Project Number	AC095
Project Title	ASPeCT: Advanced Security for Personal Communications Technologies
Document Type	Major Deliverable
Security Class	Public

Deliverable Number	D20
Title of Deliverable	D20 - Project final report and results of trials
Nature of Deliverable	Report
Document Reference	AC095/VOD/W31/DS/P/20/E
Contributing Work Packages	WP3.1; WP2.1; WP2.2; WP2.3; WP2.4; WP2.5; WP2.7.
Contractual Date of Delivery	December 1998 (Y04M04)
Actual Date of Delivery	19 January 1999
Editors	Peter Howard and Phil Gosset, Vodafone Ltd

Abstract	This deliverable contains an evaluation of the results of the trials and feedback from the demonstrations. It also contains all the publicly available major results of the project.
Keywords	ACTS ASPeCT Authentication Billing Certificates Cryptography electronic commerce EXODUS fraud detection GSM Integrity Micropayment Migration neural network non-repudiation rule-based security SIM Smart card trial TTP Value-added services UIM UMTS Vocal Biometrics

TABLE OF CONTENTS

Executive Summary **10**

1 Document Control **11**
1.1 Document History *11*
1.2 Abbreviations and Acronyms *11*
1.3 References *15*

2 Introduction **19**

3 Migration towards UMTS security **21**
3.1 Introduction *21*
3.2 Framework for authentication *21*
3.2.1 Objectives of the authentication framework 21
3.2.2 Authentication framework procedures 22
3.2.3 Operational scenarios 22
3.3 Applicability of the authentication framework for a migration scenario *24*
3.3.1 UMTS services 24
3.3.2 Packet based services 25
3.3.3 UMTS network 25
3.4 Authentication demonstration and trial *27*
3.4.1 Authentication demonstration 27
3.4.1.1 Overview 27
3.4.1.2 The UIM realisation 28
3.4.1.3 The network realisation 28
3.4.2 Authentication trial 28
3.5 Evaluation *29*
3.5.1 Evaluation of the authentication demonstration 29
3.5.2 Evaluation of the authentication framework 30
3.5.3 Evaluation of the public key based authentication mechanism 30
3.5.4 Evaluation of the authentication trial 31
3.6 Conclusions *36*

4 UIM security functionality **38**
4.1 Introduction *38*
4.2 Requirements on the UIM in UMTS *38*
4.2.1 The need for a UIM in UMTS 38
4.2.1.1 The GSM SIM 38
4.2.1.2 The UMTS UIM 39
4.3 Demonstration and trial *39*
4.3.1 Overview 39
4.3.2 ASPeCT authentication application 40
4.3.2.1 File system 40
4.3.2.2 Commands 45
4.3.2.3 Elliptic curve implementation 51
4.3.2.4 Memory requirements 54
4.3.2.5 Mapping of commands onto the authentication framework 55
4.3.2.6 EXODUS support 57
4.3.3 Vocal biometric application 57
4.3.3.1 File system 57
4.3.3.2 Commands 58
4.3.4 COLA interface 58
4.3.5 Personalisation 59
4.3.5.1 Fixed personalisation data 59
4.3.5.2 User specific personalisation data 59
4.3.5.3 Personalisation process 59
4.3.6 Evaluation of implementation and trial 59
4.3.6.1 Non byte aligned data formats 60
4.3.6.2 Hash inputs 60
4.3.6.3 Security issues 60

4.4 Conclusion *60*

5 Vocal biometrics in UMTS **62**
5.1 Introduction *62*
5.2 Background and objectives of the work *62*
5.3 Possible approaches to vocal authentication *62*
5.4 An introduction to speech recognition technology *63*
5.5 An introduction to speaker recognition technology. *64*
5.6 Background work at Lernout & Hauspie *65*
5.7 Research work done within the scope of the ASPeCT project *65*
5.8 Results from the public demonstration *69*
5.9 Extended study: Feasibility study of an algorithmic split *70*
5.9.1 Security aspect 70
5.9.2 Overview of current algorithm 70
5.9.3 Specification of current and future Smart card constraints 71
5.9.4 Split of current algorithm 71
5.9.5 Proposal alternative approach 72
5.9.6 Conclusion algorithmic split 73
5.9.7 Additional references 73

6 The use of trusted third parties in UMTS **74**
6.1 Introduction *74*
6.2 Requirements for TTPs in UMTS *74*
6.3 TTP infrastructure design *76*
6.3.1 Certificate design and format 76
6.3.1.1 Certificate applications 76
6.3.1.2 Types of Certificate 77
6.3.1.3 Certificate Information Sequence Format 78
6.3.1.4 Signature Mechanism 79
6.3.1.5 Cross-certificates 81
6.3.2 Certificate management issues 82
6.3.2.1 Definition of certificate chains 82
6.3.2.2 Alternative TTP scenarios 83
6.3.3 TTP Software architecture 85
6.3.3.1 Overview of software architecture 85
6.3.3.2 Architecture of TTP server software 86
6.3.3.3 Architecture of TTP client software 86
6.4 Key management for encryption *87*
6.4.1 Lawful interception in UMTS 87
6.4.2 Framework for evaluating key escrow schemes 87
6.4.2.1 Escrow parameters 88
6.4.2.2 Escrow requirements 88
6.4.2.3 Desirable parameters for UMTS 89
6.4.3 The basic JMW protocol 89
6.4.3.1 Detailed protocol description 90
6.4.3.2 Interception options 92
6.4.3.3 Evaluation of protocol against parameters and requirements 93
6.4.4 Protocol variants 93
6.4.4.1 Time-bounded interceptions 94
6.4.4.2 Two-way communication 95
6.4.4.3 Escrow in multiple domains 95
6.4.4.4 Split escrow 96
6.4.4.5 Increased cryptographic flexibility 96
6.4.5 Alternative schemes 96
6.4.5.1 LWY protocol 96
6.4.5.2 IBM protocol (SecureWay) 97
6.4.5.3 VKT protocol (Binding cryptography) 98
6.4.5.4 Parameters of alternative protocols 99
6.4.5.5 Evaluation of protocols against escrow requirements 101
6.4.5.6 Comparative evaluation of UMTS escrow protocols 102

6.5 Demonstration of key management for encryption *102*
6.5.1 Overview of demonstration architecture 102
6.5.2 Demonstration configuration 104
6.6 Evaluation of demonstration *107*
6.6.1 Technical feasibility 107
6.6.1.1 Estimates of processing delays 108
6.6.1.2 Analysis of processing delays 110
6.6.2 User acceptability 111
6.6.3 Summary of suggested enhancements 111
6.7 Conclusions *112*

7 Security and integrity of billing in UMTS **113**
7.1 Introduction *113*
7.2 Secure billing for value-added information services *113*
7.2.1 A payment model for value-added information services provision 114
7.2.2 Specific requirements on the charging scheme 114
7.2.3 Review of micropayment approach 116
7.2.4 Protocol design 116
7.2.4.1 Structure of charging process 117
7.2.4.2 Authentication and initialisation phase 117
7.2.4.3 Data transfer phase 121
7.2.4.4 Variants and references 122
7.2.5 Theoretical evaluation and alternative schemes 123
7.2.5.1 Theoretical evaluation work 123
7.2.5.2 Possible protocol attacks 123
7.2.5.3 Alternative protocols 126
7.3 Demonstrations and trials of secure billing for value added information services *130*
7.3.1 Overview 130
7.3.2 First demonstrator 130
7.3.2.1 Description of demonstration 130
7.3.2.2 Architecture 132
7.3.3 Second demonstrator 134
7.3.3.1 Description of demonstration 134
7.3.3.2 Protocol architecture and software configuration 134
7.3.3.3 Hardware configuration 138
7.3.4 Field trial 138
7.3.4.1 Introduction 138
7.3.4.2 Overview and background 139
7.3.4.3 Trial configuration 140
7.3.4.4 Trial scenario 140
7.3.4.5 Graphical user interface 141
7.3.4.6 Trial users 142
7.3.4.7 Support 143
7.3.4.8 Realisation vs. original plans 143
7.4 Evaluation *143*
7.4.1 Evaluation of demonstrator 143
7.4.1.1 First demo 143
7.4.1.2 Second demonstrator 143
7.4.2 Evaluation of field trial 144
7.4.2.1 Technical feasibility 144
7.4.2.2 User acceptability 150
7.4.2.3 Discussion on trial results 152
7.5 Conclusions *153*

8 Detection and management of fraud in UMTS networks **154**
8.1 Introduction *154*
8.1.1 Scope 154
8.1.2 Purpose 154
8.2 Fraud scenarios *155*
8.2.1 Non pre-payment fraud scenarios 155

8.2.1.1 Subscription fraud 155
8.2.1.2 PABX fraud 155
8.2.1.3 Freephone fraud 155
8.2.1.4 Call-back fraud 155
8.2.1.5 Premium rate fraud 155
8.2.1.6 Fax-back and malicious call-back fraud 155
8.2.1.7 Technical internal fraud 156
8.2.1.8 Mobile to mobile fraud 156
8.2.1.9 Roaming fraud 156
8.2.1.10 Tumbling fraud 156
8.2.1.11 Hijacking 156
8.2.1.12 Handset theft 156
8.2.2 New prepay fraud scenarios 156
8.2.2.1 Cheque fraud 156
8.2.2.2 Credit card fraud 157
8.2.2.3 Voucher theft 157
8.2.2.4 Voucher ID duplication 157
8.2.2.5 Faulty vouchers 157
8.2.2.6 Network access fraud 157
8.2.2.7 Network attack 157
8.2.2.8 Long duration calls 157
8.2.2.9 Handset theft 157
8.2.2.10 SMS abuse 157
8.2.2.11 Roaming fraud 157
8.3 Requirements for fraud detection *157*
8.3.1 Introduction 157
8.3.2 The fraud detection environment 158
8.3.3 Functional requirements 158
8.3.3.1 TT data feed 158
8.3.3.2 Alarm processes 158
8.3.3.3 Fraud boundaries 158
8.3.3.4 Outcome visibility 159
8.3.3.5 NO–SP co-operation 159
8.3.3.6 Fraud prediction 159
8.3.4 System requirements 159
8.3.4.1 Performance 159
8.3.4.2 Customisation 159
8.3.4.3 Scalability-flexibility 159
8.4 Available technologies *159*
8.4.1 Geographical information 160
8.4.2 Database management systems 160
8.4.3 Unsupervised profiling 160
8.4.4 Flexibility 160
8.5 ASPeCT approach *161*
8.5.1 Introduction 161
8.5.1.1 User profiling 161
8.5.2 Rule based 163
8.5.2.1 User profiling 163
8.5.2.2 Rule-based fraud analysis 165
8.5.2.3 The administration GUI 167
8.5.3 Supervised neural network 169
8.5.3.1 Profiling 169
8.5.3.2 Feature extraction 171
8.5.3.3 Storing user profiles 172
8.5.3.4 Supervised learning 172
8.5.4 Unsupervised learning tool 174
8.5.4.1 Prototyping 174
8.5.4.2 Constructing a statistical user profile for each user 177
8.5.4.3 Maintaining current and history profiles as probability distributions 177

8.5.4.4 The fraud engine 178
8.5.5 B – Number analysis tool 179
8.5.5.1 Introduction 179
8.5.5.2 Adding tags to toll ticket fields 179
8.5.5.3 Dividing the world into fraud risk categories 179
8.5.5.4 B-number profiling 180
8.5.5.5 The fraud engine 180
8.5.6 Brutus 181
8.5.6.1 Monitoring and Graphical User Interface 185
8.5.6.2 Implementation of the GUI 187
8.6 Description of trials *188*
8.6.1 Data sets 188
8.6.2 Vodafone data 188
8.6.3 Combination of the different tools 188
8.7 Results of trials *189*
8.7.1 Evaluation of technical effectiveness 189
8.7.2 Technical description of the data 189
8.7.3 Performance of the individual tools and integration 190
8.7.3.1 Comparison of alarm histograms and ROC curves 190
Integration of tools 195
8.7.3.3 Comparison via area under the ROC curve 195
8.7.4 Evaluation of user acceptability 196
8.8 Conclusions *202*

9 Overall project conclusions **203**

TABLE OF FIGURES

Figure 3.1 - Operational scenario for 'User not registered, no roaming agreement' 23
Figure 3.2 - Physical Architecture....27
Figure 3.3 - ASPeCT - EXODUS trial configuration....29
Figure 3.4 - New Siemens Authentication - EXODUS timings....32
Figure 3.5 - Current Siemens Authentication - EXODUS timings....33
Figure 3.6 - New Siemens Authentication - Demo timings....35
Figure 3.7 - Current Siemens Authentication - Demo timings....35
Figure 4.1 - Mapping of Authentication Framework to Commands....56
Figure 5.1 - Functional Diagram of the Implemented Single Password System....67
Figure 5.2 - Fuzzy Trial System....68
Figure 5.3 - Dual Name/Password System....69
Figure 5.4 - Terminal/Card Functional Split....70
Figure 5.5 - Alternative Terminal/Card functionality split....72
Figure 6.1 - UMTS role model....75
Figure 6.2 - Overview of a TTP server and its client....85
Figure 6.3 - The JMWprotocol as implemented in the Demonstrator....90
Figure 6.4 - The logical and physical connections between the entities involved in the demonstration....102
Figure 6.5 - External communication interfaces between the entities....103
Figure 6.6 – Message exchanges in demonstrator....104
Figure 6.7 - Sender A - sequence of events....105
Figure 6.8 - TTP TA - sequence of events....106
Figure 6.9 - TTP TB - sequence of events....107
Figure 6.10 - Receiver B - sequence of events....107
Figure 6.11 - Measurement points for Part 2....108
Figure 6.12 - Measurement points for Parts 3 and 4....109
Figure 7.1 - Charging model....114
Figure 7.2 - Authentication and initialisation of payment protocol....119
Figure 7.3 - Charge ticks protocol....121
Figure 7.4 - Protocol stack....132
Figure 7.5 - Physical configuration of version 1....133
Figure 7.6 - Physical configuration of version 2....133
Figure 7.7 - Global structure of WWW service....135
Figure 7.8 - Global structure with intermediate layers....135

Figure 7.9 - Global structure with FSMs....136
Figure 7.10 - Global structure with TTP....137
Figure 7.11 - Configuration of second demonstrator....138
Figure 7.12 - Authentication and Initialisation of Payment Protocol....145
Figure 7.13 - Charge Ticks Protocol....146
Figure 7.14 - Re-initialisation of Payment Protocol....147
Figure 8.1 - Cost Analysis of Fraud Tools....154
Figure 8.2 - Architecture of the rule-based part within the integrated prototype....166
Figure 8.3 - Administration GUI of the Rule-based Fraud Detection Component....169
Figure 8.4 - Sigmoidal neuron and multilayer perceptron architecture....173
Figure 8.5 - 50 Prototypes for national calls....176
Figure 8.6 - 50,000 National calls to Neural Network prototyper....176
Figure 8.7 - 50 Prototypes for international calls....177
Figure 8.8 - 50,000 International calls to Neural Network prototyper....177
Figure 8.9 - A CUP and an UPH of a subscriber exhibiting acceptable behaviour....179
Figure 8.10 - A CUP and an UPH of a subscriber who raised an alarm.....179
Figure 8.11 - Number of calls made to each class over the sequence of international Toll Tickets.....181
Figure 8.12 - Trial Architecture....182
Figure 8.13 - Monitoring within the First Demonstrator....185
Figure 8.14 - Monitoring within the Integrated Fraud System....186
Figure 8.15 - Screenshot of the graphical user interface of the integrated fraud detection system.....187
Figure 8.16 - The results of the B-number analysis tool:....191
Figure 8.17 - The results of the unsupervised neural network tool:....192
Figure 8.18 - The results of the supervised neural network tool:....193
Figure 8.19 - The results of the rule based tool: 194
Figure 8.20 - The results of the integrated tool:....195
Figure 8.21 - Fraudster F234154858596f69281f6104. 198
Figure 8.22 - Fraudster F234154828265901382e7f56.....199
Figure 8.23 - Fraudster F23415526754463d06522b66.....200
Figure 8.24 - Fraudster F2341548203340076a532f3d.....201

TABLE OF TABLES

Table 3.1 - Siemens New Registration Times 34
Table 3.2 - Siemens Current Registration Times 34
Table 3.3 - Standard Deviation of New Registration Times 34
Table 3.4 - Standard Deviation of Current Registration Times 34
Table 4.1 - File system for the UIM 41
Table 4.2 - Implementation Size of UIM Components 55
Table 4.3 - Data Support for EXODUS in the UIM 57
Table 5.1 - System Overview 65
Table 6.1 - A certificate information sequence format 79
Table 6.2 - A recoverable string S_r 80
Table 6.3 - Certificate format based on RSA-signature 80
Table 6.4 - Certificate format based on AMV-signature 81
Table 6.5 - Measured delays during Part 2 108
Table 6.6 - Delay at A during Parts 3 and 4 109
Table 6.7 - Delay at TB during Parts 3 and 4 109
Table 6.8 - Delay at B during Parts 3 and 4 110
Table 6.9 - Total delay at B and TB during Parts 3 and 4 110
Table 6.10 - Total processing delays at each entity during Parts 1 to 4 110
Table 7.1 - Timings of the protocol sessions 147
Table 7.2 - Detailed description of the timings of the three protocols 148
Table 7.3 - Cryptographic functions timings 150
Table 8.1 - Performance of the different modules and the integrated tools on the two subsets of the data. 195
Table 8.2 - List of 27 suspicious users with their classification as probably fraudulent or not 196
Table 8.3 - Fraudulent users with the date of first call in the data, date of first alarm, and date of barring 197

Executive Summary

This document is the final deliverable of the ACTS ASPeCT project, and contains an evaluation of the trials, and feedback from the demonstrators. It also contains all of the publicly available major results of the project.

The deliverable is divided into the main work package areas of the project.

The second chapter contains a brief introduction to the project.

The third chapter contains the results of the work on Migration towards UMTS Security. This includes the definition of the authentication framework that allows a negotiation to take place to establish which authentication protocol should be used. In addition, the applicability of this framework to a migration scenario is discussed against UMTS services, packet based services and a UMTS network. Both the demonstrator and trial of the authentication framework are then discussed, and the results evaluated.

The fourth chapter contains the results of the work on UIM Security functionality. It discusses the requirements for a UIM, and compares these requirements to those of the GSM SIM. The demonstrator and trial are then discussed, including the implementation, and the support of the vocal biometric application. The results of the trial are evaluated along with the chosen implementation, and conclusions drawn.

The fifth chapter contains the results of the work on Vocal Biometrics. The background to the creation of a public demonstrator is discussed, along with an evaluation of the results, and a discussion of future possible directions.

The sixth chapter contains the results of the work on trusted third parties in UMTS. It begins with a discussion on requirements for TTPs in UMTS, TTP infrastructure design including certificate design, management and TTP software architecture, and key management for encryption. It discusses the use of the JMW protocol within a mobile environment, its variants, and alternative schemes. The demonstrator and trial are then discussed, and evaluated. Finally, possible future directions are discussed.

The seventh chapter contains the results of the work on Security and integrity of billing in UMTS. It begins by discussing the adopted micropayment approach to billing for value-added services, and outlines the protocol design. Several alternative approaches are also discussed, and an analysis of authentication and key agreement protocols for mobile systems is included. The demonstrator and trials are then discussed and evaluated.

The eighth chapter contains the results of the work on the detection and management of fraud in UMTS networks. It begins by discussing various fraud scenarios, the requirements for fraud detection, the available technologies, and the approach taken. The use of rule based, and both supervised and unsupervised neural network technologies is described, both in terms of single implementations and a combined implementation, BRUTUS. Finally the trial is described, the results evaluated, and conclusions drawn.

The ninth chapter contains the overall project conclusions.

1 Document Control

1.1 Document History

Version	Date	
Draft A	13-Oct-1998	Skeleton document.
Draft B	01-Dec-1998	Partially completed document released for comment.
Draft C	05-Dec-1998	Nearly completed document released to Keith Howker for comment.
Draft D	14-Dec-1998	First complete draft released for comment.
Draft E	18-Dec-1998	Final draft released to PMC for approval.

