechanism to transport the selected ACCL to the UIM and the acknowledgement back to the network.
The rest will consist of some more pairs of proactive or non-proactive commands that are necessary to perform the actual user-network authentication procedure. For example, this may be an ordinary one-step RUN GSM command.
From a more theoretical point of view, authentication capability classes depend on algorithms implemented in the card as well. The UIM must be able to inform the network on its capabilities. As said before, every UIM is allocated some ACCL value. This value may be a simple counter, but it could also be a text string describing what the card is capable of. Due to the ongoing discussion on this topic, the authentication capability class of a UIM should be subject to OTA reallocation.

4.6.2.4 Non-repudiation services

From a user's point of view, the OTA feature offers an easy way to adapt the UIM to current needs. This is nice as long as nothing goes wrong. In case of a quarrel, the situation for the user looks quite bad. He or she can say that some money had been stolen from the UIM, but it is nearly impossible to prove that fact in court. The mechanisms which may resolve this asymmetry between users and service providers are the subject of this paragraph. At the end, we will look at undeniable service requests.
In the paragraph on access control, two alternatives were given which can implement message authentication. The first one was based on symmetric algorithms and appended MACs, the second one relied on digital signatures. The great disadvantage of the MAC based method is that it cannot easily be extended to offer the *non-repudiation of origin* service to the user.
Any access attempt and in particular any reconfiguration in the UIM which was done over-the-air represents some kind of danger. For that reason, it is easy to see that OTA commands should be stored in a special *logging file*. Entries in that file can be read out and be used to prove the existence of the respective commands.
A simple proof of existence does not help as long as there is no way to find out the state of the operating system at the time of execution. This state should be coded (or at least the relevant parameters) and stored together with the OTA command. Last but not least, the time of execution and some additional information may also be appended.
This data string describes in some detail what happened in the UIM some time in the past. It does neither prove that a command was actually sent by a service provider at this particular point in time nor that the command was indeed executed in the card by the operating system. Both goals can be satisfied with digital signatures. The question is which signature keys are appropriate.
If the message authentication mechanism for access control is based on asymmetric cryptosystems, everything is fine. But if the UIM does not support signatures, the situation is more difficult. The keys used for MAC generation are secret and only known to the user (or his UIM) and the service provider. Without a third party that also knows the key, a user cannot prove that a command was sent by a service provider because he (the user) can do anything the VAS provider can and vice versa.
This demonstrates once more the urgent need to implement a signature generation and verification scheme in the UIM.

From a service provider's point of view, the situation is in a sense symmetric. A user can request a service through a service control string that is sent to a service provider or a network operator. The problem is that this string is not an undeniable proof of request. If the UIM already supports signatures, such service requests may be signed by the user and the signed request can be sent to a Short Message Entity that stores them. This gives a provider the same confidence in the trustworthiness of the system as the user.

4.6.2.5 Key management

This last paragraph will first define criteria for classifying keys which are relevant for the UIM and will follow the line

- Intended use
- Responsibilities
- Validity

Based on this scheme, command structures for the generation and distribution of keys will be considered.

Intended Use – The way a key is used depends both on the algorithm and the protocol. This is obvious for encipherment or decipherment, but anonymity algorithms, key-agreement keys, cipher-key generators, algorithms to compute authentication tokens or simply signature verification or generation schemes are also important. A UIM will probably support all of them.

Responsibilities – The question which entity allocates or invalidates a key depends on the protocol. The UIM may generate different types of keys if this is necessary. It may also store keys allocated at (pre)(re)personalisation or even keys which are stored over-the-air.

Validity – Explicit expiry dates are usually only applicable for public keys embedded in certificates. A UIM should be able to

- present a list of certification authorities (CA) the signatures of which it can verify
- present a list of CAs which have issued certificates for some public key of the UIM
- store certificates for public keys of the user or some other entity and present a description of such certificates

A UIM may also assist in the generation and distribution of keys. On request, it may produce some key pair and either store it or send it to the terminal which distributes it to the requesting entity. Such keys have to be encrypted before they are delivered. The details of such requests should be standardised and describe the desired properties of the key.

4.6.3 Operating system features

4.6.3.1 Introduction

The operating system is of course the place where a large part of the security services a UIM will support are implemented. Only some very basic operations may be available in hardware. In spite of the memory constraints which limit the possibilities in this area, this subsection tries to outline what an operating system for a third generation card (UIM) should offer.

4.6.3.2 Cryptographic modules

The following subparagraphs identify modules which have to be implemented in a UIM.

Finite abelian groups

A basic tool which is necessary for nearly all good cryptosystems is a finite group in which the discrete logarithm problem $a^x=b$ is hard. Typical candidates are

- the multiplicative group of a prime field: GF(p)
- the multiplicative group of a finite field of characteristic 2: $GF(2^n)$
- elliptic curve groups over finite fields F: EC(F)

It is very important to have in mind on which prerequisites the security (the difficulty of solving the DL-problem) of a particular group relies. Some groups are secure due to extensive precomputations which have to be done once per field. If the validity period of a prime is over, it has to be deleted and substituted by a new one.

For that reason a card has to support parameterised groups. The specific parameters which define the group are changed over-the-air. In critical applications, the card may even invalidate a group which is evaluated as insecure by the operating system and so enforce the replacement.

One-way hash functions

Another building block of many protocols are one-way hash functions. From a card's point of view, the difficulty here is to satisfy competing demands. The hash-length should be as short as possible to save memory, but this invites different types of attacks.
Generally, the requirements on a one-way hash function h are:

- given a message M, it is easy to compute h(M)
- given some value h´, it is hard to find M such that h(M)=h´
- given M, it is hard to find another message M´ such that h(M)=h(M´)
- it is hard to find two random messages M and M´ such that h(M)=h(M´)

The likely candidates are SHA-1 [19], one of the Ripe-MD variants [20] and constructions based on block ciphers.

Message Authentication Codes

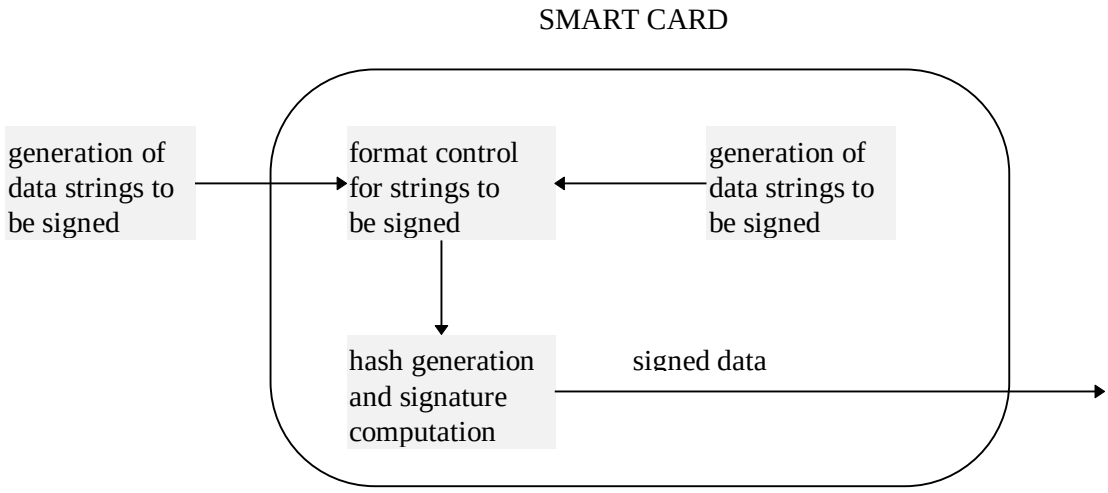
A message authentication code (or MAC) is a key-dependent one-way hash function. Only someone with an identical key can verify the hash. If a card already supports one-way hash functions, they can be taken for granted.
In practise, MACs are used to provide authenticity without secrecy. A message that is to be authenticated either by its sender or its receiver is used as input to the hash function together with the secret key. The output is appended to the message. In the context of UIM migration, MACs are a central tool for message authentication which is a prerequisite for access control. The way they are implemented will heavily influence how flexible the OTA feature will be. Efficiency is here at least as important as security.

Encipherment and decipherment

To achieve secrecy, symmetric algorithms should be used. From an efficiency point of view, block ciphers are better suited than stream ciphers – they will probably be implemented anyway in the card. If security for the next decades is a concern, the choice of algorithm becomes harder. Triple-DES is good, but not very fast. Faster alternatives like IDEA should be taken into account.
For the UIM, symmetric algorithms are not really used to achieve secrecy. The only application which requires the card to encipher messages is secure messaging (transfer of data between the terminal and the UIM). As long as this feature is not supported, they are employed to generate MACs.

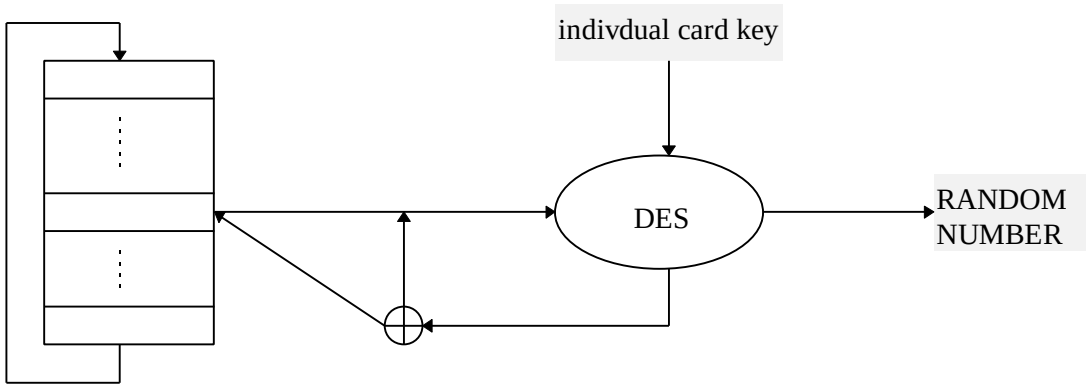
Signature systems

For signatures, the situation looks similar. It is necessary to have modules for the generation as well as the verification of signatures in the card. Computational efficiency is always a limiting factor, but this is not so crucial here because signatures will be used to sign hash values which are approximately 160 bits long. Therefore security considerations are more important here. Popular candidates are RSA, DSA or elliptic curve based systems.
The question what should be signed is more complicated. Generally speaking, a command which sends some data to the card, tells the number of some key to use and expects the card to send back the signed sample is dangerous. This allows a pure chosen plaintext attack.
A better idea is to look in more detail at the use of signatures. They will probably be employed to sign service requests or OTA access commands. In both cases, the data strings to be signed adhere to well defined coding rules or patterns. The following picture illustrates an architecture which defines the framework:



Random number generation

There are lot of different algorithms in the field for the generation of pseudo-random numbers. In this subparagraph, a solution based on DES will be outlined. It is desirable because it uses modules which are already present in the card.
Basically, the random number generator (RNG) consists of a cyclic buffer for 8 byte values, a binary adder and a DES encryption module. The buffer is shifted after each step of the algorithm and stores the new random number. In the following picture, the algorithm is illustrated:



The key used for the DES encryption as well as the initial values for the buffer are stored during the initialisation phase of the card. It may be possible to generate the keys at any point in time based on a RESET_RANDOM command if the card supports appropriate key generators. The implementation of such generators should take into account the keys needed for random number generation.

Key-generators

A quite complicated task is the generation of *good* cryptographic keys. What good means depends heavily on the algorithms used. As a rule of thumb, some very basic modules are expected to be necessary:

- probabilistic methods for primality testing
- methods to guess prime roots mod p
- generation of random bit strings
- heuristic rules for specific algorithms

The distribution of secret keys outside the card should not be permitted. It makes sense to introduce a command which says "*Generate a new RSA key for your signatures and tell me your public key*". This improves the security of the UIM. But it is not a good idea to allow a terminal to requests pairs of private and public keys for use outside the card. This is simply bad practice.

4.6.3.3 Multi-tasking

Currently, the terminal can issue one command at time and has to wait for the response from the SIM for the one issued last. With at later phase of the SIM Application Toolkit, the situation may be different. Through command numbers, the SIM may be able to issue several commands one after another without having to wait for the associated TERMINAL RESPONSE. The current specification of the SIM Application Toolkit is being designed to be compatible with such an eventuality.
A more general approach could be to introduce multi-tasking concepts in the UIM. A process could be created together with a security context for each ISO-7816 command received from the terminal and similarly one process for each proactive command that is not finished yet.
The difficulty with this idea is not only that memory technology prohibits such elaborated mechanisms. It poses some logical difficulties too. Working on terminal commands in parallel is only possible if these commands are independent of each other. If one commands uses the output of its predecessor, the terminal has to wait anyway.
Nevertheless, if the number of processes is limited in advance and resources in the card are shared between different applications, it may be possible to implement multi-tasking up to some extent.
The ISO concept of logical channels addresses this topic to a certain extent and is considered in Section 5.3.5

4.7 A preliminary view of UIM migration

4.7.1 Introduction

This section tries to identify a preliminary migration path for the User Identity Module. It looks at the SIM as a possible starting point and envisages a way up to a multi-functional telecommunications card.