1.2 Abbreviations and Acronyms

ACRYL	Advanced Cryptographic Library (by Siemens)
ACTS	Advanced Communications Technologies and Services
AMV	Agnew-Mullin-Vanstone (equation)
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation (version 1)
ASPeCT	Advanced Security for Personal Communications Technologies
ATM	Asynchronous Transfer Mode
ATT	Attributed Toll Tickets
AuC	Authentication Centre
BALM	B-number analysis AlarM
BRUTUS	B-number and RUle-based analysis of Toll tickets utilizing Unsupervised and Supervised neural network technologies
CA	Certification Authority
CBC	Cipher Block Chaining mode
COLA	COnversion Layer
CoLa	Conversion Layer
CPN	Calling Party Number
CRL	Certificate Revocation List
CSA	Certificate Service Application
CUP	Current User Profile
CUSF	Call Unrelated Service Function
DECT	Digital Enhanced Cordless Telecommunications
DES	Data Encryption Standard
DIB	Directory Information Base
DIT	Directory Information Tree

DLL	Dynamic Link Library
EEPROM	Electrically Erasable Programmable Read Only Memory
EMV	Europay, Mastercard and Visa
ETSI	European Telecommunications Standards Institute
EXODUS	EXperiments On the Deployment of UMTS
FP	Fixed Part (DECT)
FPAI	Fixed Point of Attachment Identifier
FSM	Finite State Machine
GSM	Global System for Mobile communications
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HW	Hardware
IEC	International Electrotechnical Commission
IMSI	International Mobile Subscriber Identity
IMUI	International Mobile User Identity
IMUN	International Mobile User Number
IN	Intelligent Network
INAP	Intelligent Network Application Protocol
IPF	Internal Public key File
ISF	Internal Secret key File
ISO	International Standards Organisation
IV	Initialisation Vector
IWU	Interworking unit
LAI	Local Area Identifier
MAC	Message Authentication Code
MB	Mobile Broadband
NN	Neural Network
NNI	Network Node Interface
NO	Network Operator
OFB	Output FeedBack mode
PABX	Private Automatic Branch Exchange
PDAT	Protocol Data Analysis Tool
PDU	Protocol Data Unit
PIN	Personal Identification Number
PP	Portable Part (DECT)

RACE	Research and development in Advanced Communication technologies in Europe
RIPEMD	RACE Integrity Primitives Evaluation Message Digest
RNG	Random Number Generator
ROM	Read Only Memory
RSA	Rivest, Shamir and Adleman (public key algorithm)
SALR	Supervised NN ALaRm
SALV	Supervised NN ALarm leVel
SCF	Service Control Function
SCP	Service Control Point
SIM	Subscriber Identity Module
SLP	Service Logic Program
SMG	Special Mobile Group
SP	Service Provider
SPI	Service Provider Identity
SSP	Service Switching Point
SW	Software
TBNB	Tag for B-NumBer
TBTP	Tag for B TyPe of number
TBZC	Toll ticket B-number Zone Code
TCDR	Tag for Chargeable DuRation
TCP/IP	Transmission Control Protocol / Internet Protocol
TCSD	Tag for Charge Start Date
TCST	Tag for Charge Start Time
TMUI	Temporary Mobile User Identity
TSDN	Toll ticket Starting Date Normalised
TSTS	Toll ticket Starting Time in Seconds
TT	Toll Ticket
TTP	Trusted Third Party
UALM	Unsupervised NN ALarM
UIM	User Identity Module
UMTS	Universal Mobile Telecommunications System
UNI	User Network Interface
UPH	User Profile History
UPR	User Profile Record
USIM	UMTS SIM
UTC	Universal Time Coordinates

VAS	Value Added Service
VASP	Value Added Service Provider
VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier
Winsocks	Windows Sockets
WWW	World Wide Web

1.3 References

[AAB97] H. Abelson, R. Anderson, S.M. Bellovin, J. Benaloh, M. Blaze, W. Diffie, J. Gilmore, P.G. Neumann, R.L. Rivest, J.I. Schiller and B. Schneier. "The risks of key recovery, key escrow, and trusted third-party encryption", May 1997. Available at http://www.crypto.com

[AMS96] R. Anderson, H. Manifavas and C. Sutherland. "A practical electronic cash system". Available at http://www.cl.cam.ac.uk/users/rja14/

[AR97] R. J. Anderson and M. Roe. "The GCHQ protocol and its problems". Advances in Cryptology - EUROCRYPT '97, Lecture Notes in Comput. Sci. 1233:134-148, 1997.

[BJM97] S. Blake-Wilson, D. Johnson and A. Menezes. "Key agreement protocols and their security analysis". Cryptography and Coding, Lecture Notes in Computer Science 1355, pp. 30-45, 1997.

[BPV97] J. Borst, B. Preneel and J. Vandewalle. "On the time-memory trade-off between exhaustive key search and table precomputation". Proceedings of the Nineteenth Symposium on Information Theory in the Benelux, Veldhoven, The Netherlands (1998) 111 - 118.

[CGM96] L. Chen, D. Gollman and C.J. Mitchell. "Key escrow in mutually mistrusting domains", Proceedings of 1996 Cambridge Workshop on Security Protocols, Lecture Notes in Comput. Sci., 1189:139-153, 1996.

[CM96] L Chen and C J Mitchell, "Key escrow in multiple domains". Pre-print, 1996.

[D02] ACTS AC095, ASPeCT Deliverable D02, Initial report on security requirements, February 1996.

[D05] ACTS AC095, ASPeCT Deliverable D05, Migration scenarios , Issue A.1, August 1996.

[D09] ACTS AC095, ASPeCT Deliverable D09, Trusted third parties: First implementation, March 1997.

[D10] ACTS AC095, ASPeCT Deliverable D10, Secure billing: First implementation, February 1997.

[D12] ACTS AC095, ASPeCT Deliverable D12, Migration scenarios: first implementation, October 1997.

[D14] ACTS AC095, ASPeCT Deliverable D14, Trusted third parties: evaluation report, June 1997.

[D16] ACTS AC095, ASPeCT Deliverable D16, Secure billing: Evaluation report, June 1997.

[D17] ACTS AC095, ASPeCT Deliverable D17, Migration scenarios: final version, Issue 1, October 1997.

[D19] ACTS AC095, ASPeCT Deliverable D19, Report on final trial and demonstration, Issue 1, April 1998.

[D21] ACTS AC095, ASPeCT Deliverable D21, Single Password Verification, June 1997.

[D22] ACTS AC095, ASPeCT Deliverable D22, Multiple Password Verification.

[D23] ACTS AC095, ASPeCT Deliverable D23, Low Storage Speaker Verification Demo.

[D24] ACTS AC095, ASPeCT Deliverable D23, Vocal Password-Based User Authentication Report.

[D95] D.E. Denning. "Critical factors of key escrow encryption systems", Proceedings of the 18th National Information Systems Conference, 10-13 October 1995, Baltimore, Maryland, pp384-394.

[DB96] D.E. Denning and D.K. Branstad. "A taxonomy for key escrow encryption systems", Communications of the ACM, 39(3):33-40, 1996.

[DBP96] H. Dobbertin, A. Bosselaers, B. Preneel. "RIPEMD-160: a strengthened version of RIPEMD", Fast Software Encryption, Third International Workshop. Lecture Notes in Comput. Sci., 1039:71-82, 1996.

[DH76] W. Diffie and M. Hellman. "New directions in cryptography", IEEE Transactions on Information Theory, 22:644-654, 1976.

[ElG85] T.ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms". IEEE Transactions on Information Theory, 31:469-472, 1985.

[EMV96] EMV'96: Integrated Circuit Card Specification for Payment Systems, Version 3.1.1, May 31, 1998. Available at http://www.visa.com/cgi-bin/vee/nt/chip/download.html

[ETS23.01] ETSI ETS draft UMTS 23.01, Special Mobile Group (SMG): Universal Mobile Telecommunications System (UMTS): General UMTS Architecture, Version 0.2.0, November 1997

[ETS33.21] ETSI draft Technical Specification UMTS 33.21. Special Mobile Group (SMG): Universal Mobile Telecommunications System (UMTS); Security Requirements, Version 1.0.4, November 1998.

[ETS97a] ETSI Draft prEG 201 057, Telecommunications Security; Trusted Third Parties (TTPs); Requirements for TTP services, Edition 1.1.1, May 1997.

[ETSI050103] ETSI/SMG/ETR 050103, System requirements, version 3.1.0, SMG5 TD276/95.

[GCS-API] X/Open CAE Preliminary specification, generic cryptographic service API (GCS-API) base – draft 3 April 1995.

[GSS-API] X/Open CAE Specification, generic security service API (GSS-API) base – December 1995.

[G89] I. Grabec, "The self organisation of neurons described by the second maximal entropy principle" Proc. 1st IEE ICANN 1989 No. 313 pp12-16

[H80] M. Hellman. "A cryptanalytic time-memory tradeoff", IEEE Transactions on Information Theory, Vol. 26 (1980) 401 - 406.

[HHM98] G. Horn, P. Howard, K.M. Martin, C.J. Mitchell, B. Preneel, K. Rantos. "Trialling secure billing with trusted third party support for UMTS applications". 5th International Conference in Services and Networks, Proceedings of 3rd ACTS Mobile Communications Summit (1998) 574 - 579.

[HM98] M.P. Hoyle and C.J. Mitchell. "On solutions top the key escrow problem", State of the art and evolution on computer security and industrial cryptography, Lecture Notes in Comput. Sci., To appear.

[HMM98] G. Horn, K.M. Martin and C.J. Mitchell. "Evaluation of authentication protocols for mobile environment value-added services". K.U. Leuven, Technical Report, 1998.

[HP98] G. Horn and B. Preneel. "Authentication and payment in future mobile systems". Computer Security - ESORICS 98, Lecture Notes in Computer Science, 1485 (1998) 277 - 293.

[HSW96] R.Hauser, M. Steiner and M. Waidner. "Micro-payments based on iKP". Presented at SECURICOM 96. Available from http://www.zurich.ibm.com

[IBM97] IBM Secure Way Key Recovery Technology. Available at ftp://service2.boulder.com/software/icserver/doc/keyrec.pdf

[ISO10181-1] ISO/IEC 10181-1, Information Technology - Security frameworks in open systems - Part 1: Frameworks overview.

[ISO14888-3] ISO/IEC 2nd CD 14888-3. Information technology - Security techniques - Digital signature with appendix - Part 3: Certificate-based mechanisms, June 1996.

[ISO7816-3] ISO/IEC 7816-3:1997. Information technology – Identification cards – Integrated circuit(s) cards with contacts –Part 3: Electronic signals and transmission protocols.

[ISO7816-4] ISO/IEC 7816-4:1995. Information technology – Identification cards – Integrated circuit(s) cards with contacts –Part 4: Interindustry commands for interchange.

[ISO9796-2] ISO/IEC 9796-2 (review). Information technology - Security techniques - Digital signature techniques giving message recovery - Part 2: Mechanisms using a hash function, June 1996.

[ISO9797] ISO/IEC 9797:1994. Information technology – Security techniques – Data integrity mechanism using a cryptographic check function employing a block cipher algorithm.

[X.509] ISO/IEC 9594-8 ITU-T Rec X.509, Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1994

[JMW96a] N. Jefferies, C. Mitchell, M. Walker. "A proposed architecture for trusted third party services", Cryptography : Policy and Algorithms, Lecture Notes in Comput. Sci., 1029:98-104, 1996.

[JMW96b] N. Jefferies, C. Mitchell, M. Walker. "Combining TTP-based key management with key escrow", Royal Holloway Computer Science Department Technical Report CSD-TR-96-10, 1996.

[JY96] C.S. Jutla and M.Yung. "Paytree: "Amortised-signature" for flexible micropayments". Proceedings of Second USENIX Association Workshop on Electronic Commerce, November 1996, pp213-221.

[KM97] L.R. Knudsen and K.M. Martin. "In search of multiple domain key recovery". K.U. Leuven Technical Report, June 1997.

[KMP98] L.R. Knudsen, K.M. Martin and B. Preneel. "One-way functions for chain-based micropayments". K.U. Leuven Technical Report, October 1998.

[KP96] L. R. Knudsen and T. P. Pederson. "On the difficulty of software key escrow", Advances in Cryptology - EUROCRYPT '96, Lecture Notes in Comput. Sci. 1070:237-244, 1996.

[L81] L. Lamport. "Password authentication with insecure communication". Communications of the ACM, 24 (1981), pp770-772.

[Lap97a] Martine Lapere & Eric Johnson, "User Authentication in Mobile Telecommunication Environment Using Voice Biometrics and Smartcards", 4th International Conference on Intelligence in Services and Networks, IS&N '97, Lecture Notes in Computer Science, 1238 (1997) 437-444.

[LWY95] A.K. Lenstra, P. Winkler and Y. Yacobi. "A key escrow system with warranted bounds", Advances in Cryptology - CRYPTO '95, lecture Notes in Comput. Sci. 963:197-207, 1995.

[M63] D. Marquardt. “An algorithm for least squares estimation of non-linear parameters Journal of the Society of Industrial and Applied Mathematics Vol 11 No.2 pp 431-441

[M90] R.C. Merkle. "A certified digital signature". Proceedings of CRYPTO '89, Lecture Notes in Computer Science, 435, pp218-238, 1990.

[M97] K.M. Martin. "Increasing efficiency of international key escrow in mutually mistrusting domains", Cryptography and Coding, Lecture Notes in Comput. Sci. 1355: 221-232, 1997.

[M98] K.M. Martin. "Applying cryptography within the ASPeCT project". Information Security Technical Report, 2 (4) pp.41-53, 1997.

[MOV97] A. Menezes, P. van Oorschot and S. Vanstone. "Handbook of Applied Cryptography", CRC Press, Boca Raton (1997).

[MPM98] K.M. Martin, B. Preneel, C. Mitchell, H.J. Hitz, G. Horn, A. Poliakova and P. Howard.

"Secure Billing for Mobile Information Services in UMTS". 5th International Conference in Services and Networks, IS&N '98, Lecture Notes in Computer Science, 1430 (1998) 535-548.

[MR98] C. J. Mitchell, K. Rantos. "Key recovery in ASPeCT authentication and initialization of payment protocols". For submission.

[P95] T. P. Pedersen. "Electronic payments of small amounts". DAIMI PB-495, Computer Science Department, Aarhus University, August 1995.

[PKCS#1] RSA Laboratories, "The Public Key Cryptography Standards – PKCS#1 RSA Encryption Standard", Version 1.5, November 1, 1993.

[RS96] R.L. Rivest and A. Shamir. "PayWord and MicroMint: Two simple micropayment schemes". Cryptobytes Vol 2, No 1, pp7-11, May 1996. Extended version also available from http://theory.lcs.mit.edu/~rivest

[UKC96] UK CESG. "Securing Electronic Mail within HMG - Part 1: Infrastructure and Protocol", Draft C T/3113/TL/2776/11, 21 March 1996.

[VKT97] E.R. Verheul, B. Koops, H.C.A. van Tilborg. "Binding cryptography - a fraud-detectable alternative to key-escrow proposals", Computer Law and Security Report, vol. 13, No 1, 1997.

[VT97] E.R. Verheul and H.C.A. van Tilborg. "Binding ElGamal: A Fraud -Detectable Alternative to Key-Escrow Protocols". Advances in Cryptology - EUROCRYPT '97, Lecture Notes in Comput. Sci. 1233:119-133, 1997.

[W80] H.C. Williams. "A modification of RSA public-key encryption". IEEE Transactions on Information Theory 26, pp. 726-729, 1980.

2 Introduction

This chapter consists of a brief introduction to the ACTS ASPeCT project.

The mobile telecommunications world is undergoing a continuing transformation as increasing numbers of services are being offered to a growing number of users by more and more operators. It is essential for the continuing success of this process that the evolving security requirements of users and service providers are addressed in an appropriate and timely way. The aim of this project was to ensure that this could happen by implementing and running trials of advanced security features to prove their feasibility and acceptability. The project thus contributed to Task AC415 of the ACTS work plan.

This motivated the general objective of the project, which is summarised as follows:

- to study the feasibility and acceptability of new and advanced security features in existing and future personal communication networks, based on trials and demonstrations.

In addition to the above, the detailed objectives of this project are also summarised as follows:

- to investigate, implement and test in trials solutions in the following areas:
    - migration from existing mobile telecommunications systems to UMTS;
    - fraud detection in UMTS;
    - Trusted Third parties for end-to-end services in UMTS;
    - capabilities of future UIMs;
    - security and integrity of billing in UMTS.

For obvious reasons of practicality and limited resources, the project could not encompass all envisaged communications networks. Therefore the project focused on the security aspects related to a particular system, namely UMTS as being standardised by ETSI. The results of the project provided useful input into this standardisation process throughout the project's lifetime.

Existing personal communications networks were taken into account solely from the point of view of migration towards UMTS. Concrete solutions were developed in this project for UMTS only. However, the application of some of the results to other systems is also possible.

Furthermore, in order to be able to conduct this study in any depth and concreteness, the project concentrated its efforts on a subset of the security features in UMTS that are new and advanced, in the sense that they have no analogue in existing systems and are expected to be crucial for the success of UMTS.

Existing mobile and cordless telecommunications networks, such as those conforming to the ETSI GSM and DECT standards, already provide certain essential security features, primarily designed to address the vulnerability of the air interface.

One of the most fundamental problems that will be faced by operators of existing networks will be to evolve the current security features towards the more sophisticated and all-embracing set of security features likely to be necessary for UMTS/3GPP networks. This will, in particular, raise questions about the possible functionality that can be provided by existing and future User Identity Modules (UIMs) - the smart cards for access to UMTS/3GPP networks. A further problem is posed by the ever-present threat of fraud, which the DECT and GSM security features have only partly addressed. New techniques for detecting fraudulent behaviour will need to be brought into use, following the lead currently being set by large financial organisations. Totally new, and in particular end-to-end, security features will be required to address many new application domains likely to be found for mobile telecommunications networks; the possible ramifications for European business are immense, and the likely end-to-end security requirements are wide-ranging. Finally, it can be observed that disputes about bills, often connected with cases of fraud, have increasingly become a problem for users, service providers and network operators with serious economic and legal implications for all parties involved. Therefore, features guaranteeing the security and integrity of billing that help to settle these disputes are urgently needed. Although the issue of the security and integrity

of billing is of concern to many other systems besides UMTS, UMTS is seen to be a good candidate to implement corresponding measures as the users are equipped with a security module (the UIM) and the design of UMTS can still be influenced.

All of these major security issues were addressed by the ASPeCT project. A migration demonstration was produced which showed how UMTS security features could be migrated using an authentication framework. A variety of techniques for fraud detection were applied in an experimental way to ‘real’ network data, derived from operational networks, and trials of a fraud detection tool took place. Future end-to-end security requirements were analysed, and trials took place of new security features to meet these requirements. The functional requirements for future UIMs were investigated, and security capabilities expected to be necessary in future networks were demonstrated. Finally, the security and integrity of billing was addressed. Corresponding new security features were investigated and solutions implemented.

3 Migration towards UMTS security

3.1 Introduction

This Chapter discusses the three main activities of the ASPeCT security migration work:

- Applicability of the authentication framework as a basis for a migration scenario
- Design and implementation of the security migration demonstrator
- Specification and implementation of the joint EXODUS-ASPeCT trial

3.2 Framework for authentication

A detailed description of the authentication framework can be found in [D05].

3.2.1 Objectives of the authentication framework

The proposed authentication framework has the following objectives:

- to provide a flexible procedure for user-network authentication
- to provide a procedure for SP-NO roaming agreement
- to provide a procedure for SP-NO authentication

The principle objective of the Authentication Framework is to provide a *flexible procedure for user-network authentication* allowing a number of different mechanisms and algorithms to be incorporated, with the ability to migrate smoothly from one mechanism to another. This framework allows the authentication capabilities of SIMs, network operators (NOs) and service providers (SPs) to be taken into consideration for the selection of the mechanism to be used. A list of capability classes (including the mechanisms supported) will need to be maintained so that different entities (SIMs, NOs, SPs and TTPs) can permit the negotiation of the mechanisms to be used.

In order to facilitate roaming in a network with a large number of NOs and SPs, it might be desirable (or even necessary) for *roaming agreements to be set-up dynamically*, as and when they are required. In practice, the roaming agreement would be first requested as a result of an initial authentication request sent by the user/terminal to a network visited for the first time. A prerequisite of this procedure is that the SP and NO wishing to establish the agreement have authenticated each other.

NO-SP authentication will be carried out using a globally agreed mechanism in order to ensure that NOs and SPs world-wide have the capability to authenticate each other. Unlike the user-network authentication mechanism, flexibility to change mechanisms is not considered to be a crucial factor. Apart from being a prerequisite to a roaming agreement, NO-SP authentication will permit the SP to delegate user-network authentication to the NO. The SP would send authentication data to the NO in advance, permitting the NO to carry out authentication on behalf of the SP.

It should be noted that the identity of the User is not released until the stage of user-network authentication. The rationale for this is that the identity of the User is immaterial until the stage of authentication is reached; it is only the identity of the Service Provider that is required up until the stage of authentication. Note also that the identity of the User is never necessarily required by the Network Operator, hence temporary identities are used to provide party anonymity of the User towards the Network Operator.