4.7.2 The SIM as a starting point

As it stands, the SIM is a single-application processor card that offers the basic commands according to ISO/IEC 7816-4 and stores user or service related data in two independent directories which are called DF_{GSM} and $DF_{TELECOM}$.
The system migration scenarios which are considered identify the GSM Phase 2+ network as the initial network level. This leads to the assumption that the SIM is an appropriate starting point for card migration. The more interesting question is how the SIM can be developed to fulfil the requirements on a UIM step-by-step.

4.7.3 Migration of services and security

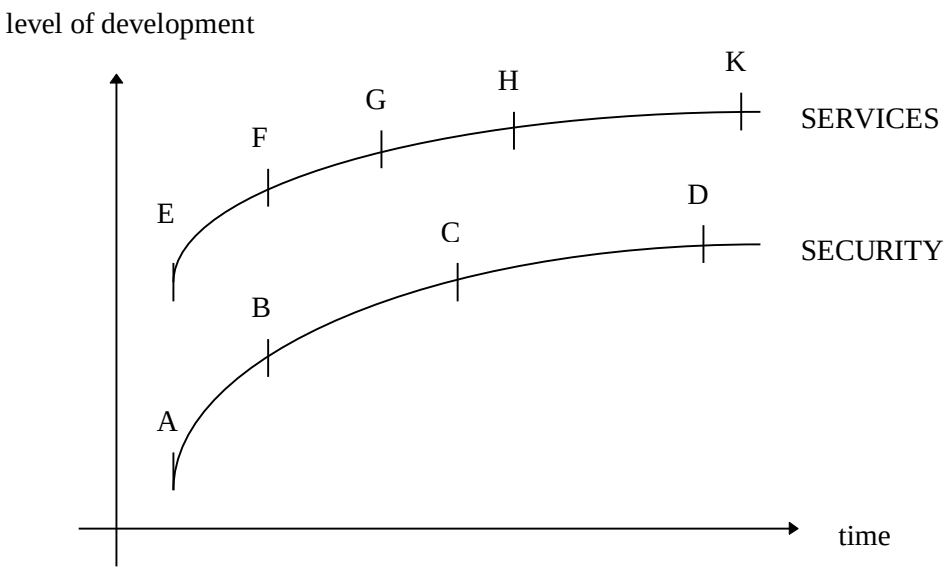
One of the major characteristics of a UIM is that is a multi-functional card :

- it contains several telecom applications: GSM900/DCS1800 and DECT
- it offers payment functionality and value added information services
- the security functionality of the card is enhanced

To guarantee a soft migration requires that the migration steps should be small. It is also necessary to look at the migration path from different perspectives:

- the service perspective: which services are offered on what level
- the security perspective: which security services are supported
- the time perspective: what happens at which point in time

The situation as to the development of services and security relevant to the SIM looks like follows:



Letters indicate migration points. The placement of curves suggests that the level of services will probably be higher than the corresponding level of security. Based on the migration scenarios for the ASPeCT project, five different levels of service provision which reflect on the UIM can be identified:

E this represents the SIM as it is specified for the GSM Phase 2+ network

F the first step will be to integrate the SIM functionality in a generic multi-functional card. The basis will be card operating systems that allow static application management. Such a card will not support full OTA – it will support the SIM Application Toolkit features and integrate more than one application in a card. These applications will be loaded during the intialisation phase of the card's life and cannot be customised thereafter.

G the second step could be to implement the most important OTA features as described in this document. For example, adaptation of services or security parameters depending on the users needs will be possible at this point in time. This is probably the most critical step because it defines the framework for what can be done in the rest of the service side of UIM migration.

H from now on, it is possible to offer UMTS services or features to users. The way such services are supported depends on factors which are independent of card migration, for example whether a UMTS air interface has been introduced yet or not. It may be necessary to emulate some features.

K this last step of migration should be seen as a final goal. It stands for full UMTS service support, remote application management by VAS providers, more powerful smart cards and a complete suite of security functions implemented in the UIM.

For the migration of security, the steps may be:

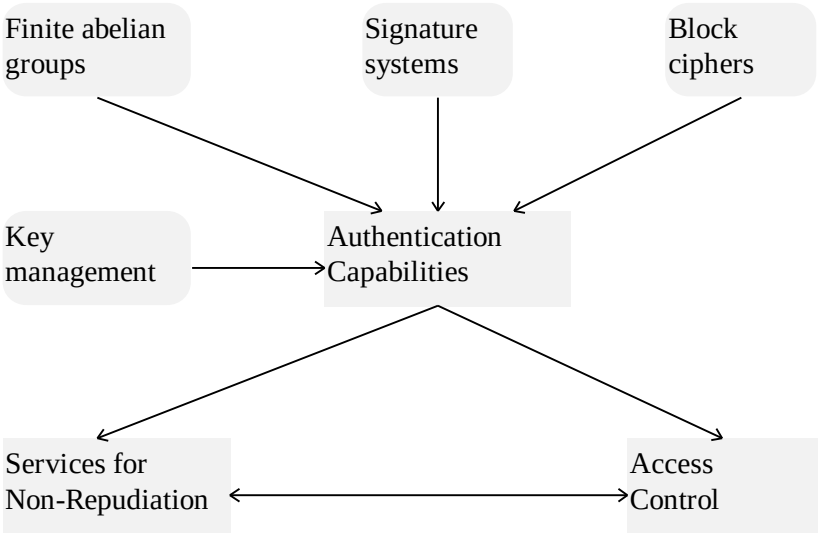
A this represents the SIM's security functionality as specified in GSM Phase 2+

B the first step will be to implement the authentication procedures and algorithms identified in the ASPeCT project. This requires all of the cryptographic modules described in this document – the point is that they are only used to implement the authentication capabilities of the UIM.

C based on the large suite of cryptographic tools, it is now possible to support the other security services – these are non-repudiation services and access control. The key management framework may not be fully developed because it depends on VAS providers and their intentions.

D this is the final goal. It stands for a fully elaborated card operating system which allows to parameterisation of the cryptographic modules and which supports multi-tasking.

Some interdependencies which have to be respected by any migration path are the following:


Finite abelian groups
Signature systems
Block ciphers
Key management
Authentication Capabilities
Services for Non-Repudiation
Access Control

5 Multiple Applications on UIMs

5.1 Introduction

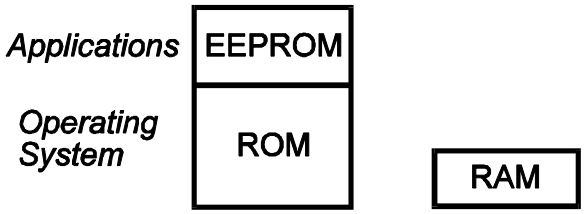
When the GSM SIM was introduced it rapidly became the largest world-wide market for smart cards. Since then smart cards have begun to win a wider acceptance in other applications and the much promised take up by the financial sector is beginning to become a reality.
As ETSI is now beginning to plan the UIM requirements for a UMTS it would seem sensible to build on the success of smart cards. Although there has been a liaison statement from ETSI [19] which suggests that the current ISO standardised smart card format may not be used for a UIM, from practical reasons this seems unlikely to happen.
The capabilities of smart cards are continually increasing and this makes the exciting development of a smart card based UIM which can support additional functionality a real possibility.
This chapter considers the problems and solutions which have and are being developed to support the idea of a multi-application smart card. It closes with a hint at the market opportunities which would be available to UMTS service providers if they could support such multiple applications on their users UIM.

5.2 Structure of SMART CARD Applications

In order to support more than one application, or indeed, even to allow a modicum of interoperability it is essential that a smart card uses a standardised structure for the applications that are stored on it. Whilst it would be entirely possible to use a customised approach for each application this would be costly for both the developers of smart card and Terminal applications – each application would require a new *learning curve* and inevitably there would be slight misunderstandings as to exactly how the Terminal and the smart card should interact.
It is exactly for these reasons that smart card suppliers have developed proprietary Card Operating Systems (COS) which support the basic functionality described by ISO, the organisation that has done the most to standardise the use of smart cards through its technical subcommittee JTC1/SC17 [1-9]. These Operating Systems also provide the platform on which the suppliers can rapidly implement new applications confident that the new application will be interoperable with other existing applications because the basic functionality is provided by an operating system which has already been proven correct and subjected to rigorous testing.
Indeed, these operating systems often allow a new application to be developed using graphical tools without the need to write any new program code, e.g. with the STARMAG tool from G&D. This has many benefits, most notable of which are the following:

- applications can be developed by parties other than the COS purveyor without compromising the security of the operating system or other applications
- the resultant application can be trivially ported between different platforms which support the COS

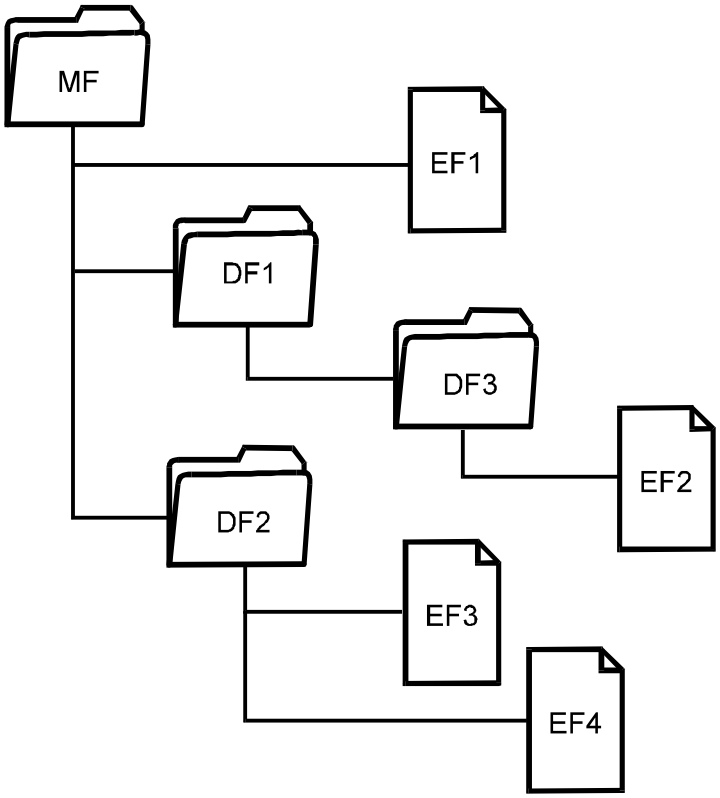
The first of these points is in fact very significant – with current day smart cards there is no hardware segregation of different applications and it is trivial to use code from one application to compromise the data in another application. In contrast, an application produced using only COS commands cannot compromise another application. This idea is discussed further in §5.4.2 below and an example of an application developed on top of the COS is given in §5.9.
The COS is application independent and is implemented in the ROM of the smart card. Individual applications are usually implemented in the EEPROM of the smart card so that they can be loaded onto the card after it has been produced and can be changed or tailored as necessary with minimal cost. Clearly, this distinction does not always hold. For example, if a special application has a large enough market (and this does not need to be all that large) then it is more cost-effective to implement the application in ROM code, along with the COS. This allows smart cards with smaller amounts of EEPROM to be used, with the associated reduction in cost. It is, of course, possible to load further applications into any free EEPROM space. Similarly, it may be necessary to modify the code in the EEPROM due to an error in the implementation or because of a change in the specification. In this case part of the COS would also be in the EEPROM.



The remainder of this section attempts to enumerate the features that provide interoperability to applications.
Note that it is the interface between the Terminal and the smart card that is important rather than the actual implementation on the smart card – that is, it is the logical model supported by the smart card that must be consistent with the terminal application. Historically, the interface between this logical model and the terminal has been based on a *File System* and the *Command Set*. More recently, the perspective is beginning to change towards an object orientated model and this is discussed below in the EMV interoperability section.

5.2.1 File Structure

An essential part of every application that is implemented on a smart card is clearly *data*. This data is normally organised in a hierarchical file system. The data in question is stored in a file which may or may not be structured. In order, to retrieve the data it is necessary to know whereabouts in the file system the data is stored and how it should be retrieved. It is possible to associate *Access Conditions* to the individual files which control how and when the data may be accessed.
Other data, which is only needed internally by the COS need not be stored in the file system because there is no external requirement to directly access it. However, it is normally simpler if this data is stored in the file system.
If the organisation of the file structure and the data in it can be standardised then the software in a terminal accessing the data can be simplified.
The files that contain data elements which are not constant must be stored in the EEPROM of the smart card. Files which are constant and the same for all cards could be stored in the ROM of the card – this is not typical and the files would probably also be stored in the EEPROM of the card.
ISO has defined a file structure for smart cards which is based on Elementary Files (EFs) and Dedicated Files (DFs). In comparing to a normal computer file system DFs can be equated to directory files and EFs to normal files. At the “root” level there is a DF called the Master File (MF).



Within the filing system there are four types of file that are supported by ISO [6] (the COS may internally support a larger number of file types):

- Transparent files
- Linear fixed length record files
- Linear variable length record files
- Cyclic fixed length record files

Transparent files contain data in an unstructured form. The other file types are record orientated which simply means that access to them is based on complete records. The access to the records is determined either by the record index or by a record indicator. The records can be of variable length or always the same length for linear files. The final type of file is a cyclic file, with this type of file the record pointer wraps around when it reaches a certain value and the earliest written records will begin to be overwritten.

5.2.2 Commands

Every application also needs to support commands which allow it to operate. These commands must support the file structure and should be sufficiently generic that they can be used in more than one application. The usual approach is to have a group of commands that can be used for all the applications on a smart card and then to have special commands associated with each application. These supplementary commands can only be used within the context of their associated application whilst the generic commands can be used by all the applications (unless they explicitly prohibit this). Note that the access conditions on the individual files can be used to prevent security breaches with these generic commands.
The generic commands are typically a component of the COS and so are implemented in the ROM code. The application specific commands are implemented in the EEPROM, so that they are only loaded onto the smart card if they are actually required.

5.2.3 Transport Protocol

In order to transmit commands to the smart card and to receive any response from it the terminal and the card must use a transport protocol that they both understand. This transport protocol forms layer 2 of the OSI protocol stack with the commands lying at the application layer (layer 7). The transport layer is responsible for implementing any error recovery procedures and thus ensuring that at layer 7 commands are transmitted in an error free manner.

There are two protocols in common usage in the smart card world and both have been defined by ISO in ISO/IEC 7816-3 [3,4,5]. One of these is a character based protocol, called T=0, and the other is a block orientated protocol, designated T=1. Note that current GSM SIMs use the T=0 protocol. The support of at least one and possibly both of the protocols will be built into the operating system.
A smart card indicates the terminal which protocol it wishes to use in the first message that it sends to the terminal after being reset – this is the so called *Answer to Reset*. Following this message, the terminal and the card can alter the settings using the Protocol Type Select (PTS) mechanism.

5.2.3.1 APDU Transport

The aim of the transport layer is the transportation of so called *Application Protocol Data Units* between the smart card and terminal. The way these are transmitted is dependent on the protocol used but the actual APDUs are the same in each case. They have the following form:
Command APDU:

CLA	INS	P1	P2	[Lc]	[Data Field]	[Le]

Response APDU:

[Data Field]	SW1	SW2

where:

CLA	Class Byte	INS	Instruction Byte
P1, P2	Parameter Bytes	SWx	Status Bytes
Lc	Length of command data field		
Le	Length of expected response data field		

5.3 ISO / ETSI / EMV Interoperability

There have been three main bodies who have published standards which allow for the interoperablity of smart cards. These are:

- ISO/IEC, in particular JTC1 SC17
- ETSI (CEN), for the telecommunications field
- EMV (Europay, MasterCard and Visa) for their payment systems

ISO is concerned with general standards and for the interoperability of equipment complying to these standards, the other two organisations only with standards appropriate to their respective fields. The standards they have produced are, by and large, compatible. This is natural because a goal of interoperability and the support of multiple applications on smart cards should be to allow any two applications to coexist on the same card.
ISO has been working on standardising commands for smart cards for a substantial period of time and finally published ISO/IEC 7816-4 [6] in 1995. This is a description of *Interindustry commands for interchange*. It, or at least earlier draft versions, have provided the main reference commands that are widely accepted by both smart card producers and application providers. Based on these commands other bodies have defined their own commands and have usually tried to remain consistent with the ISO approach. This has not always been easy since:

1. small details of the commands have changed during the development of the standard
2. the philosophy of the ISO committee has changed with time and as the membership has changed
3. there is a political need for compromise in an international standard which sometimes leads to imprecise and confusing definitions.

It is the last of these that has caused the greatest number of problems. For example, in the ISO description of Secure Messaging the concept has so many options and has so little description that almost any of the many inconsistent implementations that exist can accurately claim to support ISO compliant Secure Messaging. This is a chicken and egg situation – the ISO standard was so long in being produced that application designers who needed Secure Messaging had to guess which way ISO would jump. Then when ISO wanted to use a different approach, political pressure meant that existing applications had to be encompassed in the standard. The EMV standards attempt to narrow down the Secure Messaging definition to something workable but it really falls outside the scope of EMV activities.

ETSI is the organisation responsible for producing European standards in the telecommunications industry. Its greatest success has been the standards defining the GSM system which uses smart cards known as SIMs.
EMV is the name of the collaboration between Europay, MasterCard and Visa. Their objective is to define specifications which enable cards from all of the payment system members to operate correctly in a generic terminal but still be able to offer the specific services peculiar to each member. To achieve this objective EMV only needs to define the application interface between the terminal and the smart card.

5.3.1 Commands

The commands that are supported by a smart card are a crucial component of interoperability – without a common command set there is no chance that a card will be able to operate in a diverse set of terminals, or that it will be relatively straight-forward to support more than one application on the card.
It is also clear that not only must the format of the command and the response be standardised to allow interoperability but the functionality of the commands must be defined. This inevitably leads to the need to standardise, at a logical level, the data representation inside the card. That is, the commands must be defined in terms of some sort of functional model of a smart card. The actual mechanism internal to the card is not important as long as the interface is as expected.
The following sections summarise and describe the basic commands that have been standardised for interoperability by ISO/IEC, ETSI and EMV. It will be clear that there is a certain amount of compatibility between the commands and also a common functional goal between the commands. Nonetheless, the mechanisms that have been defined to achieve this common goal are sufficiently different to cause a less than optimal implementation in the operating system of a smart card which must support all the commands.

5.3.1.1 ISO 7816-4 Commands

ISO defines certain valid values for the CLA and INS bytes. Other values may be used but their meaning will fall outside the scope of the ISO standard to some degree. The CLA byte is coded according to the following table:

‘0X’	Structure and coding of command and response according to ISO/IEC 7816-4
‘10’ to ‘7F’	Reserved for future use
‘8X, ‘9X’	Structure and coding of command and response according to ISO/IEC 7816-4. Meaning of the command and response is proprietary
‘AX’	Unless otherwise specified by the application context the structure and coding of response is according to ISO/IEC 7816-4
‘B0’ to ‘CF’	Structure of command and response is according to ISO/IEC 7816-4
‘D0’ to ‘FE’	Proprietary structure and coding of command and response
‘FF’	Reserved for PTS (Protocol Type Select)

In the table the ‘X’ nibble represents the usage of secure messaging and logical channels.
The INS byte can be coded in any way that is compatible with the transmission protocols, for T=0 and T=1, this means the byte must be either even or not have either of the forms ‘6X’ or ‘9X’.
The ISO/IEC document describes many commands and several have many options which would lead to a smart card of high complexity and consequently of high cost! In practice only a subset of the commands are required and consequently the card operating systems available only support a subset comprising of the most commonly used commands.
The ISO/IEC document [6] describes the following set of interindustry commands:

READ BINARY

This command allows data to be read from at arbitrary offset within a transparent EF.

WRITE BINARY
This command allows data to be overwritten at an arbitrary offset within a transparent EF. The write mode (specified in the file attributes) causes either a simple write, a logical OR or a logical AND operation to be performed on the already existing data.

UPDATE BINARY
This command allows data to be updated at an arbitrary offset within a transparent EF.

ERASE BINARY
This command allows data to be erased to 0, commencing from an arbitrary offset within a transparent EF.

READ RECORD
This command allows all, or the first part, of a record to be read from a record orientated EF.

WRITE RECORD
This command allows all of a record to be written to a record orientated EF. The write modes are as WRITE BINARY.

APPEND RECORD
This command allows a record to be appended at the end of a linear structure EF or the writing of record number 1 in a cyclic EF.

UPDATE RECORD
This command allows a record to be updated in a record orientated EF.

GET DATA
This command allows the retrieval of a data object resident on the card in the current context. The object is specified using the tag provided in P1 and P2. The value field of the object is returned – that is the leading tag and length are **not** returned.

PUT DATA
This command allows the storage of a data object into the current context on the smart card.

SELECT FILE
This command allows the selection of a current file in the filing system. The selected file can be an EF or a DF. In the latter case this means that the SELECT FILE command is used to select the current application and is therefore fundamental to the support of multiple applications on a single smart card. The file can be selected by means of its file ID or by its file name.

VERIFY
This command compares the verification data provided in the command with the reference data stored on the card. The type of reference data and the comparison mechanism is left open. Typically this command is used for PIN verification.

INTERNAL AUTHENTICATE
This command computes authentication data based on a challenge sent to the card by the terminal. The terminal will be able to verify the returned authentication token and determine if the card is authentic – that is, that it is in possession of the appropriate secret key.
Note that the INTERNAL AUTHENTICATE and EXTERNAL AUTHENTICATE commands are viewed from the card's perspective. Thus INTERNAL AUTHENTICATE is used to authenticate the card.

EXTERNAL AUTHENTICATE
This command checks the authentication data received from the terminal and if it is correct then the card updates an internal security status. The authentication data is based on a previous challenge received from the card using the GET CHALLENGE command and an internally referenced secret.

GET CHALLENGE
This command requests a challenge from the card, typically a random number, for use in a security related procedure. It is normally an immediate precursor to an EXTERNAL AUTHENTICATE command.

MANAGE CHANNEL

This command is used to open and close a logical channel. Logical channels are discussed later in this document.

GET RESPONSE

This command is related to the T=0 transport protocol. As such it should not really be handled at the application layer (layer 7) but for historical reasons it is considered a true command.
It is needed only when a command must provide both data to the card and expects data back in the response from the card (a so called Case 4 command). The T=0 protocol can only send data in one direction and so if the terminal must both send data and read data back from the card, then it must issue a GET RESPONSE command to obtain the response from the card.

ENVELOPE

This command is used to transmit data to the smart card from the terminal which could otherwise not be transmitted by the transport protocol.

5.3.1.2 ISO 7816-8 Commands

ISO is currently working on a new part, Part 8, to 7816 [9]. This addresses the issues being raised by the applications of public key cryptography to smart cards. In public key methods the size and number of parameters is much larger than in symmetric key cryptography; as a result different methods are needed to efficiently support these on a smart card. Since these commands are in an early state of development no detail is given here but the following commands are defined

- SET SM ENVIRONMENT
- PERFORM SECURITY OPERATION (encipher, decipher, compute signature, hash, verify signature, verify certificate)
- MANAGE VERIFICATION DATA

5.3.1.3 ETSI Commands

ETSI has defined standards for use in the telecommunications field. These are usually later adopted as standards by CEN, for example, the payment methods standard [10].
By far the most important commands defined by ETSI have been for the GSM SIM [11] card specification. For obvious reasons of compatibility a UIM which is used in the migratory phase towards UMTS will have to support these GSM commands.
All the ETSI commands are defined with a CLA byte of ‘A0’ which means that the command and response structure is compatible with the ISO standard.
Many of the commands used by SIM cards are subsets of those defined by ISO. These include the following:

- SELECT (equivalent to the ISO SELECT FILE command)
- READ BINARY
- UPDATE BINARY
- READ RECORD
- UPDATE RECORD
- VERIFY (referred to as VERIFY CHV by ETSI)
- GET RESPONSE
- ENVELOPE

For example, the ETSI SELECT command can only be used to select a file by reference to the file ID. The file control information (FCI) sent in the response to this command is not in a BER-TLV encoded template as is the case for the ISO SELECT FILE command. Note that the ETSI version does send back FCI information but that it is implicitly encoded.
Additionally, however, ETSI introduces many other commands. These are listed and briefly described below:

STATUS

This command is used to return the same FCI for the current DF as would be returned by the SELECT command if this DF had been selected.
It is apparently also used to allow a pro-active SIM to indicate that it has information available for the terminal (pro-active polling).

SEEK
This command is used to search through the current linear fixed EF for a record which commences with the supplied pattern.

INCREASE
This command is used to maintain a counter in a cyclic EF. The oldest record in the file is replaced by a record which is the sum of the current record and the input value. The counter does not wrap around to 0.