A further characteristic of the Authentication Framework is the use of an authentication Capability Class, which acts to identify the particular authentication mechanisms that are supported by the UIM of a User. Each respective authentication mechanism is identified by an unique identifier. The rationale for this is that visited Network Operators may immediately identify whether they can support a particular Capability Class; unknown authentication mechanisms would be defined by the respective Service Provider upon request from the Network Operator.

3.2.2 Authentication framework procedures

Procedure P1: User - NO authentication capability agreement

User and NO inform each other of their respective authentication capabilities, and subsequently agree the mechanisms to be used during their interaction.

Procedure P2: NO-SP authentication capability agreement

SP and NO interact in order to negotiate the user-network authentication mechanism to be used, based on the capabilities and preferences of the entities involved.

Procedure P3: Service provider - network operator authentication

SP and NO interact to authenticate each other.

Procedure P4: Establishment of NO - SP roaming agreement

NO or SP initiate a procedure to establish a roaming agreement. This may be done on-line as part of an authentication request in a registration attempt, or off-line as a separate procedure.

Procedure P5: User - network authentication

The user and network interact to authenticate each other. This procedure should allow requests from network to user, or vice-versa, to perform user-network authentication. The requesting network entity may be the NO or the SP.

3.2.3 Operational scenarios

Operational scenarios involving the Authentication Framework fit into two main categories: those that involve a request for user-network authentication, and those that involve a request to establish an NO-SP roaming agreement.

Operational scenarios for user-network authentication:

1. user initiates authentication request:
    - New registrations, no roaming agreement exists
    - New registrations, roaming agreement exists
    - Current registrations
2. NO initiates authentication request
3. SP initiates authentication request.

Operational scenarios for NO-SP roaming agreements:

1. NO initiates roaming agreement request:
    - SP and NO have not authenticated
    - SP and NO have authenticated
2. SP initiates roaming agreement request:
    - SP and NO have not authenticated
    - SP and NO have authenticated

A description of these operational scenarios can be found in [D05].

As an example the operational scenario is described where a user, not registered in the network, initiates authentication and no roaming agreement exists between the Network and the user's service provider.

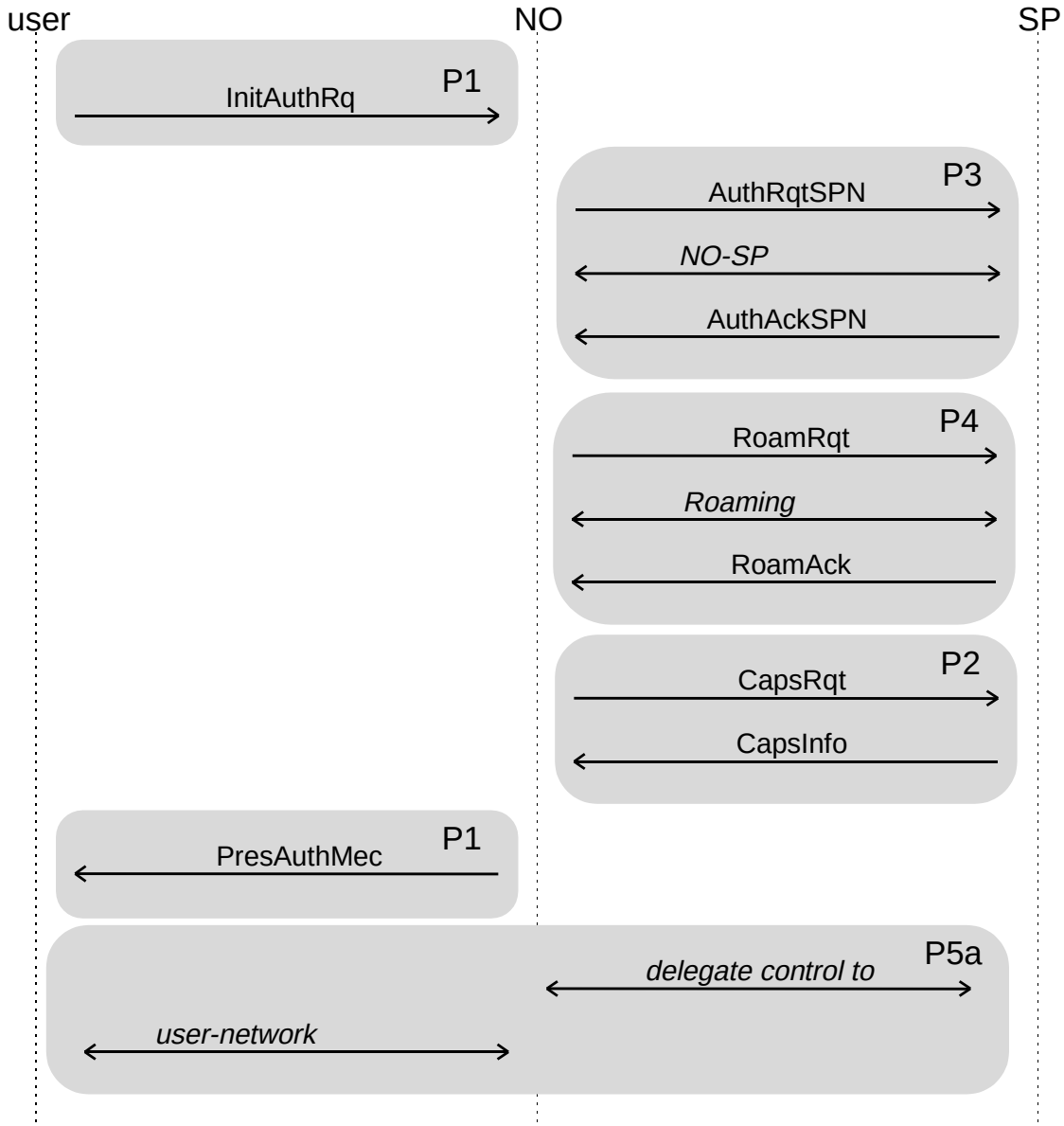


Figure 3.1 - Operational scenario for 'User not registered, no roaming agreement'

The user sends an initial message to a NO - this will include the user's service provider, authentication capability class, but not his identity nor his temporary identity. The NO does not have a roaming agreement with the SP so it initiates a procedure to establish one dynamically - if one cannot be established dynamically, then the request is refused. A procedure to establish a roaming agreement begins with the NO and SP authenticating each other. After authentication the NO and SP negotiate a roaming agreement which will involve each party digitally signing the agreement. Once an agreement has been established, the NO checks the authentication Capability Class of the User to establish if it is known. If it is known, the Network operator compares the associated authentication mechanisms with its own supported authentication mechanisms. If it is not known, the Network Operator sends the user's authentication capability class to his SP. The SP will respond by providing the NO with the authentication capabilities of that particular authentication capability class - this will include the authentication mechanisms the user is capable of handling. The NO will then choose an authentication mechanism, from those of the User's Capability Class, which is both supported by the Network Operator and by the User's UIM. The NO then sends the identity of the prescribed mechanism to the user. The authentication mechanism for new registrations involving the SP, NO and user is initiated. Note, however, that the SP may choose to delegate the actual authentication to a Certification Authority (CA).

3.3 Applicability of the authentication framework for a migration scenario

3.3.1 UMTS services

Future mobile telecommunication networks, such as UMTS, are envisaged to offer a wide range of high quality services. Following advanced provisioning methodologies, UMTS services are composed in a modular way by combining independent service components. A basic aspect of the UMTS service provision framework is the separation among bearer control, which enables the information transfer of an information flow, call control, which is closely related to the users involved in the call, and mobility management which enables service provisioning to mobile users. In addition, a service platform provides the interfaces among the service provider and the network operator in case these two entities are completely different.

The UMTS services may be seen as the combination of a service type and of one or more information flows. Concerning the service type, a service can be characterised either as an interactive service or as a distribution service. Interactive services are subdivided into conversational services which involve a connection-based, point-to point, and bi-directional communication (e.g. telephony), conferencing services which involve a connection-based, point-to-multipoint communication (e.g. video-conference), retrieval services which involve a connection-based, multipoint-to-point, and unidirectional communication (e.g. source information retrieval) and finally messaging services which involve a connectionless, point-to-multipoint, and unidirectional communication (e.g. e-mail). Distribution services may be divided into services with and without user control (e.g. television). Concerning the information flow the service may involve audio, video, data, text, graphics and pictures. These information flows may be characterised as rate-oriented, like audio and video, or as unit-oriented, like data, graphics and pictures.

It is very important for future communication to prevent the usage of services by unauthorised parties. The Authentication mechanism aims to provide mutual authentication, that is verification of identity between entities, and transmitted data origin authentication, that is verification of identity of the data originator by the data receiver. Various approaches for realising a particular mechanism exist. Some of the approaches for authentication mechanism are symmetric, public key, cryptographic check functions and zero knowledge approach. In order to ensure flexibility, which is one of the main UMTS key objectives, security aspects are separated from other UMTS procedures such as registration and service provision. In practice, service provision is interrupted by the authentication process as service request precedes authentication mechanism while service provision activities follows it.

Whilst a main feature of UMTS is flexibility as well as global interworking, the authentication framework should comply with a number of requirements. First of all, the authentication framework should provide support for all of the authentication approaches and existing mechanisms. It should also be prepared to support future upgraded mechanisms used by different service providers. Additional requirements concern the definitions of the procedure for obtaining and distributing information on the authentication capabilities of all parties involved and of the information flows in order to support the modular approach of UMTS. Finally, the interaction between the mobility procedures and the authentication procedures should be defined, especially the common parameters or requirements on the interfaces.

ASPeCT proposed an authentication framework which allows a number of authentication approaches to be used providing a flexible procedure for user-network authentication mechanism. In addition, it provides the capability to migrate smoothly from one mechanism to another. The selection of the mechanism to be used depends on the capabilities of the users, the service providers and the network operators. A list of capability classes, including the mechanism supported, will need to be maintained so that different entities (users, network operators, service providers and trusted third parties) can negotiate these mechanisms.

The authentication mechanism proposed by ASPeCT is compliant with the majority of the UMTS requirements. A number of authentication approaches are supported, permitting different service providers to use different mechanisms. The procedure for distribution of the authentication capabilities' information and the information flows are clearly defined, thus supporting the UMTS modular approach. Finally, the user anonymity is preserved throughout the authentication process and the cost of additional information

transfer is minimal, compared to the extensive flexibility in selecting the authentication mechanism, which is imperative for a UMTS environment.

3.3.2 Packet based services

A Packet Based service uses a fundamentally different mechanism for information transfer than a Circuit Switched service. In a Circuit Switched (CS) service, the Calling Station uses a Call Request signal to establish a single route between itself and the Destination Station. Once received by the Destination Station, a Call Accept signal is returned along the created path. Once the Calling Station receives this, the information transfer can begin. While the call is supported, the Network resources used en-route between the Calling and Destination Stations are dedicated to this call. This results in a high utilisation for voice connections. However, for data connections, much of the time the line is idle. A further limitation on the Network is that the transmission rate must be the same as the receive rate at the other end. A Packet Based Network addresses both of these problems.

A Packet Based (PB) Network views the Network in a fundamentally different way. The PB Network is considered to be composed of an array of nodes, where each node can communicate with a set of surrounding nodes. Data within the Network is divided up into and transmitted as short packets. Each packet is composed of the User's data, and sufficient control data to allow the Network to route the data to its destination. This is achieved by a packet at a node in the Network determining which is the best node to be forwarded to according to some criteria. Examples of criteria are queue size at the destination node, quality of link, speed of link etc. The packet is then forwarded to the chosen node once the link is available. This approach has several advantages such as efficient resources usage, reduced congestion, multi-rate and priority information transfer.

Within GSM, if the User data is unencrypted, then all Network data is unencrypted, which opens up the Network for attack. Clearly this is unacceptable within UMTS, so an additional requirement can be stated which is that the protection of User and Network data must be considered as independent requirements. The User data consists of the User generated data that is embedded within packets for distribution through the network. The Network data consists of both the routing and authentication information that surrounds the User data within the packets, and non-User Network information such as signalling and management data contained within the packets.

The Authentication Framework currently authenticates the User, and establishes the encryption key to be used for data traffic. The messages required to establish this key and authenticate the User are distributed as Network traffic, and are carried as data within packets. These cannot be encrypted between the User and the NO as a key has not been established at that point. Once the key is established, then any User data such as voice traffic can be encrypted using this key. The messages are sent as User data within the packets, and are surrounded by the Network data related to routing and authentication. This satisfies the requirements for establishing an encryption key, and for providing security for the User data. However, there are no requirements addressed with regard to the Network data.

The existing Authentication framework is concerned with authenticating the physical User to the Network, and not authenticating the Mobile Station (MS) to be part of the Network. Within a PB Network, the MS may become a node within the Network, and hence there is a complete set of requirements for the authentication and registering of the MS that are currently missing. It is logically important to draw a distinction between User and MS authentication, as they satisfy very different requirements.

3.3.3 UMTS network

In UMTS architecture the physical aspects are modelled using the domain concept. A domain is defined as a high level grouping which focuses on physical entity aspects, and comprises all of the functionality in those entities. A module is then seen as a specific instance of a domain with the capability to provide a subset of the functions of one or more strata.

A basic architectural split is between the user equipment (terminals) and the infrastructure, which results in two domains: the User equipment domain and the Infrastructure domain. The user equipment domain itself

can again be split up into several elements. One example is the USIM, an application that usually resides on a removable smart card. The level of functionality within the user equipment can vary, e.g. dual mode UMTS-GSM terminals. The Infrastructure domain is further split into the Access Domain that is characterised by being in direct contact with the User Equipment, and the Core Network Domain. The Access Domain comprises roughly the functions specific to the access technique, while the functions in the Core network domain may potentially be used with information flows using any access technique. This split allows different approaches to be used for the core network domain, with each approach specifying distinct types of Core Networks each connectable to an Access Domain, and also allows different access techniques, each type of Access Network connectable to a Core Network Domain. The Core network is further split into the Serving Network and the Home Network. No more work is available on how the tasks will be assigned. The Application Network Domain is defined but not further described. This can be seen as the equivalent of the Value Added Service Provider, VASP, or content provider, CP, which are described in the UMTS role models.

The logical aspects are modelled using the strata concept. The stratum is defined as a high level functional groupings focusing on one or more protocols, and comprising all the entities involved in those protocol. The Access stratum is the functional groupings consisting of the parts in the infrastructure and in the user equipment and the protocols between these parts being specific to the access technique (i.e. the way the specific physical media between the User Equipment and the Infrastructure is used to carry information). Functions related to the control and management of the radio resources are located in this stratum.

The Serving stratum consists of protocols and functions to route and transmit data/information, user or network generated, from source to destination. The source and destination may be within the same or different networks. Functions related to telecommunication services and mobility management may be located in this stratum. The Home stratum contains the protocols and functions related to the handling and storage of subscription data and possibly home network specific services. Functions related to subscription data management, customer care, mobility management, including billing and charging, may be located in this stratum. The Application stratum includes end-to-end protocols and functions that make use of services provided by the home, serving and access strata and infrastructure to support value added services. End-to-end functions are applications that are consumed by users at the edge of/outside the overall network. These applications may be accessed by authenticated users who are authorised to access such applications. The users may access the applications by using any of the variety of available user equipment.

The messages defined in the authentication framework have been generalised and mapped upon the physical model, the domains. The request for authorisation to the HN only has to occur if the user initiated the contact. Authorisation has to be granted for the use of the services in the specified AN. In addition, the preferred Serving network has to be indicated by the HN. The HN has to achieve the best buy for the user. This can be different depending on the requested service. The communication between the AN and HN, which is completely user independent, is supposed to be authenticated. During the security procedures, there can be a further need by the AN to interrogate the HN for security data (e.g. response parameters, that can be calculated only in the HN). This is a user dependent interrogation. When the user is already known in the AN, the procedure can be simplified and good real-time procedures will no longer need an interrogation of the HN and fewer data to be transferred. The main aim of security procedures is mutual authentication, setting of a cipher key between the Mobile and the AN, anonymous user access, non-repudiation of access of service and initialisation of on-line billing process by means of micro-payments (between the mobile and the HN). At the end of a successful authentication sequence a cipher key is set between the AN and the smart card. This cipher key can be used to cipher the connection between the mobile terminal and the AN.

The authentication framework can be used within the UMTS architecture currently defined. More work has to be done to integrate the authentication framework with the signalling procedures generated for UMTS. In addition it has to be enhanced to support more security mechanisms, especially for non-repudiation of billing, on-line billing and support for electronic commerce

3.4 Authentication demonstration and trial

3.4.1 Authentication demonstration

3.4.1.1 Overview

The aim of the demonstrator is to show a migratory path for security features. After an in depth study on the migration problem, a multi-application card for GSM and UMTS is proposed as the migratory path to introduce new and enhanced security features. The UMTS authentication framework is implemented in combination with a public key based authentication mechanism (see [D19]).

Within the demonstration, three logical entities (roles) are involved:

- The User: is authorised by a subscriber to access the telecommunication services of a Service Provider with whom the subscriber has a subscription.
- The Network Operator: provides the network capabilities necessary for the support of the services or set of services offered to the users.
- Service Provider: has overall responsibility for the provision of a service or set of services to users associated with a subscription and for negotiating the network capabilities associated with that service or set of services with network operators.

These roles have been mapped upon the following physical architecture:

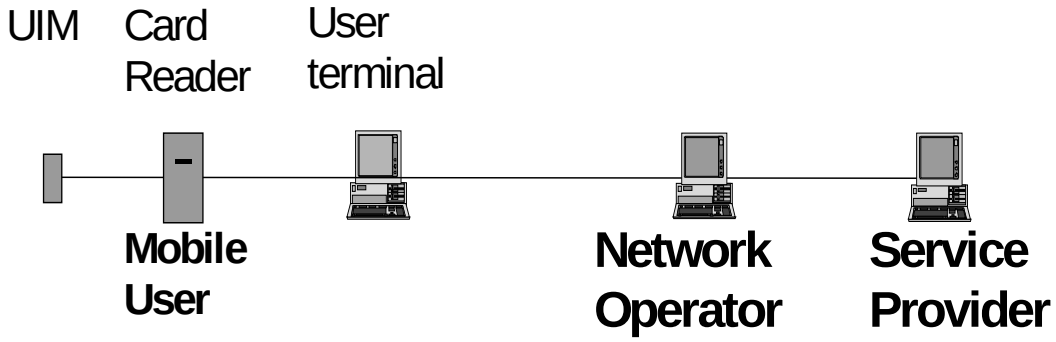


Figure 3.2 - Physical Architecture

Description of the demonstrated features:

New registration, roaming agreement exists:

The user is not registered with the NO, the NO and user's SP do have a roaming agreement. The NO has no security related data for the user.

The user wants to register and sends a registration request to the network. The network recognises that it has a roaming agreement for the user's service provider. The NO will receive the authentication capabilities from the registered user by the SP. An authentication mechanism will be negotiated and executed between the NO and the user.

Authentication of registered user:

The user is registered with the NO, the NO has security related data for the user.

The NO or the user can initiate the authentication by sending the appropriate message. The NO has all necessary data of the user, it has a roaming agreement with the SP and knows the users

authentication capabilities. An authentication mechanism will be executed between the NO and the user.

3.4.1.2 The UIM realisation

Details about the UIM realisation are presented in Section 4.

3.4.1.3 The network realisation

A detailed description can be found in [D19].

The demonstrator consists of several entities exchanging messages to each other. Each entity acts like a finite state machine. It receives an event (a communication message over TCP/IP, serial link or message queue or a user message from the Graphical User Interface) and responses to that event by taking some actions like calculating an algorithm and sending a message. Both communication between entities in the same application (via a message queue) as well as communication between entities in different applications (via TCP/IP) is possible.

It was agreed between the ASPeCT partners to use the ACRYL library from Siemens ZT IK 3 for the provision of basic cryptographic functions. Following functions are provided by ACRYL, which stands for Advanced CRYptographic Library :

- Random number generation based on DES-OFB and triple DES-OFB
- Hash functions RIPEMD-128 and RIPEMD-160
- RSA signature generation and verification
- AMV signature generation and verification based on an elliptic curve over GF(p)
- Encryption with DES-CBC and triple DES-CBC
- Exponentiation in GF(p)
- Exponentiation in an elliptic curve over GF(p)
- Key generation for RSA, DES and elliptic curves

3.4.2 Authentication trial

The UMTS authentication trial used an ASPeCT mutual authentication protocol between smart cards (or UIMs) attached to EXODUS terminals, and an ASPeCT AuC attached to an EXODUS SCP. In the trial, ASPeCT security services were integrated into the EXODUS signalling system. The experimental UMTS platform provided by EXODUS was enhanced by the authentication functionality provided by ASPeCT. The main objective of the trial was to show the feasibility of the implementation on a real network.