CHANGE CHV
This command is used to change one of the CHV values. The current value and the replacement value are supplied and if the current value is correct it will be replaced by the new value.

DISABLE CHV
This command can be used to disable the use of the first CHV value. It means that the SIM believes the CHV has already been authenticated when checking the security conditions for file access. The CHV value must be supplied with the command.

ENABLE CHV
This command reverses the effect of the DISABLE CHV command.

UNBLOCK CHV
This command supplies the value of the Unblock CHV, an index to a CHV value and a new CHV. If the Unblock CHV value is correct then the CHV value corresponding to the index is replaced by the new CHV value.

INVALIDATE
This command invalidates the current EF, meaning that a flag indicating this is set in the file's header. This flag controls access to the file by further commands.

REHABILITATE
This command reverses the effect of the INVALIDATE command by clearing the flag in the currently selected file's header.

RUN GSM ALGORITHM
This command runs the A3 and A8 algorithm using the supplied data and returns data which is used by the terminal for authentication of the card.

TERMINAL PROFILE
This command is used to inform the SIM about the terminal capabilities with regard to the SIM Application Toolkit functionality.

FETCH
This command is used by the terminal to read a SIM Application Toolkit command from the SIM. The SIM will have previously indicated that such a command was available.

TERMINAL RESPONSE
This command is used by the terminal to send to the SIM the response to a previously fetched SIM Application Toolkit command.

5.3.1.4 EMV Commands

In the same way that ETSI has defined a subset of ISO commands and a set of their own the EMV group has also defined a set of commands that must be supported by terminals and smart cards used in Payment systems [12,13,14].
The approach adopted by EMV is slightly different because it started to define its command set after the ISO standard and the use of smart cards had achieved a greater level of maturity. It was also far more important to the payment organisations that the cards could support multiple applications – this is reflected in the philosophy of the commands and the usage of files.
All the EMV commands are defined with a CLA byte of '0X' or '8X' which means that they are either identical to the ISO command or else are proprietary but that the command and response structure is compatible with ISO.

The EMV specifications [12-14] define the following commands which are subsets of those defined by ISO (with a class byte of '0X'):

- EXTERNAL AUTHENTICATE
- INTERNAL AUTHENTICATE
- READ RECORD
- SELECT
- VERIFY

In addition, they define the following commands (with a class byte of '8X'):

APPLICATION BLOCK

This command invalidates the currently selected application but does not affect other applications on the card. It is a good example of a command introduced by EMV for multi-functional cards.

APPLICATION UNBLOCK

This command reverses the affect of an APPLICATION BLOCK command; it allows the currently selected application to be used once more.

CARD BLOCK

This command is used to block the card. After successful execution of this command the card will not allow any of the applications on the card to be selected. Because of the irreversible nature of this command it should only be used on a multi-application card where one party is responsible for all the applications on the card.

GENERATE APPLICATION CRYPTOGRAM

This command calculates a cryptogram which is used to authorise an EMV transaction. The type of cryptogram that is requested and returned indicates the result of the various risk management procedures that have taken place.
This command is the *raison d'être* for an EMV card in the same way the RUN GSM ALGORITHM is for a GSM SIM.

GET DATA

This command is identical to the ISO version except that it returns the tag and length of the requested data object as well as the value field. The ISO variant returns only the value field.

GET PROCESSING OPTIONS

This command is used to initiate a financial transaction. It is used to supply terminal data, which has been previously requested, to the card and returns information to the terminal on where to find the data that the terminal requires.

PIN CHANGE/UNBLOCK

This command is used to change or unblock the PIN on a card. There is no Unblock PIN as in a SIM but the command is secured using Secure Messaging.

5.3.1.5 Command Summary

Command	INS	ISO/IEC	ETSI	EMV
READ BINARY	'B0'	✓	✓	
WRITE BINARY	'D0'	✓		
UPDATE BINARY	'D6'	✓	✓	
ERASE BINARY	'0E'	✓		
READ RECORD	'B2'	✓	✓	✓
WRITE RECORD	'D2'	✓		
APPEND RECORD	'E2'	✓		
UPDATE RECORD	'DC'	✓	✓	
SEEK	'A2'		✓	
SELECT FILE	'A4'	✓	✓	✓
STATUS	'F2'		✓	
INVALIDATE	'04'		✓	
REHABILITATE	'44'		✓	
GET DATA	'CA'	✓		✓
PUT DATA	'DA'	✓		
VERIFY	'20'	✓	✓	✓
CHANGE CHV	'24'		✓	
DISABLE CHV	'26'		✓	
ENABLE CHV	'28'		✓	
UNBLOCK CHV	'2C'		✓	
PIN CHANGE/UNBLOCK	'1A'			✓
INTERNAL AUTHENTICATE	'88'	✓		✓
EXTERNAL AUTHENTICATE	'82'	✓		✓
GET CHALLENGE	'84'	✓		
MANAGE CHANNEL	'70'	✓		
GET RESPONSE	'C0'	✓	✓	
TERMINAL PROFILE	'10'		✓	
ENVELOPE	'C2'	✓	✓	
FETCH	'12'		✓	
TERMINAL RESPONSE	'14'		✓	
INCREASE	'32'		✓	
RUN GSM ALGORITHM	'88'		✓	
APPLICATION BLOCK	'1E'			✓
APPLICATION UNBLOCK	'18'			✓
CARD BLOCK	'16'			✓

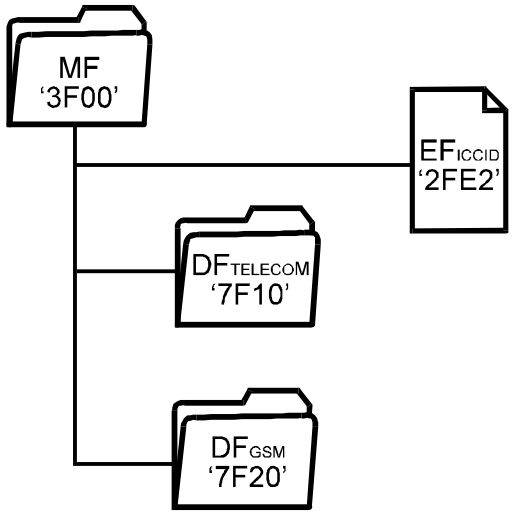
GENERATE APPLICATION CRYPTOGRAM	'AE'			✓
GET PROCESSING OPTIONS	'A8'			✓

5.3.2 File Structure

To support multiple applications, not only does the file system have to support a common type of structure but also there must be no clashes in the names or file identifiers between two applications on the same smart card.
This is not normally a problem with EFs since they are usually in the DF of their particular application, where it may be assumed collision will not occur. However, it may be necessary for some EFs to be resident in the MF and this means there could be a clash between two different applications. The problem is perhaps more problematic in the case of DFs. Typically, all the DFs will be at the MF level. Clearly, it is then essential that all the DFs have a different file ID.
Applications typically put certain security requirements on accessing files. For multiple applications to be efficiently implemented on a smart card it is desirable if the categories of file access conditions are consistent throughout the applications.
This section considers how the ETSI and EMV bodies have chosen to identify files and the security attributes that they require.

5.3.2.1 ETSI

ETSI is clearly a firm believer in the power standardisation bodies. ETSI applications, such as the GSM SIM card, identify files exclusively by their file identity. All the files in a GSM SIM card have pre-determined file IDs and there is no other way to reference the files.
Thus for a SIM to be on a card with another application it is essential that the file IDs '2FE2', '7F10' and '7F20' are not used at the MF level. These hold the files EF_{ICCID}, $DF_{TELECOM}$, and DF_{GSM} respectively, as shown in the figure. Furthermore GSM 11.11 reserves the following file IDs in the MF: '7F2X', '2F01', '2F1X' and '2FEX'.



The GSM SIM makes use of the following access conditions levels:

ALWAYS	the action can be performed without restriction
CHV1	the action can be performed if CHV1 has been presented, CHV1 has been disabled or UNBLOCK CHV1 has be successfully performed
CHV2	the action can be performed if CHV1 has been presented, CHV1 has been disabled or UNBLOCK CHV1 has be successfully performed
ADM	the conditions are determined by the appropriate administrative authority
NEVER	the action can never be performed (over the ICC/IFD interface)

These access conditions are assigned to all the commands supported by the operating system which would need to access the contents an EF for their execution. That is, the following commands:

- READ
- UPDATE
- INCREASE
- INVALIDATE
- REHABILITATE

Here READ and UPDATE refer to these commands for both transparent and structured files.

5.3.2.2 EMV File Independent Approach

EMV adopts a quite different approach to ETSI – it does not specify the file identifiers at all. A terminal supporting an EMV application does not need to contain any pre-allocated file IDs, instead it needs to know only the names of any applications which it supports. The EMV specifications explain how the terminal can determine which (EMV compatible) applications are supported by a card and how to select them. It would be possible for two card to support the same applications but to use completely different file identifiers for every file and both applications would operate correctly on an EMV compatible terminal.

An EMV application only uses short file identifiers (SFIs) for the files it needs and even the contents of these files is not specified – it is only important that all the data required by the terminal is present somewhere in these files. For an explanation of the mechanism see Section 5.4.1.1 below.

The "1PAY.SYS.DDF01" name is reserved by the EMV specifications to define the *Payment System Environment* which is the DF (in fact usually the MF) where the other EMV compatible applications are expected to be located.

Since the EMV specifications are really only a specification of the interface between a terminal and the smart card they do not define any access conditions. They do state that all the records defined with short file identifiers must always be readable with the READ RECORD command but that is all. The specifications as to when an UPDATE RECORD, or PIN UNBLOCK command is allowed are all Issuer specific and so not addressed by the standard.

5.3.3 Data Elements

As the file system that is used is important for a multi-application card so too is the way that the data elements which the applications use are mapped onto this file system. In a sense the file system is simply a way to try and organise the data elements into a manageable way – but it only postpones the problem to knowing how the data is organised inside the files.

Perhaps the best approach would be to deal entirely with the data elements themselves – an object orientated approach. This would leave the actual organisation of the objects as a matter for the COS. The fact that the EMV specifications deal exclusively with BER-TLV objects indicates that this may be the way of the future.

5.3.3.1 Storage in an Implicit Format

This is the simplest approach – the file acts only has a placeholder. Data is simply stored in an EF in a "raw" format and both the software in the card and in the terminal must know exactly the format of the files containing the data.

This approach makes it difficult to support optional data values and means that the length of data elements cannot be altered nor can the data be moved to another position in the file or to another file. The advantage is simplicity and the lack of any storage overhead.

The data can be stored in either Transparent or Structured EFs. If a structured file is used then the record number can also act as part of the index for the data elements. A structured file is normally only used in this context if there are multiple instances of data in the same format. An example of this would be the short messages received in a mobile communications environment or the log file for an electronic purse which records details of the previous transactions.

Most existing applications use this method of data storage including the GSM SIM.

5.3.3.2 Storage in an Explicit Format

In this method of storage the data elements are stored in the memory of the smart card along with their description. This is the object orientated approach to data element storage and ISO has been standardising the "description" format in ISO/IEC 7816-6 [8] and ISO/IEC 9992-2 [15]. These are based on a BER-TLV encoding of ASN.1 [16].

Ideally the objects could then simply be stored in the card using a PUT DATA command and read back from the card using GET DATA. There remains, however, the problem of assigning access conditions to the accesses. This is best solved by grouping the data together into constructed data objects with the same access conditions and these can be attached to the constructed object. These constructed data objects can then be stored by the card using internal mechanisms or in a linear variable EF and then the access conditions would be defined for the particular file.
The EMV specifications adopt this storage method and they store some data in a way that is only retrievable by GET DATA and other data in EFs which can be read out by the terminal application.
An example of a data element that is only retrievable by means of GET DATA is the PIN Try counter:

Tag	Length	Value
'9F17'	1	PIN Try Counter

An example of a data element that is stored in a linear variable file; and so returned by means of READ RECORD, is the entry in the Directory Definition File, for example the following record:

← Application Elementary File Data Template →									
		← Application Template →							
				← ADF Name →			← Application Label →		
'70'	24	'61'	22	'4F'	9	'A0 00 00 00 03 60 10 00 00'	'50'	9	"VISA CASH"
Tag	**L**	**Tag**	**L**	**Tag**	**L**	**Value**	**Tag**	**L**	**Value**

As can be seen by this example, the drawback of this approach is the high overhead that results in providing the flexibility of the object orientated approach. In this example 18 bytes of data have required a total of 26 bytes of memory.
It should also be pointed out that the newest draft of ISO/IEC 7816-8 [9], which is concerned with public key security commands uses TLV objects almost exclusively.

5.3.4 Security

5.3.4.1 Access Controls

As mentioned above, a smart card restricts the access to certain data by attaching security attributes to the file containing the data. The smart card maintains an internal state which reflects any commands that have been executed and which have increased the security level. Commands which increase this security state are limited to commands which have successfully verified the CHV data, those which have successfully performed an EXTERNAL AUTHENTICATE command or commands which are sent using the Secure Messaging mechanism.
This topic is specifically related to the applications under consideration and is left very open by ISO. As a result it proves to be one of the barriers to the easy implementation of multiple applications on one smart card.

5.3.4.2 Secure Messaging

Secure messaging is, in effect, a supplementary layer in the OSI reference model. It lies between the transport layer and the application layer and provides a combination of the following services:

- Message Integrity
- Message Confidentiality

It achieves message integrity by appending a MAC to the data field of every command. The message confidentiality is provided by enciphering the transmitted data field. In most applications the algorithm used is DES but ISO makes no restriction on this and any algorithm could be used.
The type of Secure Messaging being used is indicated by 2 bits in the class byte of the command and by the BER-TLV template which is used to encapsulate the transmitted data. The mechanism is very general and allows the tag of an object to determine whether it should be included in the authentication value. In the case of message confidentiality it is also necessary to include a padding byte indicator to show the padding method that has been used. It is even possible to use secure messaging to conceal the instructions that the smart card is carrying out; since the INS byte can be enciphered.

Unfortunately, because of the complete generality of the ISO specification there is no clear consensus as to how secure messaging should really be implemented. Coupled with the lack of a clear-cut secure messaging layer in the system this makes secure messaging one of the hardest components to integrate into a COS or a multi-application card. It is to be hoped that a common method of using secure messaging will begin to emerge and simplify the problem.

5.3.5 Logical Channels

Logical channels are defined in the ISO specification as a means to have more than one application on a smart card open at once.
The logical channel being used is indicated in 2 bits of the CLA byte of every command and thus means there can be no more than 4 applications open at any moment.
The card still operates as a single task processor – that is it will not accept a second command before the response from the previous command has been returned.
Whilst this mechanism is elegant and could well prove useful in the future it is really beyond the reach of current smart cards. They simply do not have the memory resources to support more than one application at any one time – unless one of the applications is very simple.

5.4 Problems with Multiple Applications

This section is concerned with the problems that arise if one attempts to implement more than one application on a smart card – it also highlights some solutions to these problems.

5.4.1 Selection

One of the largest problems that arises when a smart card contains more than one application is how to determine what applications the card supports and how to select the appropriate application.
In one sense there is no problem. The terminal can simply try to select the appropriate application in the same way that it would on a mono-application card, or in the simplest case it could just assume that the application has already been selected (under the ISO specification it is possible for one application to always be selected following a card reset). However, such an approach requires the user to manage the applications – if a card is presented that uses the same DF file ID that the terminal expects then at best only confusion will arise as the terminal sends commands to the card.
If sufficient care is taken and there is no conflict of file IDs on the card then a multi-application card should work without problems in a terminal that is expecting only a mono-application card. However, in situations where a terminal can support a wide range of applications it is inefficient for the terminal to try selecting all the possible applications. It is more sensible to ask the card what applications it supports and then the terminal can choose which of these it would like to use. Traditionally, the terminal has more intelligence than the card and has the added advantage that it can request the user to select an application from a list of appropriate ones: for example to choose between a credit card and a debit card payment for a particular transaction.
ISO foresaw the need for a smart card to inform the terminal what applications it supports when they proposed the possibility for a card to indicate a DIR file in its ATR. However, the content of the DIR file is defined to be outside the scope of the 7816-4 standard and consequently does not provide the basis for multi-application support.
The EMV specifications propose a solution to this problem for payment system cards and it is quite possible that this approach will be adopted in a more general context.

5.4.1.1 EMV Application Selection

As stated above the EMV specifications do not require any fixed file IDs to be allocated. This is achieved by the card informing the terminal which files it should read and by using full BER-TLV encoded data objects in these files to identify all the data that the terminal is likely to need.
The flow of the EMV application selection process is as follows:

1. The terminal explicitly selects the file “1PAY.SYS.DDF01” using the SELECT command. This selects the payment system environment which contains all the application DFs.
2. From the FCI returned by the SELECT command the terminal retrieves the SFI of the payment system directory file (EF_{EMVDIR}). This is a file in the payment system environment which contains information about all the applications in this DF. It is organised as a linear variable file and the terminal reads all the records of this file.

3. If the terminal finds any applications that it supports as it reads the records in this file then it adds them to a list of candidate applications for final selection.
4. If an entry in the directory file indicates there is another DF which contains applications and also its own payment system directory then the terminal should recursively select this DF, read the directory file and process the records in it, finally returning to the processing of the original directory file.
5. At this point the terminal has a list of applications which could be selected. Based on application specific information it will either select one of these applications or offer the choice to the user. The terminal decision is based partly on the Application Priority Indicator which is present in the records of the directory file.

5.4.2 Independence of Applications

Currently available chips for smart cards are primitive components which do not support the more sophisticated hardware features present on newer processors. One of the most serious omissions from the multi-application perspective is the absence of any form of Memory Management Unit (MMU). Semiconductor manufacturers are beginning to address this problem and it is clear that the next generation of smart card processors will support real memory management.
The lack of memory management means that there is no way to prevent code from one application reading the data for another application. This has obvious consequences for the security of cryptographic keys or other confidential information stored on the card. The security of all the applications on the card is reduced to the security of the weakest one.
Note that this problem does not arise if the applications on the card are implemented using only the features of the operating system. A COS will typically prevent the interference of two applications by the following mechanisms:

- ensure all READs and WRITEs to EFs do not extend outside the range of the EF
- ensure an application does not read from or write to a data address outside the range defined for the current DF

These mechanisms of the COS mean that multiple applications can coexist on a card without interference. However, if an application uses real processor code rather than simply COS constructions then the new application can bypass the protection mechanism of the COS – there is no real solution other than hardware memory management.
One solution to this problem would be to ensure that all implementations of applications are certified by an independent third party which could certify that the application will not misbehave and access another applications data. Nonetheless this could be difficult to prove and would increase the cost of implementation and so the final product.
This approach would also require that a service provider must approve the usage of his application on a card containing specified other applications. This is certain to prove difficult to obtain in practice. If all the applications come from the same organisation then there should be fewer problems: for example, if a banking organisation wanted to have a credit and debit card functionality on the same card, or if a telecommunications operator offered a range of services.

5.4.3 Multiple PINs

If a smart card has more than one application loaded onto it, and these applications require the use of a PIN then it is likely that the user will have to enter a different PIN for each application. This may not be the case if the two applications come from the same service provider, e.g. a bank, but in the general case it will be true.
This leads to certain potential security problems:

- if the PINs are different then the user is more likely to record them rather than simply committing them to memory. This written record may then be discovered.
- if possible, the user is more likely to disable the PIN in one or more of the applications
- if possible, the user is likely to change all PINs to the same value, thereby lowering the security to the level of the weakest system.

There is no obvious solution to this problem – banking applications have so far adopted the approach of having one PIN which is used to unlock the card for all the applications.
The use of alternative cardholder verification methods, such as the biometric methods discussed later in this report, may prove to be the answer.

5.4.4 Protocol

Another potential problem that may arise in practice is how to cope with a multi-application smart card where the applications must use different transport protocols. For example, the GSM SIM card uses the transport protocol T=0 whilst the electronic purse being offered by Visa (*Visa Cash*) uses the T=1 protocol. Could these two applications be implemented on the same card ?
Of course, the transport protocol should really be transparent to the terminal application and so it should be possible to simply change the protocol used by one of the applications. However, although many terminals can support both protocols (this is an EMV requirement) it is not always the case and there is an alternative solution.
The EMV specifications talk about a 'cold' and a 'warm' reset. Essentially, the card is reset and the ATR is returned. If the terminal accepts this ATR, and consequently the card's choice of protocol, then operation continues as normal. If the terminal does not accept the ATR then it resets the card again (a 'warm' reset) and the card can try offering a different ATR. If the new ATR is also not acceptable then the transaction is aborted.
The ETSI SIM specifications are similar, except that the terminal must not reject the SIM until at least three consecutive unsuccessful ATRs are returned.
In the example given above, the card should return an ATR indicating T=1 the first time so that an EMV terminal would accept the ATR and use T=1. A GSM mobile would reject this ATR and reset the card at which time it should return an ATR indicating T=0 which would be accepted by the terminal.

5.4.5 Physical & Electrical Compatibility

Physical compatibility is an obvious consideration for a multi-application card. That is the card must have the same physical size and possess the same electrical characteristics as those expected by the appropriate terminal.
ISO has standardised the physical and electrical characteristic of smart card in ISO/IEC 7816-1,2 and 3 [1,2,3]. These standards have been adopted universally and have been used by other standardisation bodies.
However, technical reasons have caused the other organisations to further develop their specifications according to the needs of their applications.

5.4.5.1 ETSI Modifications

In the GSM world the size of the plastic carrier was excessively large for some mobile terminals and since the SIM is not often removed from the terminal; ETSI defined the micro-SIM. This is a plastic carrier for the chip module but is a much smaller dimension. In fact it is so small that some chip modules cannot be used as a micro-SIM because they are too large.
Similarly, the power requirements in a mobile terminal are far more critical than in a bank terminal and so there is a move afoot to modify the electrical characteristic standards for smart cards so that 3V chips can be used since this voltage level is better for the mobile terminal, such a proposal is described by ETSI in [18]. It is clear that the 3V chips must be able to survive being inserted into a normal terminal but need not operate with this voltage. ISO is modifying 7816-3 to account for this change and the new revision is currently a Committee Draft [17]. The new standard defines Class A and Class B operating conditions, these correspond to the 5V and 3V respectively. The interoperability of cards and terminals for these is given by the following table:

ICC Type / IFD Type	Class A	Class AB	Class B
Class A	✓	✓	
Class AB	✓	✓	✓
Class B		✓	✓

5.4.5.2 EMV Modifications

In EMV applications it is clear that smart cards supporting cryptographic coprocessors are becoming more common. These devices often need larger current consumption than conventional cards. This is particularly so when the new cards contain an internal clock multiplier so that the crypto-processor can run at a higher rate than the main processor. As a result of this the EMV specifications

recommend that a terminal is capable of supplying a steady state current of 200mA even though the ISO specifications limit the maximum current to 55mA.

5.4.6 Blocking / Unblocking / Disable

In a mono-application card the situation is quite clear when it comes to blocking or disabling a card. Either the card can be used or it can't. With a multi-application card the situation is not so clear and it becomes important why a application provider would want to block an application.
Clearly an application can be blocked or unblocked by an application provider as he sees appropriate. Blocking one application should not affect other applications on the card. Typically blocking an application prevents the application from being used but still allows data from the application files to be read out.
Disabling a card will prevent the use of all the applications on the card and so should only be done with the consent of all the application providers. Typically it is not possible to perform any operations on a disabled card.
It thus appears that a card should only be disabled if it is in the interest of all the application providers for that card. This may be the case if a card has been reported as stolen or if one of the applications has been compromised and may therefore put the other applications at risk. In other cases the application should only be disabled.

5.5 Benefits of Multi-Application Cards

This section addresses some of the benefits that can be achieved by allowing more than one application to be stored on a smart card. This is considered from a technical perspective of the card producer or the application provider rather than the user. The main advantage to the user is enhanced services without the need to carry multiple cards.

5.5.1 Shared Code & Data

By implementing more than one application on a card it is possible for an application provider to share data between the applications and share the code used to implement the application. This means that the application provider can save on the cost of the cards distributed to the user – there are fewer of them and they will require less memory than a solution in terms of mono-application cards. (This assumes that in the future a card containing, for example, 8K of EEPROM will be cheaper than two 4K EEPROM cards.)

5.5.2 Delayed Loading of Applications

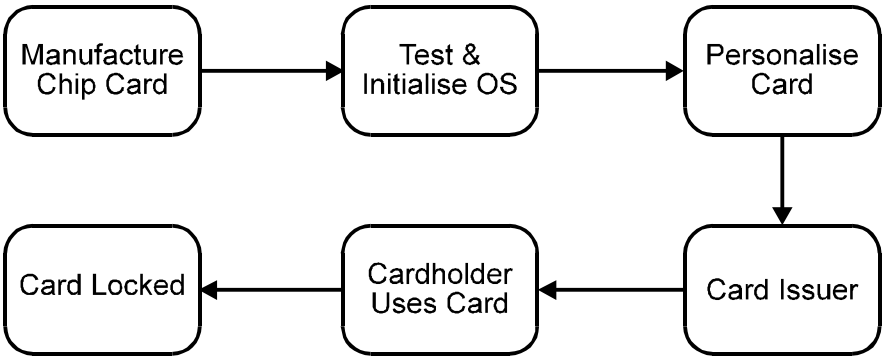
Another advantage to the application provider is the option he has to defer the loading of an application. Cards can be issued to users and then at a later date they or the application provider can choose to have additional applications loaded.
This allows for a dynamic situation where the application provider can enhance the value of cards that are already in use and is substantially cheaper than the costs of issuing new cards as users request more or different services. In the case of mobile telecommunications there is also the opportunity to load a new application over the air interface – it would not be necessary for the user to present his card for the change, in fact the user may not even need to know that the new application has been loaded onto his card.
There are clearly management issues that arise in this situation:

- Who can load a new application ?
- Who can delete a new application ?
- How can the process be performed securely ?