Communication between the ASPeCT terminal and the network should have been provided by DECT and fixed broadband access. In the event, only the fixed broadband access was provided. The terminal itself was enhanced with the necessary ASPeCT software to interface with a smart card-based UIM connected to the terminal. The user part of the authentication protocol ran on the ASPeCT UIM.

The UMTS network functionality was provided by the EXODUS core network. The security mechanisms were incorporated into the EXODUS core network using an ASPeCT AuC connected to the EXODUS SCP. Authentication was conducted over INAP, using the Authentication-request operation. Interrogation of the service provider was realised using the INAP HandleInformationRequest operation.

It was the intention to make the trial configuration available across two UMTS Islands in Basel and Milan. Within this configuration the distinction between the network operator and the service provider could not be made. Each SCP would have home subscribers and visiting subscribers. The AuC connected to the SCP would fulfil two roles, home-AuC and visited-AuC. As home-AuC, its only functionality would be the translation of the authentication capability class into authentication mechanisms. As visited-AuC, it would have to run the full protocol.

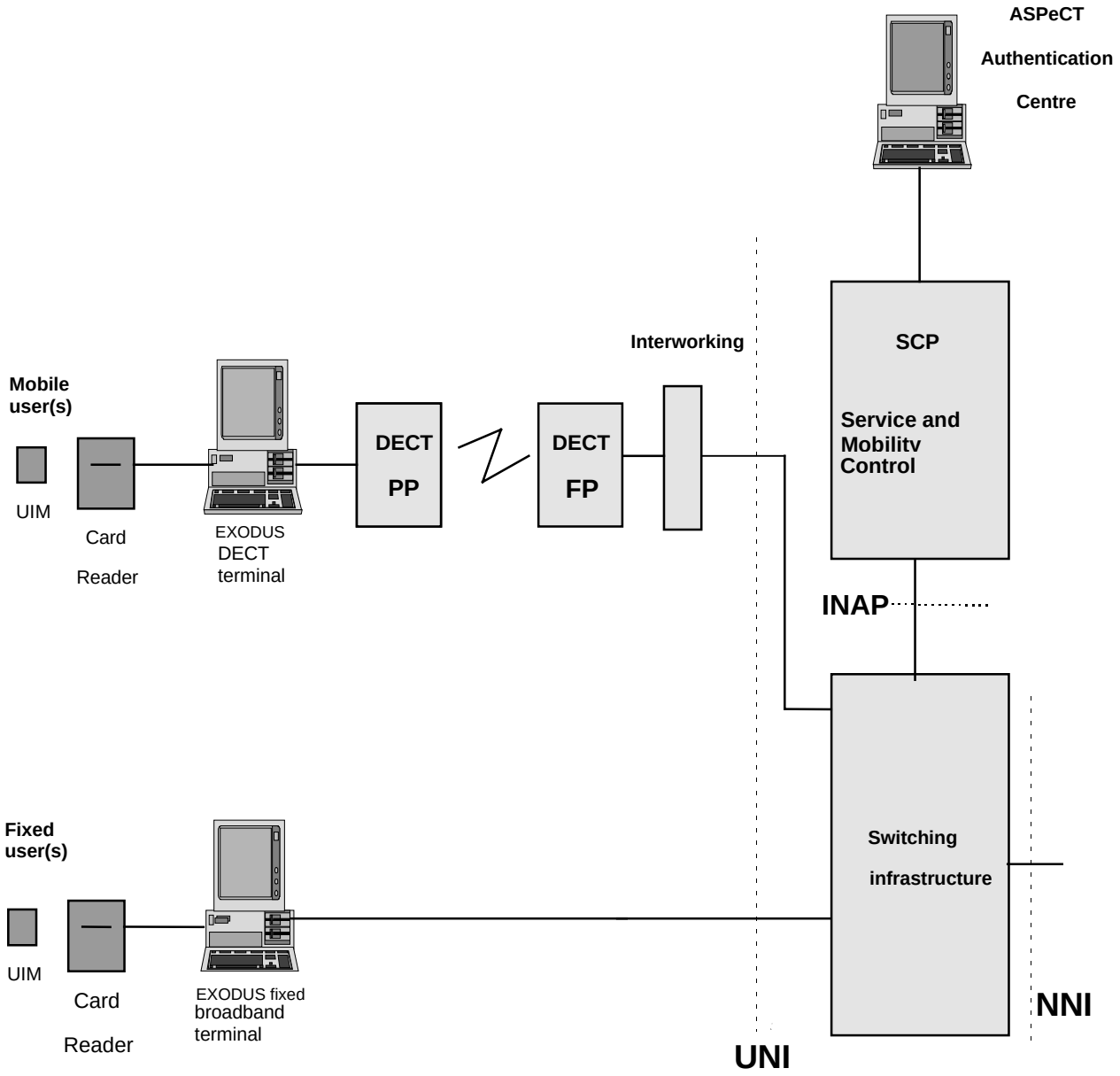


Figure 3.3 - ASPeCT - EXODUS trial configuration

3.5 Evaluation

3.5.1 Evaluation of the authentication demonstration

The following table gives an overview of the public demonstrations and the reaction on the demonstration.

Date	Event or exhibition	Nature of demonstration	Reaction
16-18-06.97	2nd International Distributed Conference on Network Interoperability, Madeira	Demonstration of rudimentary UIM implementation together with full implementation of network side	Demonstrator well received. Provoked strong interest from industry.
7-9.10.97	ACTS Mobile Summit, Aalborg	Demonstration of further evolved UIM implementation	Demonstrator well received.
25-28.5.98	IS&N98 conference, Antwerp	Demonstration of authentication framework with a public key authentication mechanism	Demonstrator well received.
8-11.6.98	ACTS Mobile Summit, Rhodes	Demonstration of authentication framework with a public key authentication mechanism	Considerable interest in the demonstration; 10 demo's each coffee break;

3.5.2 Evaluation of the authentication framework

The authentication framework has been evaluated as a basis for a migration scenario in UMTS, enabling the evolution of security functionality. The principle objective of the authentication framework, namely *'to provide a flexible procedure for user-network authentication allowing a number of different mechanisms and algorithms to be incorporated, with the ability to migrate smoothly from one mechanism to another'* is fulfilled. The authentication framework is compliant with the majority of UMTS requirements. Both public and secret key approaches are supported, permitting different service providers to use different mechanisms. The UMTS modular approach is supported. User anonymity is preserved throughout the authentication process. Finally, the cost of additional information transfer is minimal, compared to the extensive flexibility in selecting the authentication mechanism, and this will be imperative in the UMTS environment

3.5.3 Evaluation of the public key based authentication mechanism

The public key authentication mechanism for UMTS has a number of benefits over the GSM secret key authentication mechanism. Some of these benefits are listed below:

- Mutual authentication between the user and the network operator is achieved. In GSM only the user is authenticated towards the network.
- No on-line connection to the home network (i.e., the service provider) is required. In GSM the triplets used for authentication are generated within the home network (i.e., HLR/AuC) and must be retrieved from there.
- The user's identity is never transmitted in plaintext over the air-interface. The protocol starts anonymously and the identity is sent encrypted. In GSM, 'IMSI catching' is theoretically possible and the network can always ask the mobile to send the user's identity in plaintext.

- The user is unable to deny that he or she actually used the network. This is known as non-repudiation, and may save network operators from losing revenue when users claim that calls were never made. Non-repudiation also ensures that the network operator cannot deny receiving particular information from the user.

The public key authentication mechanism has been compared for resource use with the GSM mechanism. Three critical resources can be considered: data to be saved in the entities, data transmitted over the air, processing capacity.

- Data to be saved in the entities:

Home network: In GSM for each subscriber the Ki (16 bytes) has to be securely saved in the AuC. In addition more space is occupied with pre-calculated triplets (28 bytes each). The UMTS public key mechanism requires no keys to be saved in the Home network.

Visiting network: In GSM 5 triplets (140 bytes) are saved for each subscriber roaming in the network. The UMTS public key mechanism requires 112 bytes to be saved, network specific (the networks key set and the public key of the CA). In addition for each subscriber roaming in the network the public key (48 bytes) has to be saved.

The smart card: For GSM only the Ki (16 bytes) had to be saved. For UMTS 112 bytes (the users key set and the public key of the CA) have to be saved. When the user is registered the public key of the network (48 bytes) has to be saved additionally.

- Data transmitted over the air:

From the network to the MS: In GSM only a random number (16 bytes) is sent from the Network to the MS. For UMTS an authentication of a new user requires 166 bytes (a random number, an authentication response and a certificate) to be sent. An authentication of a known user requires 32 bytes (random number and authentication response).

From the MS to the Network: In GSM only an authentication response (SRES, 4 bytes) is sent. For UMTS an authentication of a new user requires 236 bytes (a key agreement value, a signature and a certificate) to be sent. An authentication of a known user requires 96 bytes (key agreement value and a signature).

- Processing capacity:

  Visiting network: measurements were done on the demo architecture, using INTEL 486 33,4 Mhz - PCs.

  The user's side (smart card): most of the time necessary for the protocol is spent by the processing of the user's smart card (first prototype). Further development is required to increase the processing power of the smart card.

  The total authentication process takes several seconds in the current demonstration.

The restrictions of the demonstration environment, being just a prototype of some network entities, restricts the value of the measurements. However, the results do give a first indication on the impact and feasibility of having multi-application cards and authentication based on public key mechanisms.

3.5.4 Evaluation of the authentication trial

The joint ASPeCT/EXODUS trial was seen as a way of performing a more realistic evaluation of the Authentication Framework. However, there were many problems encountered in getting some form of Authentication trial ran at all. The original intention was to run the trial with a group of users which would provide both subjective data relating to the acceptability of the delay produced as a result of the authentication mechanism, and objective data which would provide the actual timings of each authentication. Such a trial was to use both mobile and fixed terminals.

In the event, only fixed broadband terminals could be used. In addition, only the Siemens Authentication protocol was trialled performing both new registrations and current registrations. The actual trial that took

place consisted of a new registration followed by a current registration, and this repeated twenty times. The two protocols used are given in the below diagrams.

These diagrams give the averaged timings in seconds, assuming the starting time always to be 0. The values are given without errors, as the results from all twenty trials are very similar. Indeed, the Standard Deviation is typically only 2% of the elapsed time. In addition, the change in the state of the AuC is also given in the 'state boxes'.

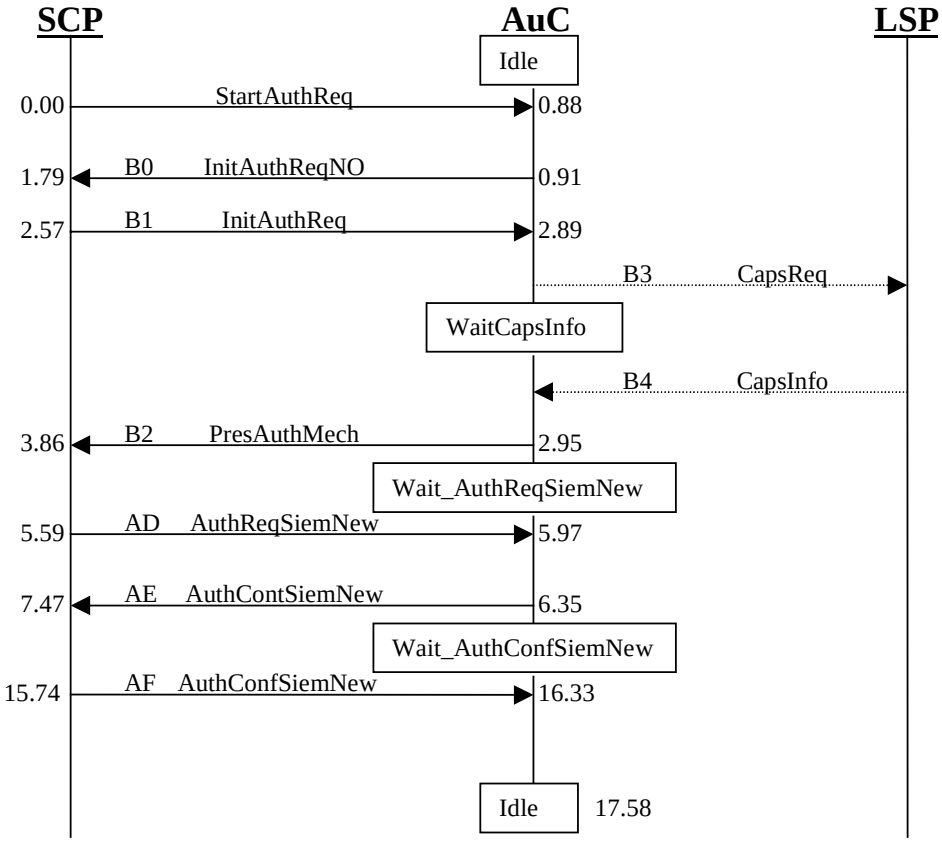


Figure 3.4 - New Siemens Authentication - EXODUS timings

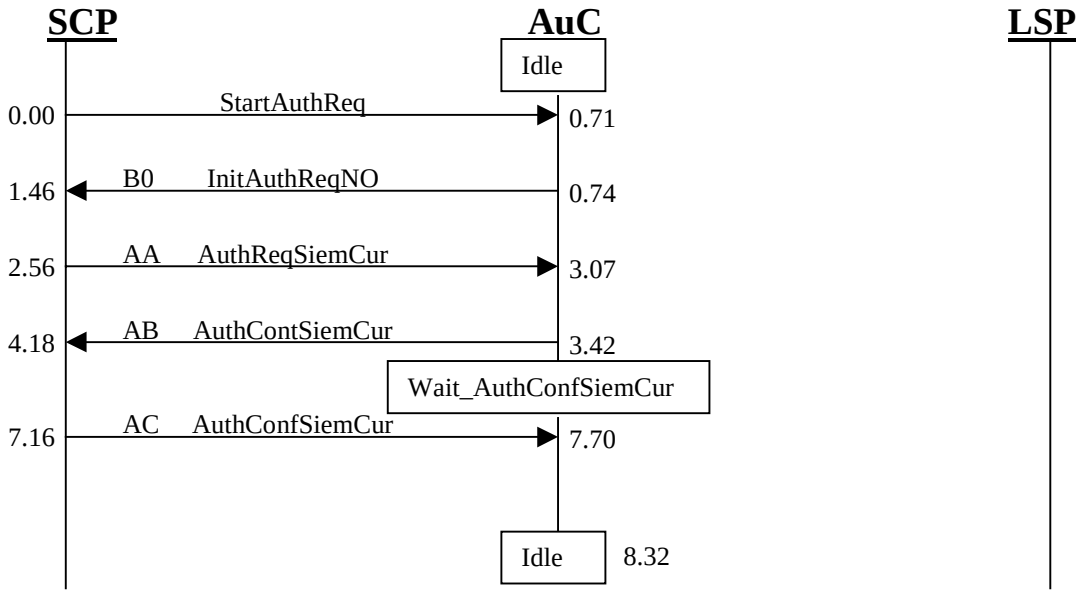


Figure 3.5 - Current Siemens Authentication - EXODUS timings

The New Registration protocol contains a message to (CapsReq) and from (CapsInfo) the Local Service Provider. However, in the actual trial, this message flow only occurred once, as after it had occurred, the data was stored at the AuC. For this reason, the first pass of the protocol was ignored, and all timings data taken using subsequent runs.

In addition to the above diagrams, the actual results are given below.

SCP		AuC
0	→	0.879737
1.787368	←	0.907632
2.572105	→	2.893421
3.858947	←	2.948684
5.594211	→	5.967105
7.471053	←	6.351316
15.74	→	16.32711
		17.57974

Table 3.1 - Siemens New Registration Times

SCP		AuC
0	→	0.043091
0.078517	←	0.040013
0.078428	→	0.055779
0.090179	←	0.055424
0.09777	→	0.077268
0.117421	←	0.077815
0.165395	→	0.150843
		0.139647

Table 3.3 - Standard Deviation of New Registration Times

SCP		AuC
0	→	0.713421
1.455263	←	0.741842
2.563158	→	3.070263
4.175789	←	3.418684
7.159474	→	7.696579
		8.322368

Table 3.2 - Siemens Current Registration Times

SCP		AuC
0	→	0.021282
0.027962	←	0.018121
0.039165	→	0.044829
0.049813	←	0.053511
0.117873	→	0.116035
		0.119326

Table 3.4 - Standard Deviation of Current Registration Times

As can be seen from the above times, it takes 8 seconds in the new registration for the message to leave the SCP, get processed by the UIM, and returned to the SCP, and 3 seconds in the case of the current registration. Both of these times are approximately equal to the time required for the rest of the protocol to be run.

By studying the differences between the two protocols, the five second difference between the two processing times in the UIM can be seen to be due to a combination of the UIM verifying the NO's certificate and returning the User's certificate enciphered under the session key. This clearly indicates that the use of certificates only becomes acceptable once the processing power of the UIM is such that these times are reduced by at least one order of magnitude. The performance could be improved by simplifying the certificate structure and reducing the amount of data which needs to be enciphered. It may also be possible to simplify the processing required in the UIM by, for example placing some of the processing in the terminal, although this would result in a reduction of the achieved security

The difference between the performance of the system using the EXODUS network, and the behaviour of the demo system can also be studied. Below are reproduced the two protocol diagrams, and these now include the timings for the demonstration system.

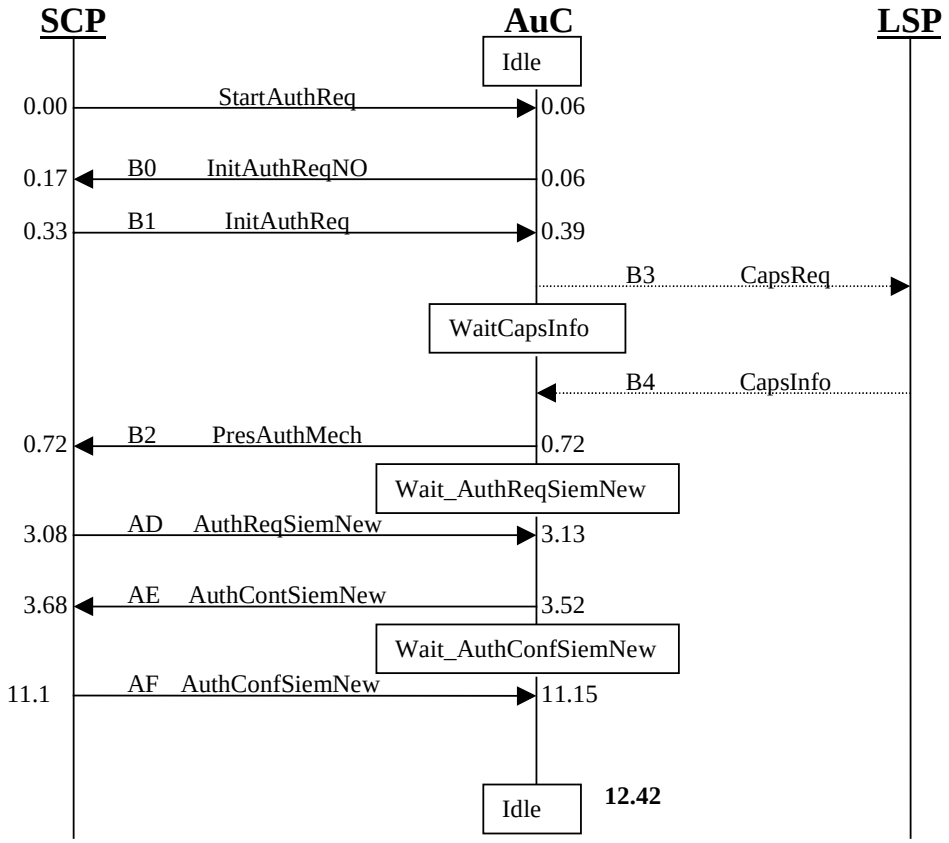


Figure 3.6 - New Siemens Authentication - Demo timings

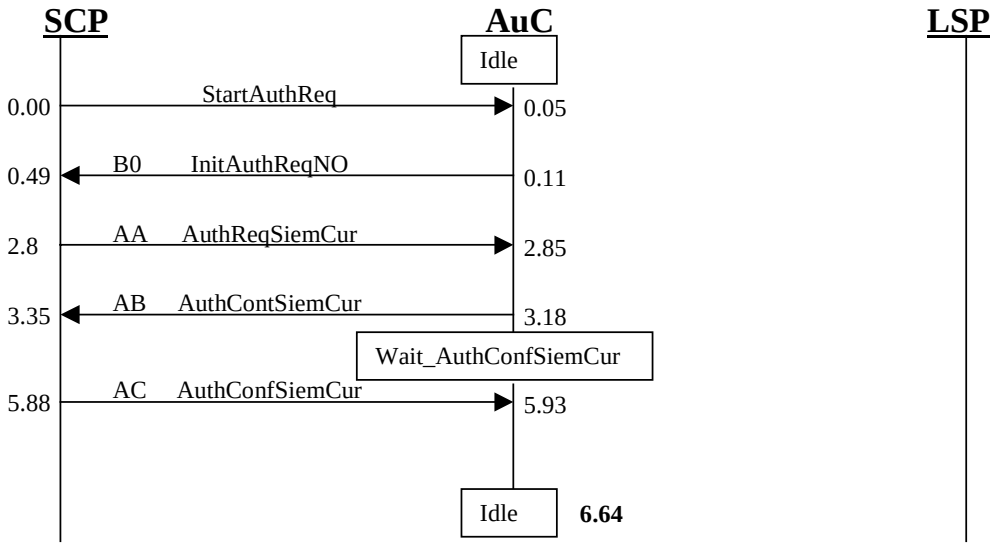


Figure 3.7 - Current Siemens Authentication - Demo timings

Following tables show a comparison between time measurements in the PC demo configuration and in the EXODUS trial. Times are expressed in seconds.