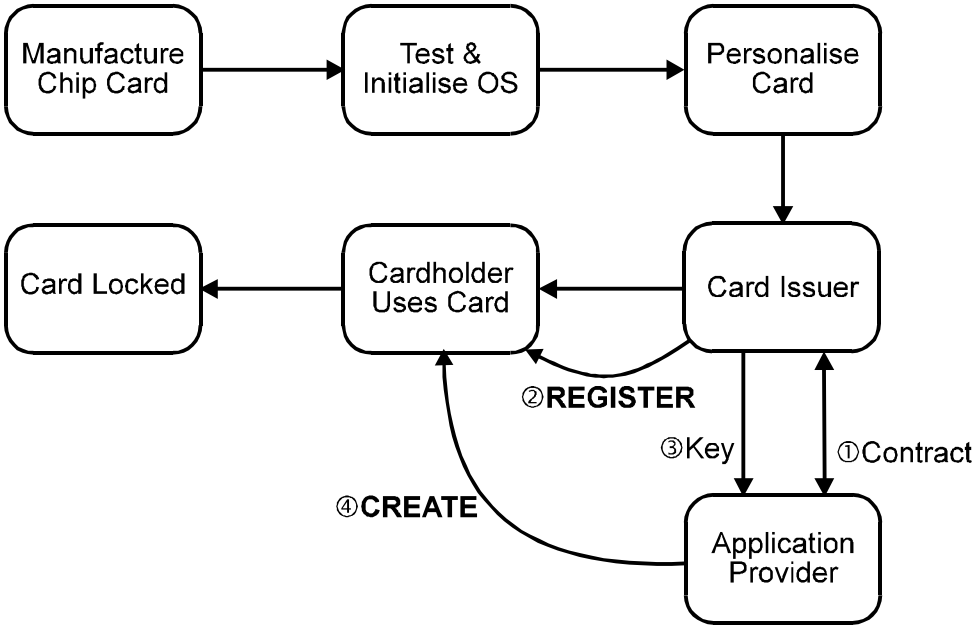
The following description explains how these questions are answered in the G & D STARCOS cards and operating system. Although it is a proprietary solution the following description shows the generic features which must be present in all such approaches.

5.5.2.1 STARCOS® REGISTER & CREATE Commands

The STARCOS operating system includes two commands CREATE and REGISTER so that applications can be loaded onto the card after the card has been issued.
Before explaining the process it is important to understand the typical life cycle for a mono-application card. This is shown in the following figure:

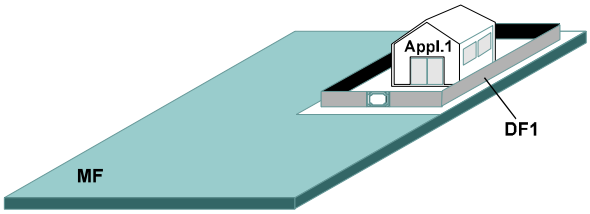


The important phases here are the personalisation and issuance of the card. The new life cycle for a multi-application card is shown in the following figure:

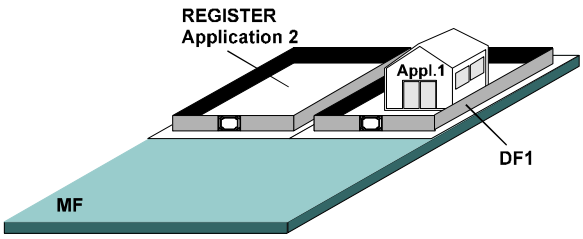


The difference is principally that the card issuer makes a contractual agreement with the application providers. At this time the card issuer loads any initial applications onto the card and reserves space for any new applications that the provider may later want to install. These are achieved by the CREATE and REGISTER commands respectively. The CREATE command creates a new DF on the card and initialises it with the application whereas the REGISTER command reserves space for a DF and allocates a key to this space. Later the application provider can perform his own CREATE command to create a new DF in the space previously reserved. He is then able to load the new application into this DF. The CREATE command is secured by a key that belongs specifically to the reserved space on the card.
The following set of pictures graphically illustrate the idea.

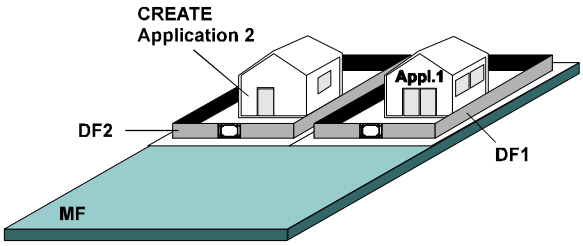
Master File (MF) and Dedicated File (DF1)



Register a new application



Create new application



5.6 Legal issues

In order to allow multiple applications on smart cards there will have to be some sort of legal agreement between the application providers. This is in addition to the contract between the application provider and the card issuer as shown in the previous section.
This legal agreement will need to cover at least the following issues:

- liability if compromise one application reveals secret information from a second application
- liability if one application denies the customer the use of the other application due to an intentional card blocking operation or due to card failure
- non-disclosure agreements to cover the auditing of the application code and specifications
- responsibility for the disabling of the card in appropriate circumstances

5.7 User Acceptance

It is also important that a user accepts the concept of multiple applications on a smart card in his possession. In general, a user should welcome the idea of multiple applications on a single card since it will free him from the problem of carrying multiple cards around.

5.7.1 Ease of Use

However, the ease of use must be considered: for example, it would probably be more inconvenient for a user to have to remove his SIM card from his mobile phone in order to be able to insert it into an Automated Teller Machine than it would be to have two cards. If the second application can be used whilst the card/SIM remained in the mobile phone then it would be perceived as an improvement by the user.

5.7.2 Common PIN

As discussed above, the issue of multiple PINs raises complexity for the user. The user is likely to regard the smart card as a unit as a whole which needs only one PIN to enable its use. If he is forced to use a different PIN for the different applications this could lead to problems. For example, if a SIM card also contains the mechanism for electronic payment and the user wishes to perform a payment transaction over the air then it does not seem acceptable that two PINs must be entered: one to enable access to the phone and the second to enable the electronic payment to take place.

5.7.3 Anonymity

A final issue which arises in the context of multi-application cards is anonymity. If one of the applications on the card is naturally anonymous – for example, an electronic purse can be operated in an anonymous way so that the purchases cannot be traced to the person spending the money – then it may lose this property if combined with another application. If the electronic purse is also on the same card as a credit application then there may be an association between the two which means that tokens spent by the electronic purse can be related to the credit card holder.
There is a sizeable minority of people who would not accept the use of payment transactions if it compromised their anonymity.

5.8 Market Opportunities

The advent of multi-application smart cards presents many new opportunities to all the application providers. The following list suggests some examples of multi-application smart cards which would allow a mobile network operator to offer value enhanced services to his customers.

- **Pay Card for Mobile Phone Use**
  Currently a GSM SIM is allocated to one particular customer. At the time of enrolment various financial checks must be made so that the network operator can be certain that the prospective user will not be a credit risk. There is certainly a market for a card that could work in the same way as a conventional phone card. This card could be implemented as a dual application card – one application being a GSM SIM and the other being some sort of electronic purse. The purse could be debited by the terminal during the course of the call setup and as the call progresses.
  This would be an example where the logical channel concept of ISO would allow both the SIM and the electronic purse applications to be active at once. Implementing the solution in this way would mean that the SIM functionality would need minimal modifications.

- **EMV Payment – Handset as terminal**
  For an electronic payment transaction following the EMV guidelines the terminal requires no security module. This means that the mobile equipment terminal and base station could offer the user the opportunity to pay for goods or services over the air. By using appropriate messaging the payment request and transmission of the authorisation cryptogram could be transmitted from the card. This would be an improvement for both the merchant and the user – the merchant is certain to receive payment and the user is saved the effort of dictating his credit card details and has the security assurance provided by an electronic payment.

- **Magnetic Stripe / Magnetic Strip Image**
  This is similar to the above case, but the second application on the card is not a full blown EMV debit or credit application but rather the magnetic stripe image of a conventional payment card. This would also allow over the air payment but would not offer the same security advantages – the security level would be the same as conventional magnetic stripe credit cards. This information would also be enough to offer the customer to perform *Telebanking* transactions using his mobile terminal.

- **Electronic Purse Interface Device**
  A mobile telephone contains a card reader and a user input device which makes it ideal as a tool to read information or initiate transactions with a smart card. This would make it possible to load value onto an electronic purse by making a telephone connection to the purse issuer or to review the log of previous transactions and the balance. Another possibility would be to use a mobile terminal as the so called *Electronic Wallet* which is used to transfer money between purses in the Mondex electronic purse scheme

5.9 Implementation Example using STARMAG

As mentioned previously, it is possible to build applications that are guaranteed not to compromise the security of other applications on the card if they just use the COS functionality. Although, the capability of such applications is limited to the operations supported by the COS, nonetheless it is possible to build quite sophisticated applications. The following, gives an example of the kind of thing that is possible using the STARMAG tool for cards supporting the STARCOS operating system. Similar approaches are possible with other operating systems. If the COS has functionality for EMV transactions built in then such a tool could also be used to initialise a payment application.
STARMAG is a Windows application which allows an application to be defined using graphical tools. The kernel of the approach is that the smart card contains a finite state machine. The state transition table for this automaton can be defined for any DF. The events which cause the automaton to change state are security events generated by sending commands to the card. For example, a possible event could be a VERIFY command using a PIN or an EXTERNAL AUTHENTICATE using a DES key. The application developer first creates the file structure that he wants to have on the card. This will include all the DFs and EFs that the application will need. The initial contents of these files can be

defined. It is also necessary to create an Internal Secret File – this file cannot be read out of the card and contains both any secret keys required in that DF and also the transition table for the state machine. Finally, the developer must define the access conditions for the various files in the DF – this is simply a list of the files that can be read or written to when the state machine is in a particular state. The following table represents the transitions table for a simple application defined in one DF. The full example program contains 3 other DFs. The columns represent the different states that are possible when the MF is selected (only MF00) or when the current DF is selected (DF00 through DF05). From the table it is clear that after an EXTERNAL AUTHENTICATE with KEY01 (a DES key) the automaton will jump to state DF02. From this state a VERIFY with KEY02 (a PIN) would take us to state DF03 whilst a authentication with KEY04 would take us to DF01. The lower half of the table shows the file accesses which are possible in the different states. In state DF00 the file with ID '0004' is readable, all other files have neither read nor write access. In state DF04 it is possible to read and write to the files with IDs '0001', '0002', '0003', or '0004'.
The security of the other applications on the card relies on the fact that the READ or WRITE accesses are verified by the operating system to ensure that they have valid addresses for the file being accessed.

KEY \ State	**MF00**	**DF00**	**DF01**	**DF02**	**DF03**	**DF04**	**DF05**	
KEY01-DES		DF02	DF02	DF02	DF02	DF02	DF02	
KEY02-PIN				DF03				
KEY03-DES					DF05			
KEY04-DES		DF01	DF01	DF01	DF01	DF01	DF01	
KEY05-PIN			DF04					
KEY06-PUK								
File Access		0004R	0004R	0001W	0001W	0001W	0001W	
				0001R	0001R	0001R	0001R	
				0004R	0002R	0002W	0002R	
					0004R	0002R	0003R	
						0003W	0004R	
						0003R		
						0004W		
						0004R		

6 Biometric methods for user identification

6.1 Introduction

In the GSM world, the use of SIMs is protected by a Personal Identification Number (PIN). After the SIM is plugged into a handset, the user is asked to type in a number with between four and eight digits. This number is sent to the card and compared with a stored value. If a match occurs, the terminal is ready for use as long as the SIM is inserted into the handset.
This approach to verify the identity of a person who claims to be the owner of the SIM has several drawbacks:

- a PIN is not easy or natural to remember
- knowledge of the PIN does not prove the identity of the user
- if there is more than one application in the card, each one will probably have its own PIN

In short, a PIN is a simple solution to the problem but it is too clumsy for easy use. A lot of people avoid the issue of user identification by simply turning off the PIN protection. This is an option offered by most SIM cards. But this means that there is no protection at all. Anyone may steal the SIM and use it for international calls before the loss is detected and the operator has barred the card. The solution for the future will probably lie somewhere between zero protection and the user being forced to remember 10 different PIN values.

6.2 Cardholder Verification Methods

From a general point of view, the PIN is a method to verify the identity of a person who wants to use a smart card. In this section, alternative cardholder verification methods, or CVMs, to the PIN will be considered. These alternatives are typically based on biometric methods. The first subsection looks at objectives of such methods and identifies some requirements on systems which are based on them. Exactly which system fits best the application needs is at the heart of the performance section. Many factors influence the selection of a biometric method; the most important ones being speed, error rates and robustness. These topics will be discussed.
Although biometric technology is comparatively well developed, it faces some serious threats which are often overlooked. From most viewpoints there is a big difference between an attacker gaining knowledge of a users' PIN (which could then be changed) and an attacker being able to successfully use a description of the users voice characteristic to pass a biometric authentication. This difference should be reflected in the templates which are used in the comparison.
The last subsection gives a survey of existing CVM solutions.

6.2.1 Objectives

The primary objective of CVM in a UMTS environment is to prevent unauthorised use of the mobile equipment. To this objective must be added the following:

- the authorisation method must be portable between different mobile terminals
- the authorisation method must be acceptable to the user

The first of these points implies that the authorisation must use the UIM which is the component in UMTS which provides both the security environment and the personalised service portability. Because the UIM is a secure computational device it can also provide the necessary confidentiality for the CVM data and also perform at least part of the verification comparison.
The remainder of this section will only consider authorisation methods which are based on a portable UIM device, which shall be considered to be a smart card.
Note that the requirement that the CVM is portable may conflict with the requirements of the CVM adopted as far as data storage and processing power are concerned.
Turning now, to the question of how to prevent unauthorised use of the mobile equipment, there are two approaches to the CVM:

1. *User verification*: this means matching a claimed identity with some expected identity and results typically with a *yes* or *no* answer, or else a probability value which measure the likelihood that the user is who he claims to be.
2. *User identification*: In this case data is presented which is characteristic of a registered user. This data is compared with a stored list of similar data and, if found, the matching identity is returned. If no match is found the result indicates that the user is *unknown.*

A PIN is a popular solution to the CVM problem because it serves both purposes. A card can compare a PIN with a single expected value or else it can look it up in a table of acceptable PINs.
More generally speaking, the way the identity of a user is verified can also be classified as follows:

- the user presents something he or she *knows*: a PIN or a password
- the access is granted based on something the user *possesses*: a key stored in a card for example.
- the verification is based on something the user *is*: a characteristic used must be unique and stable. Both biological as well as behavioural properties of the human body can be employed.

From this classification it is clear that there are many possible alternative means for CVM as opposed to the simple PIN method.

6.2.2 Features

Independent of the particular method for cardholder verification, there are some features which are common to all of them:

- the system which performs the verification comprises the card, the terminal and other additional devices.
- there must be some processing power and a data repository in the system
- the presentation of a claimed identity involves a user interface: this may range from a simple keypad to more sophisticated sensors which form a part of the system.

For example, the *PIN system* consists of a card and a handset which has a keyboard and display to allow the input of the user PIN.

An important point to note is that the detailed distribution of processing power, memory elements or sensors is completely system and product dependent. For example, it may prove necessary for the terminal to provide some pre-processing of the data before the smart card makes the comparison. This is sensible because the terminal normally has greater processing and memory resources than a card. Similarly, the terminal may store some reference data or look up tables that are not user specific – this would save storing such information on the card.

Another commonality between different systems is that they all need to *enrol* users before they can be verified. In the PIN context, this means that a PIN must be entered into the system before a user is able to use it. In biometric systems, the situation appears different but is really the same – a user has to allow the system to *learn* about the particular characteristic which is used for verification. This may be a sample phrase spoken into a microphone.

6.2.3 Performance

This paragraph looks at ideas to classify methods for cardholder verification. It tries to identify a way to find out which method is the best choice for a given purpose. Such a choice is clearly dependent on the performance of the mechanism and the weighting that is attached to the different performance criterion. This weighting is clearly heavily application dependent – 100% accuracy may be nice for UMTS but not if it comes at an extortionate cost.

6.2.3.1 Cost

This is probably the most important criterion. No CVM would be adopted unless the benefits it provides are perceived to be worth more than the cost of implementation.

6.2.3.2 Speed

The speed with which the CVM can be performed is important – especially from a user perspective. In order to be welcomed by a user the verification process should not be perceived to take a long time. This criterion is often more important for the training period. If a user has to repeat a test phrase 100 times in order to train the equipment then it will not prove acceptable.

It is clear that enrolment for a PIN scheme is negligible and the actual entry and verification of the PIN also requires negligible time so alternative CVMs must outperform a PIN in other characteristics if they are to be selected.

Depending on the CVM method it may be possible to partition the algorithm between the processors in the terminal and the card to perform any processing rapidly. If the sensors require a long delay to achieve their reading then alternative CVMs may have to be used.

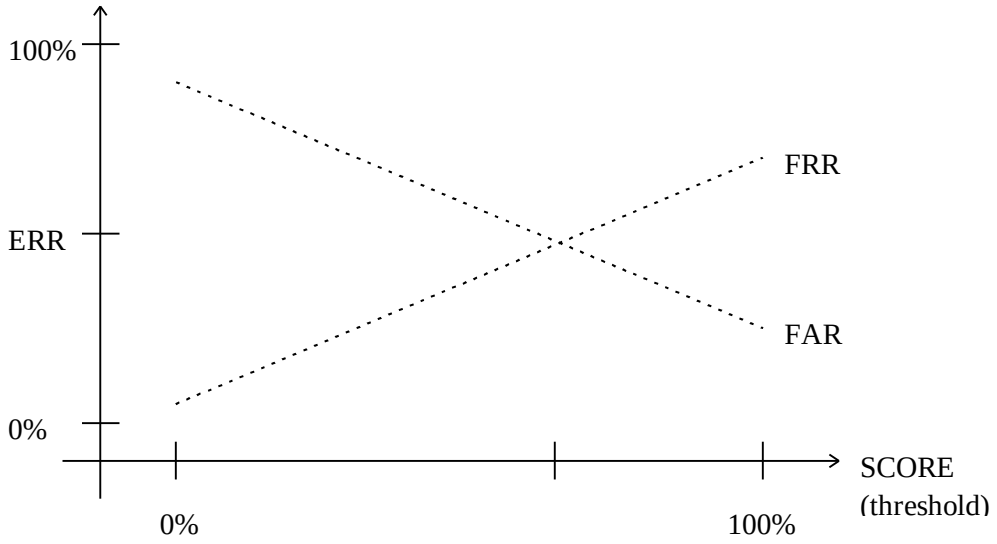
6.2.3.3 Errors

A PIN has another advantage over other methods: a PIN is, in effect, a previously digitised input value from the cardholder. Validation of the PIN is a simple comparison – the PIN either matches or it doesn't. Almost all other CVMs will have an analogue input which must be digitised before comparison. It is not possible to perform an exact match on this sort of data and so the results from the comparison are inevitably fuzzy.

For example, each sample of a person's voice will sound different but nonetheless there is a characteristic that enables the human ear and brain to detect that it originated from the same person. The samples must be correlated using an appropriate process and if the correlation is sufficiently high then the samples can be claimed to match.

The way this correlation process can be used as a CVM is as follows: a sample of the particular characteristic (voice for example) is taken and after some pre-processing is stored as a reference value.

This reference value is called the *template*. Everytime a user has to be verified, a new sample is taken pre-process and then correlated with the template. This results in a *measured score* in contrast to the *selected score* or *threshold* which defines a lower threshold for the measured score. If the measured score is greater than the threshold then the user has passed the CVM.
Depending on the *threshold*, some unpleasant things may happen. If the threshold is too high, a person who wants to use a card could be incorrectly rejected because, for some reason (perhaps his voice sounds hoarse) the correlation score was too low. This probability with which this event occurs is called the *False Rejection Rate (FRR)*. The higher the score, the higher the FRR.
On the other hand, a low score provides a low system security – the measured score of an impostor may be higher than the threshold and he would then be granted access. The probability with which the event that an invalid user is accepted by the system is called the *False Acceptance Rate (FAR)*. The lower the score, the higher the FAR.
In general statistical testing a different terminology is used: the rejection of a valid user is referred to as a *Type 1 error* and the acceptance of an incorrect user is a *Type 2 error*.
Note that a user will not tolerate many Type 1 errors because this means that he is being denied service. Conversely, the network operator will not tolerate many Type 2 errors because they are providing unauthorised service. These two errors are in tandem – it is not possible to reduce the Type 1 errors without allowing more Type 2 errors and vice versa.
The following picture illustrates this:



Because both curves are continuous, there is some score value which allows an equal error rate (ERR) for both FAR and FRR. This is one of the most interesting characteristic values of CVMs.
For the PIN system, the above picture consists only of four isolated points, namely (100%,0%) for FAR and (100%,100%) for FRR if the PIN is enabled (score of 100%) or (0%,100%) for FAR and (0%,0%) for FRR if the PIN is disabled (score of 0%).
It is clearly a difficult decision on how to trade off the FRR and the FAR – this must be done by considering both technical and marketing perspectives.

6.2.3.4 Robustness

Another important property of CVMs is their robustness. This is a measure of how tolerant the system is to background noise or other effects, such as a voice recognition system for someone with a cold..
A PIN based method is robust because the values are entered digitally and there can be no error from that point. It is perhaps debatable whether a user will be able to correctly recall his PIN at moments of high stress, or whether the transposing of two digits in the PIN (a very common human error) should be counted as a robustness problem or simply entering an invalid PIN.
Other CVMs are not so robust, a voice recognition system may fail in a noisy background environment and for fingerprint systems it is claimed " Younger people have a softer skin and it is therefore harder for an electronic device to read their fingerprints. The typical nightmare of a fingerprint reader is a young, female, Asian worker in a coal mine with sweat and grease on her fingertips."

The lack of robustness can severely affect where and how practical a system really is for use.

6.2.3.5 Ease of use

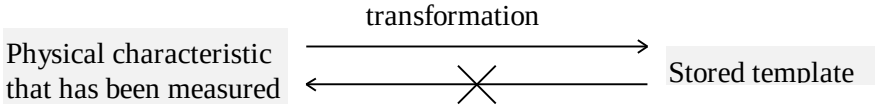
Even the best system is not worth a penny if it is not accepted and used by the intended audience. In a high security working environment users may be prepared to submit to inconvenient practices, such as a retinal scan, before gaining access to a room. In the consumer field a CVM which is not easy to use will encourage customers to say *no* and go somewhere else to for their purchases.
Any CVM must be easy to use if it is intended for a mass market application.

6.2.3.6 Security

Last but not least, the CVM itself must be sufficiently secure enough to provide to verify a user's identity but should not be more secure than need be. This means for example that PINs should not be so long that they encourage the user to write them on the card.
In a more general setting, templates can be regarded as a new sort of keys. They should therefore be securely stored and not transmitted off the card in plaintext.

6.2.4 Threats

As far as possible a CVM should try to prevent the obvious attacks on it. Although it is difficult to generalise on the attacks against CVMs in general the most basic attack would be the *replay*.
The obvious example would be that a CVM based on voice recognition should not accept a tape recording of the valid users. This means that the pass phrase used must be dynamic. Similarly, it should not be possible to fool a fingerprint system by presenting a latex copy of a fingerprint.
The above two examples used physical copying. Ideally it should also not be possible to record the data which is sent to the card as the template for comparison and then simply replay this at a later date. This again requires some sort of dynamic challenge and response in the CVM method.
Notice that a PIN is completely succumbs to these two forms of attack, unless the terminal contains a secure PIN pad.
If the CVM data has to be transmitted over the network then it is important that it is transformed in some way. This can involve a random challenge and the pre-processing discussed above. The result of this transformation would prevent the recovery of the original biometric data. The aim of this transformation is to both prevent replay attacks and to prevent an eavesdropper from obtaining personal information. As an example it would be unacceptable to transmit information about someone's fingerprint over a public network.



6.2.5 Survey of CVMs

This subsection gives an overview of existing methods for cardholder verification. It concentrates on biometric methods. The properties which are mentioned include all of the headlines in the current section.

6.2.5.1 PINs

As already explained, a PIN is the most popular CVM and is in wide use all over the world.