New Siemens Authentication		
	EXODUS-trial	Demo on PC configuration
SCP total processing time	10.78	9.94
AuC total processing time	1.72	1.99
AuC processing time till state Wait_AuthConfSiemNew	0.47	0.72
AuC processing time between state Wait_AuthConfSiemNew and state Idle	1.25	1.27
Total time for authentication	17.58	12.42

Current Siemens Authentication		
	EXODUS-trial	Demo on PC configuration
SCP total processing time	4.08	4.84
AuC total processing time	1.00	1.10
AuC processing time till state Wait_AuthConfSiemNew	0.38	0.39
AuC processing time between state Wait_AuthConfSiemNew and state Idle	0.62	0.71
Total time for authentication	8.32	6.64

Previous tables show that the measured times in the EXODUS-trial and in the PC demo configuration are very similar.

3.6 Conclusions

The principle objective of the Authentication framework '**to provide a flexible procedure for user-network authentication allowing a number of different mechanisms and algorithms to be incorporated, with the ability to migrate smoothly from one mechanism to another'** is fulfilled.

The authentication framework is compliant with the majority of UMTS requirements as laid out in [ETS33.21]. Both public and secret key approaches are supported, permitting different service providers to use different mechanisms. The UMTS modular approach is supported. User anonymity is preserved throughout the authentication process. Finally, the cost of additional information transfer is minimal, compared to the extensive flexibility in selecting the authentication mechanism which will be imperative in the UMTS environment.

Though the UMTS security architecture is not yet approved, it seems that public key techniques for service-related user authentication will not be part of UMTS Phase 1. However, ETSI SMG10 has approved the concept of an authentication framework for UMTS, which would allow new mechanisms, including public key mechanisms, to be added in later phases of UMTS.

During evaluation, topics were encountered where further work is necessary and which are not covered by this project:

- Within the UMTS society, more value is now being given to ‘support for electronic commerce’ and ‘incontestable charging’ than was originally envisaged by ASPeCT. The security implications of having a UMTS network supporting electronic commerce have not yet been fully studied, nor has there been any work to integrate incontestable charging protocols into the basic UMTS security mechanisms. However, within the secure billing work in ASPeCT, a micropayment protocol providing incontestable charging has been developed, although this protocol is integrated into mechanisms implemented at the application level and has not been integrated into basic UMTS security mechanisms.
- It is now clear that for UMTS more use will be made of packet-based services. It is not clear whether the authentication framework can fulfil the requirements of packet-based services. Further work is required into the development of appropriate security mechanisms for packet-based services in UMTS. Furthermore, mechanisms to support the security of network data (signalling and management data) have not been fully considered.
- Since the authentication framework was developed by ASPeCT, there has been an evolution of the proposed UMTS architecture. The most recent version of ETS 23.01 [ETS23.01] describes a different physical model of the UMTS architecture compared with the one originally assumed by ASPeCT. An evolution towards the current domain/strata model was not envisaged by ASPeCT. Furthermore, the network operator is now split into two roles: an access network operator and a serving network operator. In addition, the present domain/strata model lacks a security architecture. Further work is required in order to position the authentication framework within the appropriate domain/strata of the UMTS architecture.
- ASPeCT has not studied the integration of the authentication framework within the UMTS signalling procedures. At the start of the project these procedures were not available, and even now much work is still to be done.

To address the issues discussed in the previous section, work has started as part of a new ACTS project, AC336 USECA. Its objective is to define a complete security architecture for UMTS, and to present it for standardisation.

4 UIM security functionality

4.1 Introduction

This section describes the work carried out in ASPeCT with regard to the security functionality of the UIM in UMTS. It is largely concerned with the UIM, which was developed to demonstrate and trial one of the proposed authentication algorithms for UMTS. This operates in combination with the security migration demonstrator and allows a mutual authentication between the Network Operator and the User based on public key technology. Simultaneously with the authentication a shared key is established between the participants, this can then be used, for example, to encipher the voice or data traffic.

4.2 Requirements on the UIM in UMTS

4.2.1 The need for a UIM in UMTS

Before considering the role of the UIM in UMTS, it is sensible to review the position of the SIM in the GSM environment.

4.2.1.1 The GSM SIM

The GSM SIM is a security micro-controller, which was originally embedded in a credit card sized plastic body (ID-1). To allow for greater flexibility in handset design a smaller size plastic body, the plug-in or microSIM (ID-000), was standardised. The principal goal of the SIM was to provide enhanced security for the network operators who had suffered from fraud in the previous analogue networks. By requiring a SIM to be present in the mobile equipment the network operator is given a security module as close to the mobile user as could be desired. At the same time, the responsibility for developing the security of the SIM is delegated to specialist companies whose reputation depends on the security of these devices. If the security were simply required to be implemented by the terminal manufacturers themselves then the security level actually achieved would be much harder to ascertain and it is doubtful that it would have been as successful as it has been in preventing fraud.

The SIM is responsible for three components in the overall security of GSM. Firstly, it allows for a user authentication by requiring the user to enter a valid PIN before further access to the SIM functionality is granted. Secondly, it is responsible for authenticating the mobile terminal to the network operator by calculating the same response to a challenge as the Authentication Centre of the user's home network. The SIM calculates this response using an algorithm, designated A3, which is defined by the user's service provider. This algorithm can be different for every operator and its security is crucial for the overall security of the system. The SIM performs its final security operation by calculating a secret key using another operator specific algorithm, designated A8. This key is then used to encipher the voice traffic between the mobile equipment and the base station using the A5 algorithm. For performance reasons the A5 algorithm is implemented in the mobile handset. This means that the A5 algorithm must be supported by the base station and cannot be operator specific.

To be successful the SIM should not just be a convenience to the network operator, it must also be perceived by the user to directly provide some benefit. Originally, this was mainly the ability to store a list of telephone numbers, which could then be used without needing to be re-entered. Of course, this could have simply been implemented in the mobile handset itself. However, because it is in the SIM the user can remove the SIM from one handset and insert it in another. This was originally conceived to provide some form of user mobility since it was not expected that handsets would shrink in size so rapidly. Thus, a travelling business user was, for example, more likely to hire a car containing a phone which could be personalised simply by inserting the SIM rather than bringing his mobile handset with him. Other user features provided by the SIM include call barring and the ability to store short messages received by the Short Message Service (SMS).

4.2.1.2 The UMTS UIM

Whilst the above justifies the use of a SIM in the GSM system it might well be that these reasons are no longer applicable and that no SIM equivalent is required in UMTS. For example, in the US and Japan most mobile operators do not use anything corresponding to a SIM.

However, the arguments for having a removable token which stores user data and provides security features are probably even stronger in UMTS. The following points are the main justifications for having such a token in UMTS; which at the commencement of the ASPeCT project was called the UIM but has now been renamed the USIM:

- **Security** – The security requirements have clearly not disappeared and will increase in UMTS.
- **User Mobility** – The ability of the user to remove the UIM from one handset and insert it in another is an important concept in the idea of user mobility, which is fundamental to UMTS. This mobility is further enhanced by the idea of a Virtual Home Environment and is being developed by SMG1. This concept means that the user will be able to have the same profile and man-machine interface irrespective of the terminal he is using.

  It is clear that without some form of removable token it would be much more difficult to support inter-standard mobility.
- **Operator Differentiation** – The UIM is the only part of the mobile terminal belonging to the operator. It is clear, therefore, that if the operator wants to provide extra services, which require security, to his customer then this must be done in conjunction with the UIM.
- **Multiple Applications** – When GSM first came out it rapidly became the largest market for smart cards. Nowadays, smart cards are becoming ubiquitous and are being widely deployed – particularly in the banking world. This opens up many opportunities where the UIM could have additional applications on the card as well as the principal UMTS application.

The GSM SIM has been steadily evolving with the introduction of new features such as the SIM Toolkit but the advent of UMTS will require a technological jump to the UIM rather than just continued evolution of the SIM. This will be because of the new requirements of UMTS and because of advances in security and chip card technologies.

The capabilities of smart cards have substantially advanced since the GSM system was defined. The processors are becoming more powerful with more RAM, ROM and EEPROM resources. More interestingly, from a security perspective, hardware is being developed containing cryptographic co-processors. These substantially enhance the speed at which a smart card can perform the primitive cryptographic functions. There are already co-processors for performing modular arithmetic and DES and some newer designs include hardware to facilitate the implementation of Galois field arithmetic for use in elliptic curve crytposystems. These co-processors make the goal of public key cryptography achievable for both the financial and telecommunications' markets.

The advantage of public key techniques is twofold. Firstly, key management is simplified. It is no longer necessary to have a unique key for every partner with whom you could conceivably want to securely communicate. This is an important factor for the much larger number of smaller operators envisaged in UMTS. Using public key cryptography means that it is no longer need to check with the home network operator when verifying a roaming user. Secondly, the asymmetry of the keys, enhances security because it is no longer possible for someone who can verify your signatures cannot generate also them.

4.3 Demonstration and trial

4.3.1 Overview

This section provides a description of the different versions of the UIM which were developed for use in the Demonstration and the Trial. These different versions were made on the same hardware platform but with two different, although related, underlying operating systems. The primary OS target was STARCOS

SPK2.1, an operating system specifically developed to support public key based cryptographic systems. As such it was the obvious candidate to demonstrate the elliptic curve based authentication mechanism which was used in ASPeCT. The secondary OS was an operating system developed to implement a standard GSM SIM. This was used to demonstrate the migration path from GSM to UMTS by providing a smart card able to support both a GSM and the ASPeCT authentication mechanisms.

The UIM used for the demonstration and trial purposes was the UIM implemented using the STARCOS SPK2.1 operating system. In addition to the ASPeCT authentication mechanism this card could also support other applications. Support was also provided to the vocal biometric user authentication work as described in Chapter 5.

Following the specification of the UIM, the interface between the card reader and the PC host is described. This interface is performed by the COLA interface and it provides an API, whose goal was to minimise the details about the UIM concerning the remainder of the ASPeCT software. The COLA interface additionally provides an emulation of the UIM that facilitates debugging and allows the authentication mechanism to be demonstrated without needing access to either the UIM or a card reader.

Personalisation of UIM is explained and finally, there is a brief conclusion.

4.3.2 ASPeCT authentication application

This section provides an abbreviated version of the specification of the Authentication Application. The file system and the commands supported by the UIM to implement the application are described.

The overall authentication process is described in ASPeCT deliverable [D12].

Note that the two implementations use different communications protocols to transmit data between the card and the card reader. The implementation based on STARCOS SPK2.1 uses the T=1 protocol whereas the GSM compatible based implementation uses the T=0 protocol – for details on the protocols see [ISO7816-3]. The communications protocol being used has little impact on the application but does require that the commands sent to the smart card are different. In particular, if the card needs to send a response to the card reader then this naturally happens as part of the protocol in the T=1 case. However, in the T=0 case the card must be sent a GET RESPONSE command to retrieve this data. This mechanism is handled internally by the card reader and the COLA interface and is transparent to the rest of the Authentication software, such as the EXODUS Terminal.

4.3.2.1 File system

This section describes the file system present on the UIM. A summary of the files is shown in Table 4.1 and then the content of the files is considered individually.

File	Type	File Identity	Short File ID	Description
DF_UMTS	DF	7F50		UMTS Application
EF_SPID	Binary	6FE0	19	Service Provider ID
EF_AUCC	Binary	6FE1	01	Authentication Capability Class
EF_IMUI	Binary	6FE2	02	IMUI
EF_CERTU	Binary	6FE5	05	User certificate
EF_CERTN	Binary	6FE6	06	Network Operator certificate
EF_SECU	Binary	6FE7	07	User's secret key
EF_IPF	IPF	6FE8	1D	Internal Public key file 1: Certification Authority public key 2: Domain parameters of Elliptic Curve (p, q, A, $\boldsymbol{G}$) 3: Network Operator public key
EF_CAID	Binary	6FE9	18	Certification Authority ID
EF_IMUN	Fixed Record	6FEC	11	EXODUS IMUNs 17 records of 10 bytes
EF_EXODUS	Binary	6FED	12	EXODUS TMUI, LAI/FPAI, LI, CPN
EF_KS	Binary	6FEE	1E	Session Key, Ks
EF_TLV	Object	1000		Operating system Object file

Table 4.1 - File system for the UIM

DF_{UMTS}	**UMTS Application**	
File ID: 7F50	**AID:** 'UMTS'	

EF_{SPID}	**Service Provider Identity**		
File ID: 6FE0	**SFI:** 19	Binary	05 bytes
Read AC	Following successful CHV-1 authentication		
Write AC	Following successful CHV-4 authentication		
01-05	In ASPeCT this is the first 5 bytes of the IMUI encoded in ASCII: "22201" Italian Service Provider "22801" Swiss Service Provider		

EF_{AUCC}	**Authentication Capability Class**		
File ID: 6FE1	**SFI:** 01	Binary	01 byte
Read AC	Following successful CHV-1 authentication		
Write AC	Following successful CHV-4 authentication		
01	01 02 03	RHUL Mechanism Siemens Mechanism RHUL & Siemens Mechanism	

EF_{IMUI}	**International Mobile User Identity – IMUI**		
File ID: 6FE2	**SFI:** 02	Binary	10 bytes
Read AC	Following successful CHV-1 authentication		
Write AC	Following successful CHV-4 authentication		
01 02-09	The length of the subsequent IMUI IMUI, in ASPeCT, the IMUI is always 9 bytes long and is encoded in ASCII “222010100” – “222010109” Italian Users “228010200” – “228010219” Swiss Users		

EF_{CERTU}	**User Certificate**		
File ID: 6FE5	**SFI:** 05	Binary	134-135 bytes
Read AC	Following successful CHV-1 authentication		
Write AC	Following successful CHV-4 authentication		
01 02-135	Length of the certificate This is the user certificate of his public key. The format is described in Table 6.4 The length is variable because the data is packed data and is therefore dependent on the actual value of the public key.		

EF_{CERTN}	**Network Operator’s Certificate**		
File ID: 6FE6	**SFI:** 06	Binary	Up to 140 bytes
Read AC	Following successful CHV-1 authentication		
Write AC	Following successful CHV-1 authentication		
01 02-140	Length of certificate This is the certificate of the Network Operator’s public key. The format is described in Table 6.4. As in the case of the User’s Certificate the length is variable. The value is written to the card during the authentication process and the card uses the data in this file when it requires the Network Operator’s public key. This is only used if the UIM has successfully verified the certificate.		

EF_{SECU}	**User's Secret Signature Key**		
File ID: 6FE7	**SFI:** 07	Binary	16 bytes
Read AC	Never		
Write AC	Following successful CHV-4 authentication		
01-16	This is the User's secret key, for the EC signature mechanism. The key cannot be accessed externally and is established at personalisation time		

EF_{IPF}	**Internal Public Key File**		
File ID: 6FE8	**SFI:** 1D	Binary	210 bytes
Read AC	Always		
Write AC	Following successful CHV-4 authentication		
01 002-058 059-154 155-210	This file contains the public key of the card in a form that is directly usable by the card. It also contains the domain parameters of the EC. The file is made up of three entries: 0) Header – number of entries 1) Certification Authority Identity and Public Key 2) Domain Parameters of the EC (p, q, A and $\boldsymbol{G}$) 3) Network Operator Public Key		
Notes:	Every card is personalised with the same data for this file. The Network Operator public key is automatically updated during the VERIFY NO CERTIFICATE command, provided it is successful. Each entry is validated using an EDC code before use.		

EF_{CAID}	**Certification Authority ID**		
File ID: 6FE9	**SFI:** 18	Binary	4 bytes
Read AC	Following successful CHV-1 authentication		
Write AC	Following successful CHV-4 authentication		
01-04	In ASPeCT, the IMUI is always 4 bytes long and is encoded in ASCII "3211" Italian Certification Authority ID "3210" Swiss Certification Authority ID		

EF_{IMUN}	**EXODUS International Mobile User Numbers – IMUNs**		
File ID: 6FEC	**SFI:** 11	Linear Fixed	17 records of 10 bytes
Read AC	Following successful CHV-1 authentication		
Write AC	Following successful CHV-1 authentication		
	This file was provided for EXODUS to store the user numbers. The records were not used by ASPeCT.		

EF_{EXODUS}	**Miscellaneous EXODUS Data**		
File ID: 0011	**SFI:** 12	Binary	22 bytes
Read AC	Following successful CHV-1 authentication		
Write AC	Following successful CHV-1 authentication		
01-03 04-11 12 13-22	TMUI: EXODUS Temporary Mobile User Identity LAI/FPAI Local Area Identity / Fixed Point of Attachment Identifier LID Language Identity CPN Calling Party Number		
NOTE	The EXODUS TMUI is not related to the TMUI that is established in the ASPeCT authentication. This version was required by EXODUS in case there were problems with the ASPeCT authentication. Its value is not used by the ASPeCT software.		

EF_{KS}	**Session Key**		
File ID: 6FEE	**SFI:** 1E	Binary	16 bytes
Read AC	Following successful CHV-1 authentication		
Write AC	Never		
	Common session key established during the authentication mechanism		
NOTE	The EXODUS terminal required access to this key after the authentication had been successfully completed. It also proved useful in debugging the authentication mechanism but clearly in a real system it would be impossible to read out this value from the card		

EF_{TLV}	**Operating System Object File**		
File ID: 1000	**SFI:** –	Object File	
Read AC	Always		
Write AC	Never (fixed at personalisation time)		
	File for the location of various operating system constants. It is not directly used in ASPeCT but does store information such as the FCI of the DF_{UIM}. This file is only present on the STARCOS SPK2.1 version of the UIM		

EF_{ISF}	**Internal Secret File**
File ID: –	**SFI:** –
Read AC	Never
Write AC	Following successful CHV-4 authentication
	File for the internal storage of the secret keys. In ASPeCT this file contains only the PIN which is used for user card authentication.

4.3.2.2 Commands

This section describes the different UIM commands that are used in the Authentication application. The commands are only described to the extent that they are used or developed for the ASPeCT application. The appropriate ISO/IEC standard [ISO7816-4] interoperable commands are described first and then the commands specific to ASPeCT.

4.3.2.2.1 SELECT

This is the standard ISO and ETSI SELECT command. In ASPeCT it is only used to select by File ID the DF containing the desired application or the EF for a subsequent READ RECORD or READ BINARY command, and so has the form:

CLA	INS	P1	P2	Lc	Data
A0	A4	00	00	02	*File ID*

where, *File ID*¸ is the 2 byte File ID of the DF or EF being selected.

If the command executes successfully the returned result is

Data	SW1	SW2
File Control Information	90	00

The *File Control Information* is dependent on the file selected and which UIM it is running on. This information is not used by ASPeCT so is not discussed here.

4.3.2.2.2 READ BINARY

This is the standard ISO and ETSI READ BINARY command. In ASPeCT, it is used to read data from some of the application files such as EF_SPID. The version for the STARCOS UIM uses the short file identifier to address the file whereas the GSM version of the UIM cannot do this since the ETSI READ BINARY does not support this feature. There are therefore two versions of the command:

For GSM or ISO cards

CLA	INS	P1	P2	Le
A0	B0	*offset high*	*Offset low*	*len*

and, for ISO cards

CLA	INS	P1	P2	Le
A0	B0	80 | *sfi*	*offset*	*len*

where,

- *sfi* is the short file identifier of the file to read from
- *offset* is the offset into the file from which the read operation commences
- *len* is the number of bytes which should be returned by the card

If the command executes successfully the returned result is

Data	SW1	SW2
file data	90	00

4.3.2.2.3 READ RECORD

This is the standard ISO and READ RECORD command. In ASPeCT it is only used to read data from the record orientated file containing the EXODUS IMUNs. The GSM version of the UIM does not support this command. (It supports the standard ETSI READ RECORD). The format of the command is:

CLA	INS	P1	P2	Le
A0	B2	*rid*	*sfi'* | 04	00

where,

- *rid* is the record ID to read from
- *sfi'* is the short file identifier of the file to read from (left shifted by 3 bits)

If the command executes successfully the returned result is

Data	SW1	SW2
file data	90	00

4.3.2.2.4 VERIFY CHV

This is the standard ISO and ETSI VERIFY command. In ASPeCT, it is used to grant access to the data in the DF_UMTS files.