6.2.5.2 Fingerprints

Fingerprints are probably the most accepted means to prove an identity in the public. They have been used for that purpose for 100 years now and are accepted in court. The new way to do it with live-scan systems is to catch an electronic picture of the fingertips and to store it in a template. Typically, the major characteristics as arches, loops and whorls are measured and used for reference. But some new systems are based on the position of sweat pores in the fingerprint area or ultra-sonic images for example. There is an intensive research in this field at the moment and a lot of different methods are being studied. Typical error rates are less than 1 in 100 000.

One of the most promising features of fingerprint systems is that there are already scanners which can be integrated into a smart card: they measure only twenty microns thick and 1.3cm square. This will be a new dimension of security because the measured biometric data never leaves the card.

6.2.5.3 Hand geometry

Another convenient method to verify identities is based on the geometry of the human hand. Characteristics are for example the length of the fingers, palm prints and even the pattern of blood vessels.
An image of the hand is captured using a camera or a laser scanner. This image is transformed in a template of small size which takes a few seconds and is stored in a database or a smart card. The actual verification is faster and can be performed in less than two seconds. Equal error rates of one in one million have been claimed.

6.2.5.4 Eye scanning

Patterns within the eye such as retinal vascular pattern or the iris are as unique as a fingerprint. They also remain stable over a person's life. In order to measure these characteristics, users are required to position their eyes above an eyepiece built into some equipment. The eye is focused and the eye ball is scanned with a mixture of usable wavelength light.
Systems for eye scanning have the best error rates today (approximately 0%) but are also the most expensive and probably the least user friendly.

6.2.5.5 Facial recognition

The characteristics of the bones that build up the skull are also unique. They can be measured with optical devices that generate a plane of infra-red light through which the individual passes. This results in a 3-D facial contour map. Verification takes less than one second. Surprisingly, the person scanned need not know what happened.
For one chance verification, equal error rates of 0.5% have been published.

6.2.5.6 Voice recognition

Voice verification focuses on the characteristics that underlie the production of speech and not on the pronunciation itself. These vocal characteristics cannot be compromised by an impostor who tries to mimic the pronunciation of a word but has different dimensions of the vocal tract, mouth, nasal cavities or other parts of the speech producing elements in the human body.
Today, such systems come up in two different flavours. One way is to identify *who* is speaking but *not what* is spoken. This requires a generic template of a person's voice. The second way is to store predefined words or phrases and to repeat them. Both approaches are quite similar, but the first one is harder to implement and not really available yet. Error rates of less than 0.1% have been achieved with the explicit spoken word approach.
The danger with voice verification lies in replay attacks. Someone could record a phone call, turn his PC on and produce samples of words - that means he will have a dictionary of words spoken by a *real life person.* This is the perfect crime because even the best system has no chance of detecting the attack. The only way to get around with this is to use *random phrases* and generic voice recognition as the basis for access control.

6.2.5.7 Dynamic signature verification

In contrast to a written signature, a dynamic signature describes the manner in which a signature is written. The information which is gathered from the writing process includes factors such as the time to write the signature, the speed at which it is written or the number of times (and times itself) the pen is lifted from the paper.
The problem with this method lies in the fact that some people change their signature every week. Errors rates of around 0.1% are possible today.

6.2.5.8 Selection of a CVM

Probably the best way to find out which CVM fits a given purpose is to match what is necessary with what is available. This can be done in a sequence of steps - the first one is to answer some questions from a *what should be* perspective:

- the total time for enrolling a user
- the total time to verify a user
- the error rates (FAR, FRR, ERR)
- the performance across the population: who is intended to use the card ?
- is the method convenient to use ?
- can everyone use it ?
- what are the additional costs per card ?

The second step is to define preferences or limits and to look at available products to come at a decision.

6.3 CVM in Mobile Telephony

This section is about the use of CVMs in mobile communication. This is one particular application which represents a context in which smart cards are used. The first subsection identifies some general ideas and reflects a user's point of view. The subsequent section defines an architecture how CVMs based on biometrics could be implemented in a GSM environment.

6.3.1 General considerations

What a user expects from a CVM is that it prevents other people from setting up unauthorised calls. That is the standard any CVM is faced with. Whether a particular solution will be a success on the SIM market or not depends on other factors too. Among the most important ones are the acceptance of the methods by nervous users, its feasibility from a technical and economical point of view and of course legal implications. This subsection looks at them in turn.

6.3.1.1 User acceptance

Among the list of CVMs presented before, voice verification is the method of choice from a technical perspective in a telephony context. It uses existing devices in the handset which are able to process speech. Other biometrical methods may be appropriate at a point-of-sale, but they always require an extra sensor outside the card and the terminal to measure the characteristics used to verify the identity of a user. Only the integration of scanners for fingerprints into a smart card could beat this advantage. If technical details were the only important considerations, the discussion would be over. But user acceptance depends on some other factors too. These are for example ease of use, costs or the ability to allow others to use your handset.

In an accident, a policeman may want to make an emergency call on a mobile phone whilst lying in a car. Or a family member may your wife have some urgent need and upset about a barred SIM which they cannot use because their voice has not yet been registered.

This shows that it must be possible to define a list of persons who are allowed to use the SIM for telephony. But a SIM can also offer payment functionality - this is a completely independent application which resides in the same card but has nothing to do with GSM.

Even worse, depending on the environment in which the card is used, different CVMs may be required. With a handset, voice recognition is fine. But at an airport the background noise may prevent it. In this surrounding, fingerprints would be perfectly acceptable because travellers are used to being scanned and controlled. Factors which have to be taken into account are

- which applications are supported by the SIM aside from GSM ?
- what are the likely environments in which an application will be employed ? (airport, gas station)
- which CVMs are best for a particular environment ?

For each CVM supported, one or several templates have to be stored in a card and there must be some way to select them from outside the card. A user may also want to update templates.

The concept which supports these requirements is called *user profiles* and will be explained in subsection 6.3.2.

6.3.1.2 Feasibility

If a card has to store one or several templates, the availability of memory in the SIM may become a major bottleneck. The size of templates is crucial if memory is valuable.

The transmission speed between the card and the terminal also influences the use of a particular CVM. It is not difficult to transmit a PIN, but if a template is so large or complex that it takes 10 seconds to send it to the terminal and vice versa, the CVM is useless.

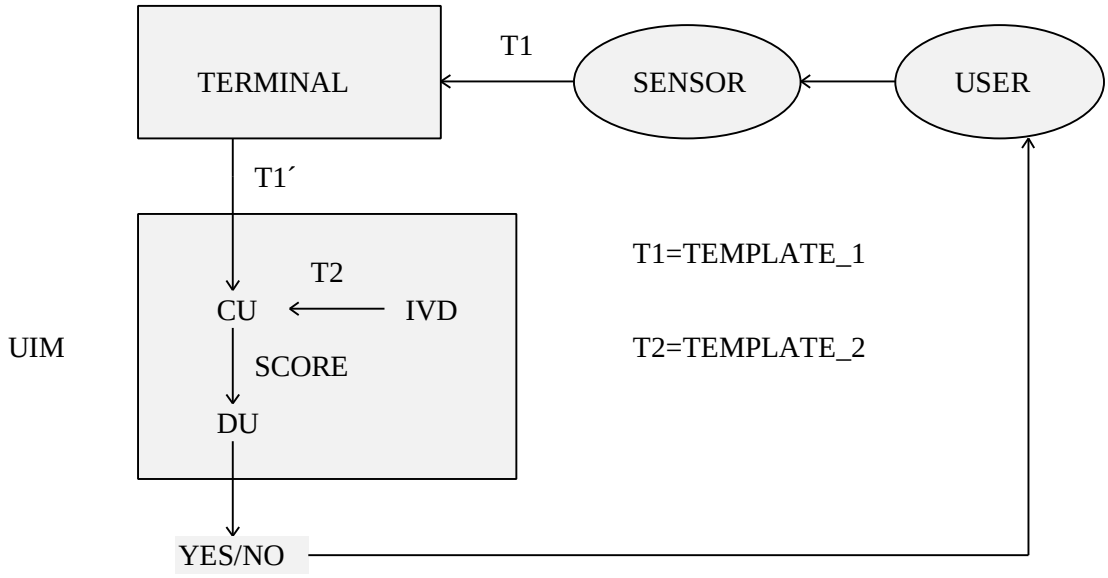
Whether a particular concept can be implemented or not heavily depends on the constraints of the chip that is embedded in the card and the interface between card and terminal.

6.3.1.3 Legal implications

Any technical innovation has to look for support from law professionals to be successful on the long run. This is true for digital signatures as well as for biometric authentication. And it depends on the organisations behind the applications on a card.
For banking and payment cards, a much higher level of security is needed than for entertainment services. The first may require a fingerprint, for the latter a PIN will be sufficient. Which one is suited best must be chosen carefully. And the choice which is eventually made has to respect the law context. Currently, there is not a general framework coming up that defines which CVMs are legally binding and which ones not. A service provider has to make its own choice and require his subscribers to sign a contract which allocates reliabilities to the entities involved in case of a dispute.

6.3.2 An architecture for SIMs

This subsection defines an architecture which describes how biometric user authentication can be implemented in a smart card context. It is a preliminary view and is general enough to be adapted to subsequent changes. The following picture can be used as a reference for the next paragraphs:

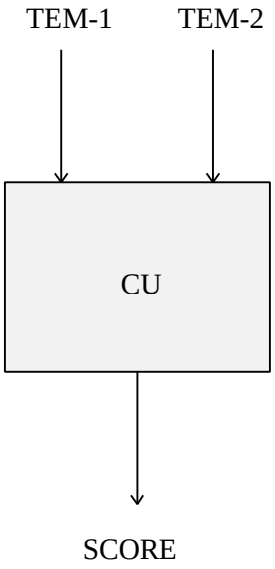


6.3.2.1 User profiles

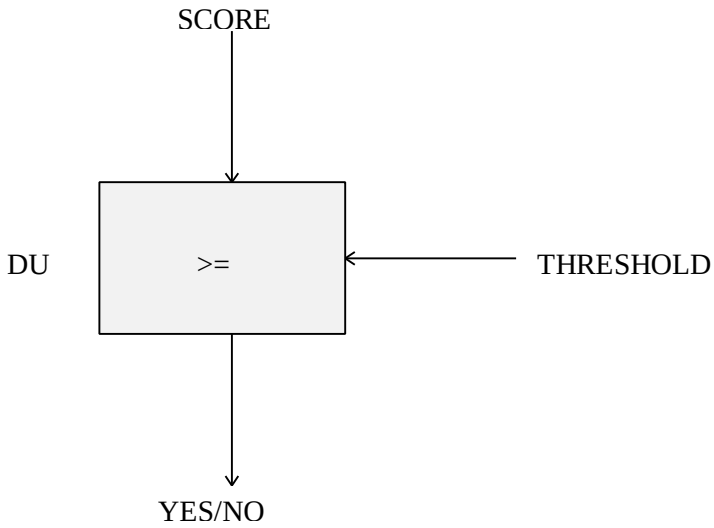
For each user who is allowed to use the card or a particular application within it, a user profile is maintained that describes the respective rights or responsibilities of that user.
The enrolment process allocates an identification vector (IV) to each user which consists of different templates for CVMs. For example, this can be a triplet containing a fingerprint, a PIN and a voice pattern. A list of IVs is stored in an IV database (IVD) inside the card and must be handled as carefully as a secret key.
A user who wants to use the card has to decide which CVM he wants to employ and allow the sensor to measure the corresponding characteristic. The measured data is compared with all the templates previously stored in the IVD until one is found. This procedure is called identification of a user. An alternative is to ask the user for his identity and to look up the corresponding IV. In this case, a user is to be verified.
The profile itself contains only user specific data. This may for example be the right to enrol other users in the IVD or to delete their IVs.

6.3.2.2 Enrolment and verification

In most environments, the sensors which measure some characteristic of a user are independent of the UIM or the terminal. This distinction is of course conceptual - fingerprint readers may be integrated into a smart card or voice recognition can be done in the handset.
The measured characteristic (this is a template) is sent to the terminal which transparently forwards it to the card. The hard part is now to find out whether there is a *match* with some IV stored in the database or not. Provided that the card already knows which kind of template (fingerprint or voice) it receives, this requires it to compare two templates of the same type: one which describes some characteristic measured before at enrolment and one measured now.
The unit inside the UIM which performs this computation is called *comparison unit (CU).* Basically, it works as follows:



The score is some probabilistic value between 0% and 100% and describes the degree of similarity between the two templates. Depending on some threshold value, the *decision unit (DU)* now decides with a *Yes* or *No* answer whether the person in front of the sensors is an authorised one or not.



Similar to a PIN, after some number of consecutive false attempts, the system can store the *wrong* patterns presented and prevent that particular user from further access attempts.