The format of the command is:

CLA	INS	P1	P2	Lc	Data
A0	20	00	*CHV No*	08	*PIN*

where,

- *CHV No* is the ID of the PIN which is being verified, it can be 01 or 02 for ASPeCT
- *PIN* is the PIN value with trailing space characters to make the length up to 8 characters.

There is no returned data from this command – although if the PIN is incorrect the return status has the form 6C8*X* where the *X* indicates the number of remaining attempts available before the card will block the PIN.

4.3.2.2.5 GET CHALLENGE

The format of the GET CHALLENGE command is:

CLA	INS	P1	P2	Le
A0	84	*mode*	00	00

where,

b7	b6	b5	b4	b3	b2	b1	b0	*Mode*
0	0	0	0	0	0	0	0	Random Challenge, (standard STARCOS)
0	0	0	0	0	0	0	1	Elliptic Curve Diffie-Hellman Challenge
1	0	0	0	0	0	0	1	Elliptic Curve Diffie-Hellman Challenge (Acryl Format)

For both cases, an 8-byte random challenge, *k*, is calculated using the card's internal random number generator. In the normal case this challenge is returned. For the Diffie-Hellman challenge, this random value is used to calculate $k\boldsymbol{G}$, where $\boldsymbol{G}$ is a generator of the Elliptic Curve chosen as a common domain parameter.

The domain parameters are found in the IPF of the currently selected application and must have a Key ID of 82.

If the command executes successfully the returned result is

Data	SW1	SW2
challenge	90	00

The *challenge* is either in the standard representation or Acryl format. The value of *k* is stored in RAM or in EEPROM in the case of the Diffie-Hellman challenge for later use. It is referred to below as RND_U, the random generated by the user.

In the case of an error, the following status words can be returned by the command:

6A 81	SW_P1P2	if P1 or P2 are invalid
6A 88	SW_KEY_NF	if the Domain parameters were not found in IPF
6F 07	SW_KEYPART_NF	if one of the required key components is missing

4.3.2.2.6 COMPUTE HASH

This command is not explicitly used in ASPeCT but the provision of the external interface allows proper testing of the Hash functionality.

CLA	INS	P1	P2	Lc	Data	Le
A0	F0	*mode*	00	*l*	*data*	00

where,

b7	b6	b5	b4	b3	b2	b1	b0	*mode*
								Initialise Hash Calculation
							1	First block – initialise
							0	Subsequent block
								Complete Hash Calculation
						1		Last data block
						0		More data follows
0	0	0	0	0	0			RFU

The length of the input data, *l*, must be 64 bytes for all but the last block of data. The RipeMD128 algorithm is used to calculate the hash value, which is stored in EEPROM, for later use, when the last block is sent.

Note that the internal counter used is only 4 bytes rather than the 8 bytes specified in the RipeMD128 definition. This means that the implementation will only produce correct results on data strings up to 2^{29} bytes. This is not a practical limitation for a smart card implementation.

It is possible for a data block to be both the first and last, so that the hash input is between 0 and 64 bytes.

If the command executes successfully, then when the last block has been processed the returned result will be

Data	SW1	SW2
hash value	90	00

In the case of an error, the following status words can be returned by the command:

65 81	SW_EXECUTION	if there is a problem writing to the EEPROM
67 00	SW_LENGTH	if the value of *l* is not valid

4.3.2.2.7 MUTUAL AUTHENTICATE

This is the fundamental command in the whole of the authentication mechanism. It accepts as input the random from the Network Operator, the authentication token calculated by the Network Operator and an enciphered TMUI. If the authentication token and the random correspond with the challenge previously issued by the card then the card deems the authentication to be successful and it can use the established session key to decipher the TMUI. It then calculates a token, which is returned to the Network Operator to complete the authentication of the card to the NO.

CLA	INS	P1	P2	Lc	Data	Le
A0	8A	*Mode*	00	*l*	$RND_N \| \| AUTH_N \| \| E(K_S; TMUI)$	00

where,

- the *mode* parameter must be AA to designate the ASPeCT Siemens' authentication protocol
- RND_N is the random value generated by the Network Operator
- $AUTH_N$ is the authentication token calculated as $h(K_S)$ by the Network Operator
- $E(K_S; TMUI)$ is the Temporary Mobile User Identity supplied by the Network Operator, and enciphered by the session key K_S

The RND_N and $AUTH_N$ values are both 16 bytes long and the enciphered TMUI is either 8 or 16 bytes long. The command will return a length error if the length not a multiple of 8 and greater than or equal to 40.

The domain parameters and the public key of the Network Operator are both found in the IPF of the currently selected application and must have Key IDs of 82 and 83 respectively. The NO public key is automatically written to this location by the Verify Certificate command.

The card operation of the command is described by the following pseudo-code:

Obtain the NO public key, $s\boldsymbol{G}$ from key ID 83 in the IPF
(s is the secret key, $\boldsymbol{G}$ the common generator of the Elliptic Curve group)
Calculate RND_U ($s\boldsymbol{G}$) using the value of RND_U stored during the GET CHALLENGE command
Convert this value to the Acryl representation and append RND_N
Apply the RipeMD128 hash function to the result: $K_S = h(RND_U\ (s\boldsymbol{G}) \| \| RND_N)$
Save K_S in EF_KS (if the authentication is successful it will be the established session key)
Apply the RipeMD128 hash function: $h(K_S)$
Erase the stored RND_U value, to prevent the challenge being used more than once
Compare $h(K_S)$ with RND_N if they are equal the UIM has authenticated the NO
Decipher the supplied token, $E(K_S; TMUI)$, using K_S to get the supplied *TMUI*
Apply the RipeMD128 hash function, to the concatenation of K_S and *TMUI:* $h(K_S \| \| TMUI)$
Compute an AMV signature of the hash value: $AMVSign(h(K_S \| \| TMUI))$
Convert this value to the Acryl representation
Encipher the Acryl representation using K_S giving $E(K_S;AMVSign(h(K_S \| \| TMUI)))$

Notes

1. Due to an error in the CBC mode implementation in the Acryl library ECB mode was chosen to encipher the *TMUI* and the returned signature. In addition, an oversight in the demonstrator meant that K_S was used as a single length rather than a double length DES key. Neither of these errors were changed for the trial so that the software would remain compatible with the demonstrator. Changing this to the correct operation would be straightforward.
2. The padding mode used for the encipherment is PKCS#1 rather than ISO/IEC 9797 [ISO9797].

If the command executes successfully then the enciphered signature is returned:

Data	SW1	SW2
$E(K_S;AMVSign(h(K_S \| TMUI)))$	90	00

In the case of an error, the following status words can be returned by the command:

63 00	SW_AUTH_NFBZ	$AUTH_N$ did not match: Authentication failed
65 81	SW_EXECUTION	if there is a problem writing to the EEPROM, or if there is an error in a file header checksum
67 00	SW_LENGTH	if the value of *l* is not valid
6A 81	SW_P1P2	if P1 is invalid
6A 82	SW_FILE_NF	if EF_KS or EF_KEYU not found
6A 88	SW_KEY_NF	if the Domain parameters were not found in the IPF
6F 07	SW_KEYPART_NF	if a particular Domain parameter was not found in the IPF

4.3.2.2.8 READ CERTIFICATE

This command allows the UIM's public key certificate to be read out and is used in the authentication process. To support user anonymity the ASPeCT authentication mechanism only requires that the certificate is available when enciphered using DES under the previously established session key, K_S. Note that the certificate can also be read out in plaintext form by using the standard READ BINARY command. This is not a security risk if it is not sent over the air interface and means that the certificate and secret key are available for other purposes. The value of K_S used is the value in EF_KS.

Note that the first byte in the EF_CERTU file is the length of the certificate in bytes but this is not included in the enciphered data.

Since the certificate can be longer than the card's IO buffer, the certificate must be read out in separate parts each of length 96 bytes. This is a handicap of the current Operating System but will also arise if certificates are longer than 256 bytes since 3 byte length fields are not in general use by smart cards or card readers.

CLA	INS	P1	P2	Le
A0	E4	*n*	00	*l*

where,

- *n* denotes which part of the certificate should be read out. Part 0 denotes bytes 0 to 95, part 1 bytes 96 to 192. *n* must be 0 or 1.
- *l* is the amount of the certificate part that should be returned, 0 denotes the maximum

Notes

1. As in the MUTUAL AUTHENTICATE command DES is used in single length ECB mode. However, the data is enciphered fully so that if is changed to true CBC mode then Part 1 will be correctly enciphered (i.e. the chaining will have been correctly taken into account).
2. The last part of the certificate will have been padded before encipherment using the method described in PKCS#1 [PKCS#1].

If the command successfully executes then when the last block has been processed the returned result is

Data	SW1	SW2
Part *n* of *E(KS; CertU)*	90	00

In the case of an error, the following status words can be returned by the command:

65 81	SW_EXECUTION	if there is an error in a file header checksum
6A 81	SW_P1P2	if P1 is invalid
6A 82	SW_FILE_NF	if EF_KS or EF_CERTU not found

4.3.2.2.9 READ IMUI

This command allows the UIM's IMUI to be read out. The IMUI is needed by both EXODUS and ASPeCT. EXODUS require the value in plaintext but for the ASPeCT Authentication protocol the IMUI is returned enciphered using DES under the session key K_S. The value of K_S used is the value in EF_KS and the IMUI is read from the EF_IMUI file.

Note that the first byte in the EF_IMUI file is the length of the IMUI in bytes but this is not included in the enciphered data.

CLA	INS	P1	P2	Le
A0	E6	*mode*	00	00

where,

- *mode* denotes whether the IMUI should be read out in enciphered form (1) or in plaintext (0)

Notes

1. As in the MUTUAL AUTHENTICATE command DES is used in single length ECB mode.
2. The IMUI will have been padded before encipherment using the method described in PKCS#1.

If the command successfully executes then when the last block has been processed the returned result is

Data	SW1	SW2
IMUI or *E(KS; IMUI)*	90	00

In the case of an error, the following status words can be returned by the command:

65 81	SW_EXECUTION	if there is an error in a file header checksum
6A 81	SW_P1P2	if P1 is not 0 or 1
6A 82	SW_FILE_NF	if EF_KS or EF_IMUI not found

4.3.2.2.10 VERIFY CERTIFICATE

This is the second principal part of the ASPeCT authentication after the MUTUAL AUTHENTICATE command. This command verifies the Network Operator's certificate, which has been previously written to the EF_CERTN file on the card. The certificate is verified using the Certification Authority public key, which was loaded into the UIM's IPF at personalisation time. If the certificate is valid then the public key of the Network operator is installed in the UIM's IPF which means that it can be later used to verify signatures in the MUTUAL AUTHENTICATE command.

Note that the current implementation requires that the User and the Network operator use the same Domain Parameters – this is a requirement for the Diffie-Hellman key-exchange mechanism that has been used in the Authentication mechanism.

CLA	INS	P1	P2	Lc
A0	E8	00	00	00

The domain parameters and the public key of the Network Operator are both found in the IPF of the currently selected application and must have Key IDs of 82 and 83 respectively. The NO public key is automatically written to this location during a successful Verify Certificate command.

The card operation of the command is described by the following pseudo-code:

Select the EF_CERTN file containing the Network Operator's Certificate
Perform some rudimentary checking of the certificate format
Calculate the RipeMD-128 hash of the Certificate Information Part, $h(CertInfo)$
Extract the signature from the certificate
Search in the IPF for the public key of the CA, which signed the NO certificate
Verify the signature
Extract the Network Operator's public key from the certificate
Save the Network Operator's public key $s\boldsymbol{G}$ to Key ID 83 in the IPF

There is no data return from the command, so if successful it returns:

SW1	SW2
90	00

In the case of an error, the following status words can be returned by the command:

65 81	SW_EXECUTION	if there is a problem writing to the EEPROM, or if there is an error in a file header checksum
6A 81	SW_P1P2	if P1 is invalid
6A 82	SW_FILE_NF	if EF_CERTN not found
6A 88	SW_KEY_NF	if the CA public key or Domain parameters could not be found in the IPF
6F 07	SW_KEYPART_NF	if a particular Domain parameter was not found in the IPF
98 02	SW_BAD_CERT	if the certificate signature did not verify or if an entry in the certificate is inconsistent

4.3.2.3 Elliptic curve implementation

The ASPeCT Authentication Trial and Demonstrator use an elliptic curve cryptosystem to implement the key exchange and authentication. The digital signature scheme used is the so-called AMV signature mechanism [ISO14888-3]. This section describes how the elliptic curve arithmetic and cryptosystem was implemented on the UIM.

The Elliptic Curve used, was defined over a finite field with prime order, p. In Weierstrass form the curve is denoted:

$$y^2 = x^3 + Ax + B \mod p$$

where, A,B are integers less than p. The order of the Elliptic Curve is also a large prime, q. These values, together with a generator of the curve, $\boldsymbol{G}$, form the Domain Parameters of the system. The Domain

Parameters are stored in the IPF with a key ID of 82, and by changing these values different elliptic curves can be supported. Note, that the implementation does not actually explicitly require the value *B* so that it is not normally stored in the IPF.

For the Trial and Demonstrator the following values for the Domain Parameters were used

Curve Coefficients in Weierstrass form	
A	2
B	40E048944FB5D907AB99F3C5C32F31752
Prime defining field for group operations	
p	1000000000000143B806B28BA2B0C4A16B
Prime order of curve	
q	1999999999999B9F99A450DF6AB46E509
Point on curve with order *q*	
G	[8, D3755C31DFEAB41D24FB7282584D08A7]

4.3.2.3.1 Representation of Points and Integers

There are two different techniques used for the representations of large integers and points on the elliptic curve. Internally, for calculation, the UIM uses a right justified big-endian format. The length of the integer is prefixed with a single byte count containing the number of bytes in the integer.

However, the external software uses a different representation based on bit strings – the Acryl library format. The Acryl library is used to perform all the cryptographic functionality for the Authentication Centre. The Acryl format is essentially a left justified bit string with a prefixed 32-bit quantity denoting the length in bits of the integer.

The representation of points in Acryl is similar but contains a bit more control information.

Clearly, whenever the domain of the data was being changed it was important to ensure that the data was used in the same way. This meant that at various points the UIM needed to convert data from its internal representation to the Acryl format. For example, the session key K_S is calculated from a point, ***P***, on the curve by evaluating a hash function over the Acryl bit string representation of ***P***. This demands that the UIM converts ***P***, to the Acryl format.

4.3.2.3.2 Elliptic Curve Arithmetic

The fundamental operation that needs to be supported to implement an elliptic curve cryptosystem is the addition of two points on the curve to give a third point, which is also on the curve. By repeated use of this operation, it is possible to define the scalar multiplication of a point on the curve by an integer. It is this latter operation which is used extensively in the authentication mechanism.

In order to reduce the number of costly modular inversions the UIM implementation represents a point on the elliptic curve using projective co-ordinates. Only at the end of the scalar multiplication is it necessary to convert the projective co-ordinates to affine co-ordinates. In addition, the use of projective co-ordinates makes the handling of the point at infinity easier.

Affine Co-ordinates	**Projective Co-ordinates**
(x, y)	$(x': y': z')$
Point at Infinity	(1: 0: 0)
where: $x = x'/z'$ and $y = y'/z'$	

The following equations show how to add the points $\boldsymbol{P_1} = (x_1: y_1: z)$ to $\boldsymbol{P_2} = (x_2: y_2: z)$ to give their sum $\boldsymbol{P_3} = (x_3: y_3: z_3)$ when the points are given in projective form and the elliptic curve is in Weierstrass form. Note that all arithmetic expressions are calculated in underlying field.

If $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ are both distinct points not equal to the point at infinity then we have the Addition Formula:

$$x_3 = -su$$

$$y_3 = t(u + s^2 x_1) - s^3 y_1$$

$$z_3 = s^3 z$$

where $s = x_2 - x_1$, $t = y_2 - y_1$ and $u = s^2(x_2 + x_1) - t^2 z$

If $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ are equal then we have the Duplication Formula

$$x_3 = -su$$

$$y_3 = t(u + x_1) - s^3 y_1$$

$$z_3 = s^3 z$$

where $t = 3x_1^2 + A z^2$, $s = 2y_2 z$ and $u = 2s^2 x_1 - t^2 z$

These are implemented in the UIM by routines equivalent to the following pseudocode:

ec_Add(x_1, y_1, x_2, y_2, z)

```
if x2 = Infinity then
        return [x1, y1, x2, y2, z]
else if x1 = Infinity then
        return [x2,y2,x2,y2,z]
else
        s := x1 - x2
        t := y1 - y2
        if s = 0 then
                if t = 0 then
                        return Double(x1, y1, x2, y2, z)
                else
                        return [Infinity, y1, x2, y2, z]
        else
                return ec_Common(x1, y1, x2, y2, z, s, t, x1 + x2)
```

ec_Double(x_1, y_1, x_2, y_2, z)

```
if x1 = Infinity then
        return [x1, y1, x2, y2, z]
else if y1 = 0 then
        return [Infinity, 0, x2, y2, z]
else
        s := 2 y1 z
        t := 3 x1^2 + A z^2
        return ec_Common(x1, y1, x2, y2, z, s, t, 2 x1)
fi
```

ec_Common(x_1, y_1, x_2, y_2, z, s, t, u')

```
u := u' s^2 - t^2 z
y1 := t (s^2 x1 + u) - s^3 y1
x1 := -s u
x2 := s^3 x2
y2 := s^3 y2
```

$z := s^3 z$

Note that in these routines the result $\boldsymbol{P_3}$ is stored in the variables x_1, y_1, z and the point at infinity is represented by having $x < 0$, denoted Infinity above, and $y = 0$.

When calculating $k\boldsymbol{G}$ we remain in the projective co-ordinates and convert to affine as the final step. This is achieved by calculating ($1/z \bmod p$) as ($z^{p-2} \bmod p$) and then scaling the x and y co-ordinates. Fermat's Little Theorem was used rather than an extended Euclidean algorithm because a modular exponentiation routine was already available in the ROM of the smart card.

4.3.2.3.3 AMV Signature Scheme

The elliptic curve arithmetic is used to implement a digital signature using the scheme commonly referred to as the AMV Digital Signature scheme [ISO14888-3]. This is essentially an elliptic curve analogue of the DSA with a minor change in the definition of the secret key, which makes the signature calculation more efficient.

To calculate a signature we need the domain parameters and the secret key. These are stored in the EF_SECU and in the IPF under key ID 82 respectively. To verify a signature we need the domain parameters and public key, which are all stored in the IPF.

Domain Parameters	Secret Key	Public Key
$A, p, q, \boldsymbol{G}$	$1/x \bmod q$	$\boldsymbol{Y} = x\boldsymbol{G}$

The signature scheme requires a function which maps a point on the curve to an integer and for ASPeCT this function was chosen to be the projection obtained by simply taking the x co-ordinate of the point mod q. Namely,

$$f(\boldsymbol{P}) = x \bmod q$$

where $\boldsymbol{P} = (x, y)$.

The signature calculation takes two inputs: the hash value, h, of the message being signed and a randomiser, k, which must remain secret or else an attacker gains information on the secret key. This randomiser is generated using the UIMs internal DES based pseudo-random generator. Currently only an 8 byte random is used – for a real implementation the length of the random should be the same as the length of q. The signature itself comprises two components, (r, s), which have the same length as q.

$$r = f(k\boldsymbol{G})$$

$$s = (r\,k - h)/x \bmod q$$

Note that because the secret key comprises $1/x \bmod q$ the calculation of s does not require any division.

The procedure to verify the signature requires

$$v_1 = s/r \bmod q$$

$$v_2 = h/r \bmod q$$

$$r' = f(\, v_1 \mathbf{Y} + v_2 \boldsymbol{G})$$

The signature is deemed valid if and only if $r = r'$

As in the elliptic curve calculations themselves, the modular inverse calculation for r is performed using Fermat's Little Theorem.

4.3.2.4 Memory requirements

Table 4.2 shows the sizes of the different parts of the code and the filing system. As is usual for such systems there is always a trade off between code size and execution speed and RAM requirements. The latter being the principal restriction in the current implementation.

Implementation Size of UIM Components					
Elliptic Curve Routines		***Authentication Routines***		***File System***	
Support routines	224	Support routines	279	**UMTS Application**	
ec_Add	85	uim_ParseCert	150	Miscellaneous	279
ec_Double	95	uim_GetChallenge	174	Certificates	315
ec_Common	161	uim_MutAuth	469	Public Key File	230
ec_Multiply	166	uim_ReadCert	70	Secret Key Files	97
amv_Sign	84	uim_ReadIMUI	59	EXODUS Data	232
amv_Verify	209	uim_VerifyCert	320	**Total**	**1035**
Total	**1024**	Ripe MD128	670		
		Total	**1957**	**Biometric Application**	
				Voice Template	1080
Grand Total	**5096**				

Table 4.2 - Implementation Size of UIM Components

As expected the major two commands for the authentication, MUTUAL AUTHENTICATE and VERIFY CERTIFICATE, consume the largest amount of space after the RipeMD-128 algorithm, the latter required 192 bytes of lookup tables which partially explains its size.

Also, note that the verify signature routine requires substantially more code than the compute signature routine. This is because most of the code costs are due to the setting up of the inputs for the various elliptic curve routines and more of these routines must be called for a verify than for a signature operation.

4.3.2.5 Mapping of commands onto the authentication framework

Figure 4.1 shows how the commands described above are used to implement the ASPeCT Authentication Framework as used in the Demonstrator and the Trial. The authentication shown is a New Registration, a Current Registration is very similar but does not need to verify the Network Operator’s certificate.

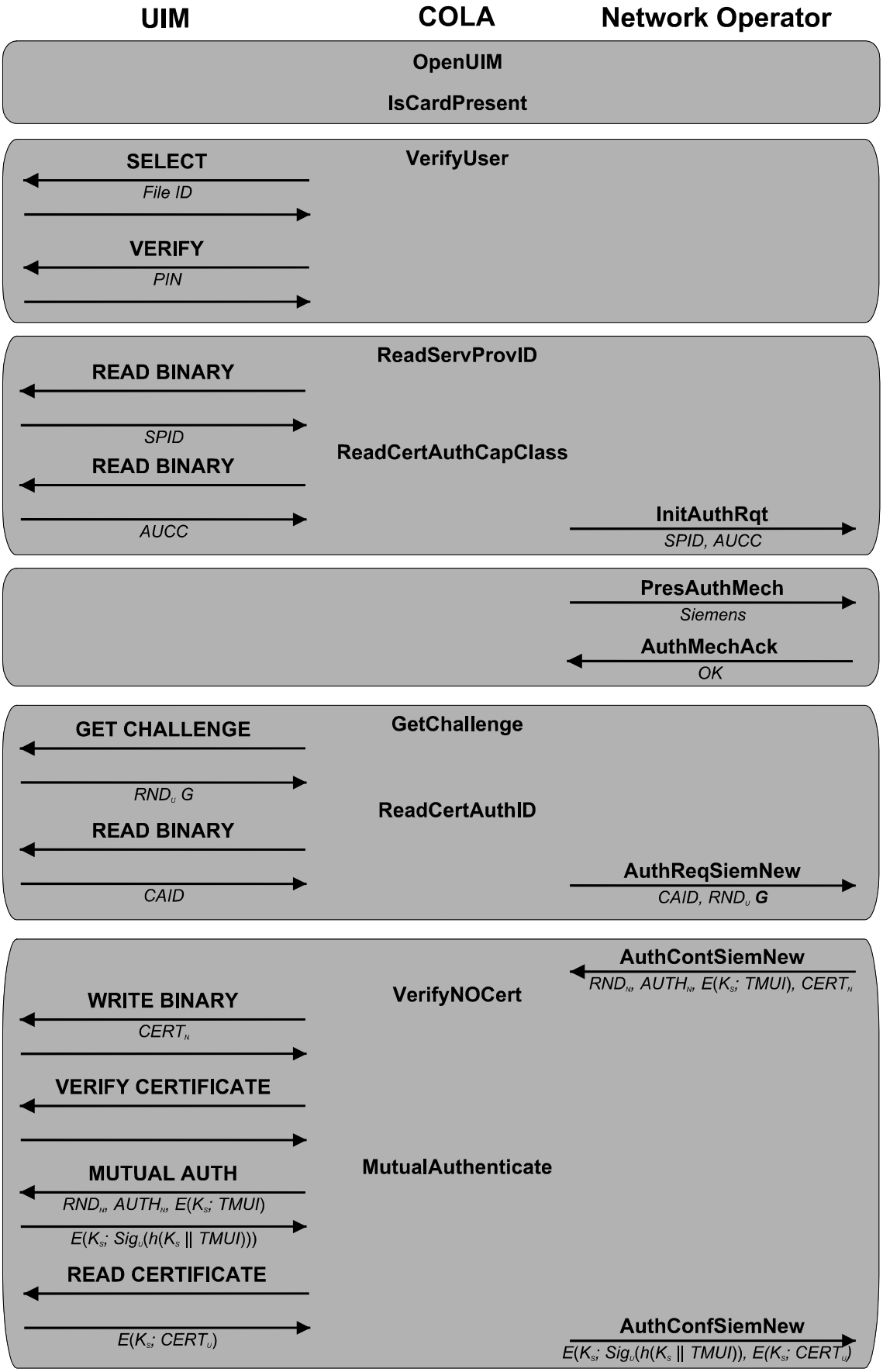


Figure 4.1 - Mapping of Authentication Framework to Commands

4.3.2.6 EXODUS support

In order for the Authentication Trial to take place on the EXODUS platform it was necessary to add some value to the EXODUS project. This was achieved by ASPeCT offering the UIM as a means of supporting user specific data, which EXODUS had originally intended supplying on a floppy disk. In order to support this a specification was drawn up describing the data that EXODUS needed to store on the UIM and the mechanism by which they would access and modify this data. The use of the ASPeCT authentication trial would also add some of the missing security features to the EXODUS platform, such as user authentication.

EXODUS requested support of the data items listed in Table 4.3. This data, including the overhead for files, required approximately 230 bytes on the UIM. Some of the data required by EXODUS was already needed by ASPeCT and this is indicated in the final column of the table.

Description	**Name**	**Access**	**File**	**ASPeCT**
International Mobile User Identity	*IMUI*	R	EF_{IMUI}	Y
International Mobile User Numbers (x 10)	*IMUN*	R/W	EF_{IMUN}	
Temporary Mobile User Identity	*TMUI*	R/W	EF_{EXODUS}	
Local Area Identity / Fixed Point of Attachment Identifier	*LAI/ FPAI*	R/W	EF_{EXODUS}	
Language Identity	*LID*	R/W	EF_{EXODUS}	
Calling Party Number	*CPN*	R/W	EF_{EXODUS}	
Session Key	K_S	R	EF_{KS}	Y

Table 4.3 - Data Support for EXODUS in the UIM

Since EXODUS required either Read Only or Read/Write access to the data, it was sufficient to use standard ISO commands such as READ BINARY to access this data. The exception to this being the IMUI because this was prefixed with a length byte which should be transparent to EXODUS. In this case, the READ IMUI command was used and an extra parameter added which allowed the value to be read out as plaintext.

4.3.3 Vocal biometric application

The section describes the application on the UIM that was developed to support the Biometric Authentication demonstrator. In this application, the UIM is acting merely as a passive carrier of the user template. The template is stored in a file on the card and can be read and written by the Biometric Demonstrator. The card does not perform any processing on the biometric data.

For simplicity, there are no security constraints on accessing the file. Ideally, the Voice Print data could only be updated after a successful user authentication but because the UIM cannot (currently) perform such a biometric authentication this has not been implemented.

4.3.3.1 File system

DF_{VP}	**Voice Print Application**	
File ID: DFF0	**AID:** 'VP'	

EF_{VP}	**Voice Print**		
File ID: `0001`	**SFI:** `01`	Binary	1014 bytes
Read AC	Always		
Write AC	Always		
001-338 339-676 677-1014	Voice Print #0 Voice Print #1 Voice Print #2		
NOTE	The voice prints are of maximum length 336 bytes, each is preceded by a 2 byte field that gives the length of the following voice print.		

4.3.3.2 Commands

The Voice Print application only uses the standard ISO commands READ BINARY and WRITE BINARY, which have been discussed above.

Note that the voice prints are too large to be read or written in one command and so are accessed in multiple commands using the offset into the file to allow different parts to be read. For the STARCOS SPK2.1 implementation no more than 112 bytes can be read or written in any one step – for the demonstrator this is handled transparently by the COLA interface.

4.3.4 COLA interface

In order to ensure that the development ran smoothly, it was important to define a clean interface between the software components between the UIM and the rest of the demonstrator. This interface is provided by COLA, a Windows dynamic link library available in 16 and 32 bit versions.

COLA was initially seen as being a relatively simple conversion layer (COnversion LAyer) between the ASN.1 messages being used in the authentication process and the command/responses pairs suitable for use with the UIM. However, the concept was considerably enhanced and now provides the following features:

- **Abstraction of Card Reader Interface:** The type of chip cad reader being used in the Migration Demonstration or the Trial is hidden from the PC application. The application only needs to make generic procedure calls to COLA, which handles any device dependent features. For example, this means that a card reader without a keyboard can be used and the user can enter the PIN on the PC keyboard rather than on the card reader's keyboard. Other device independent features were added to COLA such as the requirement to poll the reader to ask if a UIM is currently inserted. This was needed for the Trial software.
- **ICC Abstraction:** COLA also provides abstraction of the ICC interface. This is important because it frees the PC application from knowing the details of the ICC command/response format COLA also handles the difference between the GSM based UIM and the STARCOS SPK2.1 UIM. Such differences are not significant to the authentication process and therefore should be transparent to the application software.
- **Configurability:** When it starts up COLA looks for an initialisation file, `cola.ini`, and this file contains the various options needed by the software. If the file or a particular option setting is not present then a default value is used.
  This technique allows the card reader to easily be connected to different COM ports on the host PC, or indeed, more than one reader to be present on the PC, without the need for changes in the application program. Further options control issues such as the various time-outs used in the communication with the card reader or whilst waiting for user input.

- **UIM Emulation:** COLA can emulate the functionality of the UIM for the purposes of the Authentication and Biometric applications. Additionally it emulates the functionality requested by EXODUS. This is achieved by storing the file structure of the UIM in the INI file, where it can be manually changed or changed by the applications using the data. This means that the Demonstration software can be run in an environment without needing a UIM and card reader.
- **Logging / Debugging Support**: COLA is also able to log the data sent between the application and the UIM. This proved very helpful for debugging and during integration with the EXODUS platform. If used in conjunction with the emulation mode then a much deeper level of logging is possible. With a real UIM only the command input and output can be seen and this makes debugging with the main application more difficult.

4.3.5 Personalisation

For the trial, a complete set of UIMs needed to be produced. Each UIM needed to be personalised to contain the necessary data for correct operation. This comprises of two parts, the first being the same for every card and the second being user specific.

4.3.5.1 Fixed personalisation data

- File System – this contains the files described in section 4.3.2.1 above.
- IPF Information – this contains the domain parameters for the elliptic curve cryptosystem and the Certification Authority's public key and identity. These values are fixed for all cards certified by one CA. In ASPeCT there were two possible values: corresponding to the Swiss and Italian Service Providers.

4.3.5.2 User specific personalisation data

- IMUI – this, 9 character value, was provided by EXODUS
- AUCC – since the UIM currently only supported the Siemens' mechanism this value was always `02`.
- SPID – this value is taken from the first 5 characters of the IMUI.
- KsecU – this user specific secret key was generated by ASPeCT
- CertU – this user specific value corresponding to the secret key was generated by ASPeCT and then signed using the appropriate CA key
- PIN – this value was generated by EXODUS and is used to control access to the UIM functionality

4.3.5.3 Personalisation process

The File System for the card was created using the STARMAG tool and converted into an ASPeCT specific template file. This template defines the file structure for the UIM but contained tags to act as placeholders for the various personalisation fields. The actual personalisation data was stored in a structured ASCII text database. The personalisation program then converted the template to a sequence of card commands by replacing the tags with the necessary values from the database. These commands were then sent to the card being personalised. This system is flexible enough to support both the GSM and STARCOS SPK2.1 versions of the UIM by simply replacing the template file.

The cards used as input for this process have already been initialised, that is, all the code required for the UIM application has been already loaded.

The final phase of the personalisation process activates the security mechanisms so that the applications can no longer be modified.

4.3.6 Evaluation of implementation and trial

The implementation of the UIM for the Authentication Trial brought several features of the specification to light which should be modified and considered in the design of similar such systems.

Owing to the fact that extensive trials were not carried out by EXODUS, no evaluation on the special support for EXODUS can be given since these features were not used.

4.3.6.1 Non byte aligned data formats

In two distinct places, it was necessary for the UIM to unpack data that was interpreted as a bit string. This proved to be a problem when these bit strings were not byte aligned because the microprocessor in a smart card is essentially a byte oriented device.

To be able to retrieve the information from such a bit string it is necessary to copy the data elsewhere and then shift it into a byte-aligned format appropriate for subsequent use. This requires both extra memory to store the data, extra processing and extra code when compared to the situation where the bit string was already byte-aligned.

Using a byte-aligned format costs slightly more space due to the extra padding bits required. However, this overhead is normally minimal and the savings in RAM usage, which is a critical resource on a smart card, usually make the benefits well outweigh the cost.

The two places in ASPeCT where this problem arose was the public key and signature components in the ASPeCT certificate format and the occasions where a point on the elliptic curve needed to be represented in Acryl format. By a suitable redefinition of the specification of the certificate and the point representation this problem could be simply avoided.

4.3.6.2 Hash inputs

Another place where small changes to the specification could have a beneficial effect on performance is the choice of data that is input to the hash function.

ASPeCT used the RipeMD-128 hash algorithm; this has a block size of 64 bytes and requires a padding of 9 bytes in the last block. For small quantities of data, the large block size can have a significant impact on performance. For example, processing 56 bytes of data will require the processing of two blocks and so take twice as long as the processing of 55 bytes. Since the hash processing is relatively slow (90 ms per block) this can be significant and we should not unnecessarily increase the number of blocks which need to be processed.

One place, in the ASPeCT implementation, where this arose is in calculating the session key $K_S = h(RND_U \, (s\mathbf{G}) \, || \, RND_N)$. This requires two rounds of the hash function because both the x and y co-ordinates of the point are input to the hash function. From an information theory viewpoint this is unnecessary since the y co-ordinate is only adding 1 more bit of entropy to the hash function input. It would, therefore be acceptable to use only the x co-ordinate and, if considered necessary, the least significant bit or byte of the y co-ordinate. By doing this the input to the hash function would be only one block.

4.3.6.3 Security issues

There were a couple of “features” in the implementation, which have an impact on the security of the system and which should be changed. These were not part of the specification but arose in the implementation due to various historical reasons. They are listed here for completeness.

- The session key K_S should be a 16 bit double length DES key rather than a single length key
- K_S should be used in CBC mode to encipher data rather than in ECB mode

4.4 Conclusion

The Demonstration and the Trial have successfully shown that it is possible to use public key based protocols which have been implemented on a UIM to provide mutual authentication and key agreement for a mobile network in a roaming environment. The performance of the system was acceptable in a trial scenario and the delays introduced by the authentication were comparable to the delays in other parts of the environment. However, for a real system, it needs to be improved. This should be possible for the following reasons:

- Improvements in smart card hardware. The Trial used relatively old devices (which are due to be made obsolete in 1999). Newer devices will be intrinsically faster and have more RAM and ROM.
- Improvements in the implementation. The code in the Trial UIM was prototype code. Based on the experience in developing the software and the availability of more memory in the hardware, faster implementations will be possible. For example, the availability of extra RAM would mean that temporary data does not need to be stored in EEPROM, which causes a substantial performance hit. In particular this could mean that the Network Operator's certificate should be stored in RAM rather than EEPROM and operated on directly.
- Improvements in the specification. Based on the experience gained in the project, changes could be made to the various data formats which would avoid needless format conversions or packing and unpacking of data.

5 Vocal biometrics in UMTS

5.1 Introduction

In this Chapter techniques for password-based speaker verification suitable for the mobile communication environment are developed. More specifically, speaker verification algorithms that generate voiceprints small enough to be stored on currently available smart cards are demonstrated.

We give an overview of the research work done within the scope of this work. Both single- and multiple-password algorithms are discussed. This research work led to the real-time implementation of a best-choice algorithm. Deliverable D23 [D23] dealt with the speaker verification demonstration developed around this implementation. The demonstrator in question was shown in public at the IS&N 98 conference in Antwerp, Belgium, and was well received by its intended audience.

5.2 Background and objectives of the work

The emergence of extensive, affordable, accessible and effective global (mobile) telecommunication services is resulting in an explosive growth in the variety and volume of transactions conducted by electronic means. The safeguarding of the integrity and security of such transactions has a high priority and is being addressed by the developers of such services. ASPeCT is investigating the implementation of security services in general for UMTS (Universal Mobile Telecommunication System), and mutual authentication of user and network, in particular. This includes the authentication protocols of the network and also the authentication of a user to the network.

Currently the smart card acts as a security module and the authentication of a user to the smart card is carried out by means of a PIN (Personal Identification Number). Many users disable the PIN-code because the entry procedure is considered to be too cumbersome. This is especially true in the mobile communications environment. Other people use the same PIN-code for all their smart cards. Eventually these codes are written down or they are observed by others when the are entered. Biometrics is an exciting area of recent technology development that deals with user-friendly automated methods of verifying a person’s identity from one or more behavioural or physiological characteristics. Various biometric techniques are currently being developed and researched, including fingerprints, palm-prints, hand geometry, retinal and iris scans, signature capture and facial and vocal characteristics. Of these, vocal biometric authentication looks potentially the most attractive in the context of the mobile communication services. Not only can it use the existing speech sensors and signal processing power but, additionally, it allows a user interface which naturally fits in with the way the system is operated.

There are two obvious ways to perform vocal authentication – the processing can be performed either locally or remotely. Remote authentication may be appropriate for high security transactions over a telecommunications link. A bank, for instance, is very unlikely to trust the Service Provider to carry out the authentication in case of a fund transfer. In a roaming environment, however, the communication back to a remote authentication server is expensive. Therefore the biometric authentication should be performed locally in the terminal. In addition the proposed authentication mechanism must be portable between different mobile terminals by simply transferring the smart card containing the UMTS user application (the UIM: user identity module) from one terminal to another. This implies that all user specific data must be stored on the UIM. This UIM, being implemented on a smart card, will contain limited memory for template storage and possess limited processing power to perform the comparison.

5.3 Possible approaches to vocal authentication

There are three different approaches to achieve voice authentication in general:

- Free Speech Input: In this case, the subscriber simply uses the mobile equipment and the authentication takes place in the background using his speech as the input. This is the most complicated form of authentication and requires large amounts of data storage and processing power.

- Prompted Text: In this variant the terminal prompts the user to repeat a randomly generated sequence of words out of a limited vocabulary. The characteristics of his response are compared with the responses expected from the correct user. These systems have medium complexity, and need additional text-to-speech software for the prompting of the tokens.
- Pass-phrase: This mechanism relies on only one type of utterance supplied by the user. The advantage of such a system is that the knowledge of the utterance can be used as part of the authentication. The overall protection then stems from what is said and from who says it. Although the storage requirements of these systems are an order of magnitude smaller than in the previous two systems, previous implementations still require more storage than is currently available on smart cards.

Text-prompted and text-free systems have to capture a wide variety of speaker dependent phonetic events, which lead to complex systems. The storage demands and amounts of enrolment data of these systems are far beyond the quantities that are feasible to consider in the present context.

Therefore, a pass-phrase system with small storage requirements was examined during this project. In addition, we aimed at a system requiring only a short enrolment session. This is to keep the system as user friendly as possible. It is clear that the demands of short enrolment sessions and small storage capacity are directly competing with the achievable speaker separation. Multiple passwords might be expected to give better protection, where again, the storage requirements are competing with the achievable accuracy.

The main objective of this work can be summarised as follows:

- The development and implementation of a real-time demonstration of a best-choice verification algorithm that complies with the limitations of a mobile environment and generates voice templates that are small enough to be stored locally on a smart card.

5.4 An introduction to speech recognition technology

The problem of automatic speech recognition (ASR) can roughly be described as the decoding of the information conveyed by a speech signal and its transcription into a set of characters. Global recognition is basically a pattern recognition approach in which speech patterns are stored during a learning phase and recognised via pattern comparison techniques. Patterns can be phrases, words, or sublexical units such as syllables, diphones, or phones. The phonetic approach postulates the existence of a finite set of phonetic units that can be described by a set of distinctive features extracted from the speech signal. Although it should be possible to recognise speech directly from the analogue speech signal, it is common usage to extract features representing the spectral envelope and their delta – the change of the feature vector over time-, for both, the training and recognition process. A speech pattern corresponding to a word or a sentence is made up of a sequence of short-time acoustic vectors. Therefore, when applied to ASR, pattern recognition techniques must be able to compare sequences of feature vectors. A major difficulty associated with this comparison comes from the fact that different occurrences of the same speech utterance, even pronounced by the same speaker, differ in their duration and speaking rate. Since these distortions are mostly non-linear, it is necessary to design efficient time normalisation methods to perform reliable and meaningful comparison. In the Dynamic Time Warping (DTW) approach, the trial-to-trial timing variation of utterances of the same text is normalised by aligning the analysed feature vector sequence of a test utterance to the template feature vector sequence using a dynamic time warping algorithm. More popular is the use of Hidden Markov Models (HMMs) to model the statistical characteristics of the speech signal. A Markov chain consists of a set of states, with transitions between the states. Each state corresponds to a symbol, and each transition is associated a probability. Symbols are produced as the output of a Markov model by the probabilistic transitioning from one state to another. An HMM is similar to a Markov chain, except that the output symbols are probabilistic: in fact, all symbols are possible at each state, each having its own probability. HMM models consist of several states and represent whole word models or sub-word models such as phonemes. Before a speech system can be put to use, it must be trained. Training is generally conducted by first collecting speech samples from a large number of speakers. Feature vectors are then computed for every predefined time-slice of speech (5-20msec). This information is used with a dictionary containing all the words and their possible pronunciations, along with the statistics of grammar usage, to

produce a set of models. At the end of the training process, therefore, there is a set of word or phoneme models, and a dictionary and grammar, all of which make up the recognition database. One of the advantages of phoneme-based recognition is that training data may be shared across words. For example, the [ae] in “cat” and the [ae] in “bat” would both be used for training the [ae] model. When a new word is added to the recogniser, it is not necessary to obtain training data for that word. Instead, one only needs to construct a model for the word by the concatenation of some previously trained phoneme models.

5.5 An introduction to speaker recognition technology.

Speech is a dynamic acoustic signal with many sources of variation. As the production of different phonemes involves different movements of the speech articulators, there is much freedom in the timing and degree of vocal tract movements. Consequently, depending on a number of conditions, a speaker can modify his speech production while still transmitting the same linguistic message. Speaking styles do differ from spontaneous to read speech, they are influenced by stress or emotion, and the variation is more or less speaker specific. Environmental changes also induce intra-speaker variability. Background noise or stress conditions yield an increase in the speakers’ effort and a modification of speech production. These modifications at the speech production level produce acoustic-phonetic variations. Physiological differences (length and shape of the vocal tract, physiology of the vocal folds, shape of the nasal tract) are an important source of variation between speakers. These differences induce acoustic variability. For example, it is well known that the vocal tract length and the vocal tract geometry are different among speakers, and thus, the resulting formant frequencies are related in a rather non-linear fashion for different speakers. A shorter vocal tract length yields higher formant values. Articulatory habits also contribute to the inter-speaker variability. They are generally functions of the speaker’s personality and differ from the dialect type variabilities, which are specific to a broad group of speakers. In speaker dependent recognition systems, the recognition is done on the same speaker as the one used for training. Sufficient training data should be available for each speaker, however.

Speaker recognition is the process of automatically recognising who is speaking by using speaker-specific information included in speech waves. This technique can be used to verify the identity claimed by people accessing systems; that is, it enables access control of various services by voice. Applicable services include voice dialling, banking over a telephone network, telephone shopping, database access services, information and reservation services, security control for confidential information, secure billing and remote access of computers.

Speaker recognition can be further classified into speaker identification (SI) and speaker verification (SV). Speaker identification is the process of determining from which of the closed set of registered speakers a given utterance comes. Speaker verification is the process of accepting or rejecting the identity claim of a speaker. The fundamental difference between identification and verification is the number of decision alternatives. In identification, the number of decision alternatives is equal to the size of the population, whereas in verification there are only two choices, accept or reject, regardless of the population size. Therefore, speaker identification performance decreases as the size of the population increases, whereas speaker verification approaches a constant, independent of the size of the population.

All speaker verification systems use acoustic characteristics extracted from an utterance spoken by the person whose identity needs to be checked. These acoustic characteristics are compared with a model previously trained by the reference speaker during the enrolment session. A way to distinguish between systems is the freedom of the utterance and the way in which the acoustics are modelled. In our context, the utterance will be a password or phrase that was defined during the enrolment session (respectively password systems and text-dependent systems).

Speaker recognition methods can be divided into text-dependent and text-independent methods. The former require the speaker to provide utterances of key words or sentences that are the same text for both training and recognition, whereas the latter do not rely on a specific text being spoken. The text-dependent methods are usually based on template-matching techniques in which the time axes of an input speech sample and each reference template or reference model of the registered speakers are aligned, and the similarity between them is accumulated from the beginning to the end of the utterance. There are a lot of applications in which

predefined keywords cannot be used. Therefore, text-independent systems have recently attracted more attention. Text-independent systems, however, need extensive training sessions, and do in general not reach similar performance as text-dependent systems. Table 5.1 gives an overview of the specifics of each system.

	Enrolment	**Technique**	**Complexity**	**Remarks**
Password	password(s) 3X	Confidence value. SR score	medium	quick enrolment (taping of owners voice)
text prompted	phonetic rich text	Confidence value SR score	high	(extensive enrolment) (tts. Required)
free text	phonetic rich text	no SR	very high	(extensive enrolment) (reliability)

Table 5.1 - System Overview

5.6 Background work at Lernout & Hauspie

In meeting the constraints of commercial products, L&H has paid attention in the past to compact representations for the prompted approach. The password approach needs to model only a specific word as opposed to the "full" acoustic model required in the text-prompted and text-free methods. L&H has previously developed an algorithm that generates speaker specific models out of acoustic only input. The speaker specific passphrase models are generated out of a simple say-in of the utterances. Since the models are build out of the acoustic realisations of a password only, we can build a text-dependent speaker verification system in which the password can be freely chosen by the user. Free vocabulary passwords give added security, since not only the match of the test speaker towards the trained speaker specific model is available, but we can also check the correctness of the passphrase itself. The speech recognition engines available at L&H are equipped with "so-called" garbage scoring. The contrast score of the speech matched towards the speaker dependent model and the garbage model checks for the correctness of a spoken password.

The password are encoded into a few tens of bytes. This algorithm used to make coded speaker specific models is referred to as the "baseline userword algorithm" or BUA. Short enrolment sessions in which the password is uttered – say – three times are sufficient. The storage requirements of the BUA strongly contrast with methods that involve full acoustic models and which require a thousand-fold more of speaker specific storage. It is clear that these lower resolution models cannot get the same accuracy as the full models. In preliminary tests the accuracy of these low resolution models seemed to reach less than half that of corresponding L&H high resolution speaker specific models. The BUA runs on top of one of the L&H proprietary engines, running at various sample frequencies and tuned towards a typical office, car or telephone environment.

The presence of this BUA algorithm was a key element for the project: it justifies further research and development for speaker verification based on this algorithm, not only from a scientific and technical point of view, but also from an industrial and economic perspective, given its speaker specific storage requirements.

Average spectral measures are known to be speaker specific and are a cheap way of performing text-independent speaker verification from extremely long enrolment and verification sessions. L&H has previously developed a mean cepstral feature which operates on small speech segments.

An early test measured achievable error rates of 10 to 15% on systems using only 1 second of text.

5.7 Research work done within the scope of the ASPeCT project

Task 1: Single password speaker verification

Single password verification is achieved by training a word and speaker specific password by means of the Basic Userword Algorithm. In a normal situation, the occasional impostor is not aware of the right

password, so the gross protection of the system comes from the rejection of wrong passwords. Only spoken utterances which pass this first password check will be submitted to a second intrinsic speaker specific check.

Maximisation of speaker-dependence of the BUA.

In the enrolment session the BUA algorithm generates speaker-specific password models from three acoustic repetitions of a word. The separation power of the BUA was further optimised within the scope of this project . Better estimation formulae led to an increase in separation from 73% to 80% for the BUA The influence of the length of a transcription was studied: longer transcriptions give better rejection of out-of-vocabulary words, while shorter transcriptions give better speaker rejection. A best compromise was chosen. The BUA separation power was further studied on different L&H proprietary engines. These studies include the influence of huge versus small basic world models, the influence of basic feature extraction, and the influence of the sample frequency (11 versus 8 kHz). The BUA was adapted in order to get rid of initial transients.

Spectral characteristics

Another measure that is known to be speaker specific is the long term spectral average. In looking for speaker specific features that require only limited amount of storage and are compatible and complementary to the information contained in the normalised score over the password, the word specific mean cepstral coefficients were chosen. The separation power over the word-based average cepstra was further optimised in the scope of the project. Again an expected profile is calculated out of the training session. The mean cepstral feature used is a 12-th dimensional feature vector. Some dimensions are more important than others, however. After applying appropriate weighting to the different coefficients, the separation power increased from 87% to 89%. Application of an L&H proprietary spectral normalisation technique made this spectral feature more robust towards spectral biasing, while keeping the same separation power.

Combinatory logic for multiple criteria

The statistical dependence of the two criteria described above was explored. The two features do not only have another intrinsic equal error rate, but are of different dimensionality. A theory was developed in order to combine both features in an optimal way. Also, it makes a multi-dimensional decision for uncertain data. In principle, for each of the selected features a fuzzy matching score is calculated. The two fuzzy scores are modified in order to take into account the different dimensionality and the different basic separation power. The two modified features are further combined in order to give the fuzzy match of the utterance.

With the application of this theory, we found as separation power for the combined features: 10% for known passwords and 1.2% for unknown passwords.

Anti-password models for utterance rejection

Low cost out-of-vocabulary word rejection is often based on the comparison of the acoustical match with the expected word(s) and with a model that represents “any word”. In the case of password verification, there is only one single expected word, so the general model should be modified towards a so called “anti-password”. This means a model that represents any word except the password. General speech anti-password models could be trained a priori. We did notice however that in our special case of freely chosen passwords, it is favourable to train a dual anti-password simultaneously with the password itself. Generation of anti-passwords models by discriminative training of a general speech model is out of question, since this model would require too much storage. In assembling anti-password models by the selection of models out of a pre-trained pool of general speech HMM models, it was possible to keep the storage need quite low. The equal error rate for impostor trials with bad (randomly chosen) bad passwords and the bona fide users uttering the right password, dropped from 1.2% up to 0.3% .

Real-time implementation and robustness check

Test of a real time implementation of this system showed some crucial failings in terms of robustness. More precisely the system clearly suffered from initial transients, and the environmental adaptation procedure had to be modified in order to be able to work on short utterances only. This gave major improvements on

robustness. The system was implemented on different L&H proprietary engines, of which an 8kHz engine was selected over the 11kHz for implementation. First of all, the CPU load is 25% lower than for its 11kHz version counterpart. Second, the frequency response of the microphones embedded in the mobile terminal is likely to be of a lower standard than this limit.

In order to test the robustness of the system against mobile terminal and microphone swap, a simple simulator was build that models different microphone characteristics and additive noise. The typical frequency response is approximated by applying a numeric filter with a number of poles and zeros ranging from 0 to 5. Five different frequency transfers were picked up in random sequence in order to test the robustness against microphone swapping. It is clear that these different transfers lead to more dissimilarity in the utterances. Consequently, we noticed a doubling of the false rejection rate.

Tests on the real-time implementation revealed as well the need to check for abnormal signal conditions. Abnormal conditions, being “too loud”, “too quiet”, or signals having a “bad signal to noise ratio”, will all have severe impact on the performance of the system. The abnormal signal detection module which was built in is based on a signal peak and background noise tracker. Additional checks were included that check appropriate length of the passwords. Passwords that are too short give bad verification results, while passwords that are too long would require too much storage capacity.

In the real-time implementation , the required password models are synthesised from the simple speaking of the passwords. This enables the user to change his password locally in an autonomous and flexible way in order to avoid possible misuse by a third party. Since no written or typed input is required, this process can run in a completely hands-free and voice-controlled manner.

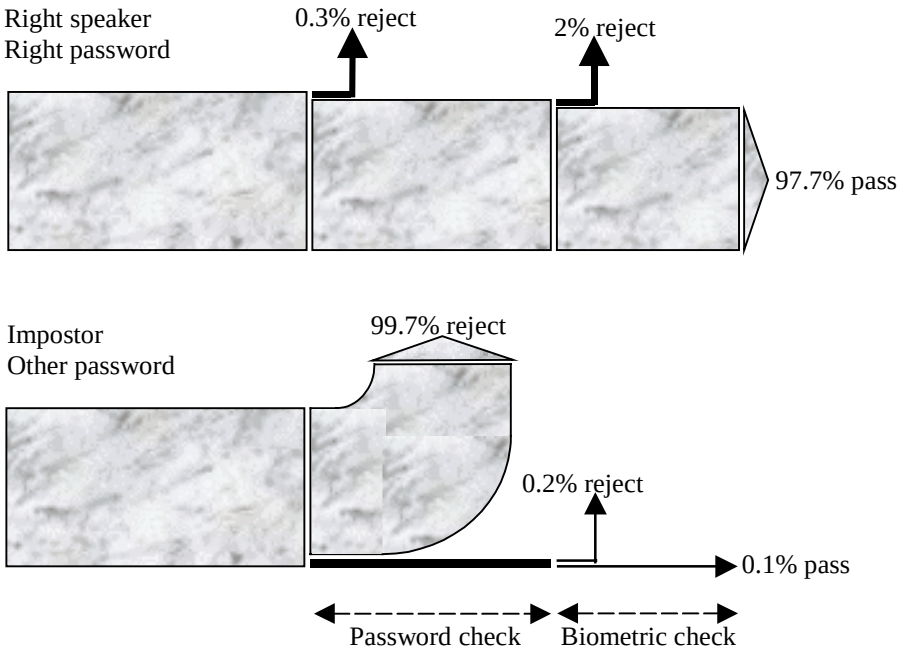


Figure 5.1 - Functional Diagram of the Implemented Single Password System

Figure 5.1 gives the functional diagram of this implemented single password system. By the cascading of the password and speaker check, we were able to construct a system where the bona fide users get accepted with a 97.7%, while impostors who are not aware of the right password get in with only a 0.1% chance. Impostors, aware of the password, would however still get easy access to the system. The multiple password system should give better results here.

D22: Multiple password verification

In the search for a better intrinsic speaker separation, we looked at a multiple password system. It is clear that multiple passwords should give better separation than single password systems. It was not clear

however, to what extent results of single password verifications were independent, and how to make the best combination of single results.

Limiting the vocabulary to a fixed predefined set and using a more speaker-specific basic model set seemed to reduce the intrinsic speaker verification error rate. In the above development, sub-word units of passwords of other speakers model the passwords of one speaker. A methodology was developed in order to make an optimal combination of single password verification results.

This was achieved by associating a certainty measure to each single password trail. This certainty measure is high for scores that are mainly achieved for bona fide users, and varies to low for scores only found with impostor trials. In between these extremes lies a fuzzy zone where the decision is hard to make. A multiple trial system is shown in Figure 5.2. The system gives better separation while not giving in on user friendliness. In general, impostors will need more trials and will therefore be penalised for being inconsistent. Multiple trials of different passwords will in addition be combined in order to make the final decision.

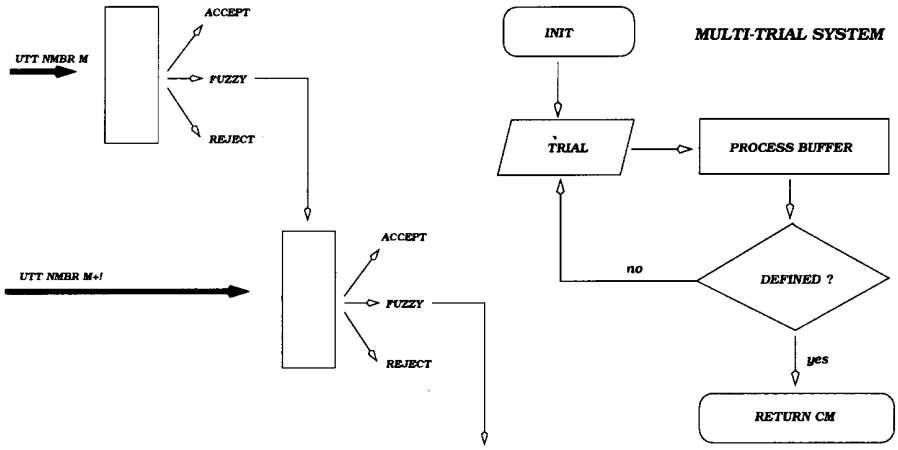


Figure 5.2 - Fuzzy Trial System

Internal deliverable [D22] describes this combinatory methodology.

For a fixed vocabulary system, the intrinsic speaker verification error (same passwords) of a triple trial, dual password system is as low as 1%. For a triple password system it dropped to 0.5%.

This performance worsens towards a 2.3% for a triple trial dual password, where “free passwords” are allowed instead of passwords restricted to a fixed vocabulary.

The influence of microphone swap on the multiple password system was tested as well. In the multiple password mode the microphone swap is responsible for a 50% increase in error rate (3.3% versus 2.3%).

D23: Low storage speaker verification demonstration

The implementation described in [D23] demonstrates the real-time implementation of the multi-password system of [D22]. The demo includes a full training session, the storage of the voiceprints onto the smart card, the retrieval of data from the smart card, and the verification process.

The hardware configuration consists of a PC system with a built-in, or add-on sound card of the Soundblaster type, an external microphone, and an external smart card terminal with the corresponding smart cards. The smart card terminal is connected to the PC via a serial port.

A top layer script was developed invoking the verification engine for the training and test sessions and performing the multi-password scoring. It is interfaced on one side with the Soundblaster card, for the acquisition of samples, and on the other side with the card reader, for the storage and retrieval of the voiceprints.

The demonstration is equipped with a vocally driven user interface, including prompts to insert the cards, and to start a training or a verification session. During the training session, a vocal feedback replays the user's input to give feedback on the recording quality. The quality of the speech recordings is checked as well, and if necessary the user is prompted to change his speaking volume.

To keep the system as user friendly as possible, we opted to incorporate a dual password verification system in this demonstration. More specific, the user will be prompted to give his/her name, together with a user defined password system.

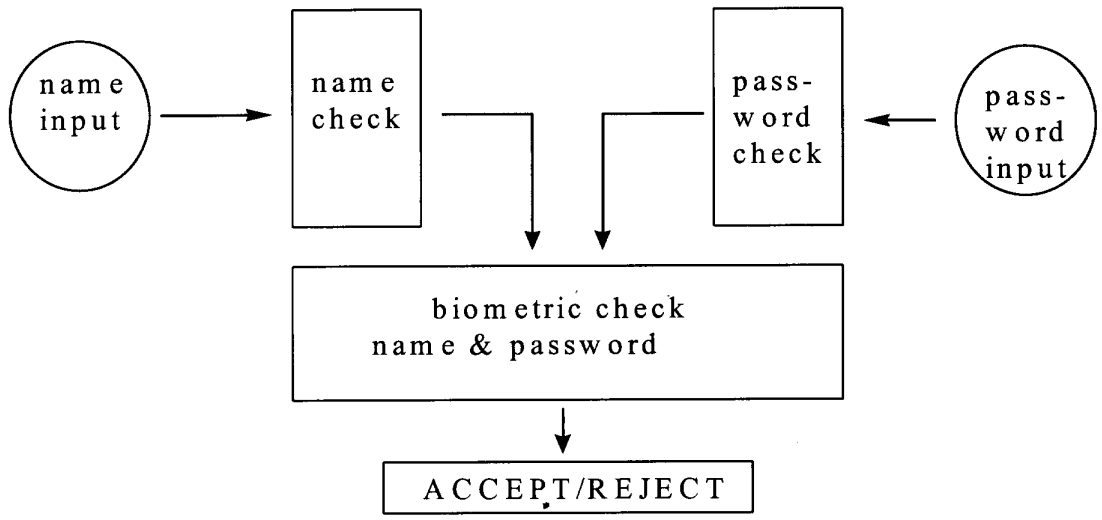


Figure 5.3 - Dual Name/Password System

Within the system, however, the user's name will be treated in completely the same way as the user's password. At training time, the user has the complete freedom to chose the 'so called' name input. He/she can use his/her proper name (first + last name), or use a nickname. A minimum length constraint of 0.8 seconds is used on the speech input. The voiceprints of the name response and the password response will be stored on the smart card. The typical storage is 336 bytes per voiceprint.

At verification time, the user will again be asked to prompt his/her name (or nickname), and his/her password. For each password the user will be given three trials, with possible exit functions after the first and second trial. This is done in order to enhance the user friendliness. In general, bona fide users will need less trials than impostors. A maximum of three trials will be used to check a single password (being the users name or password), and the scores of the different passwords of the complete session will be combined in order to make the final accept/reject decision.

To conclude, the demonstration illustrated the feasibility of the implementation of a multiple password voice verification system, generating voiceprints small enough to comply with the storage constraints of currently available smart cards. All functionality of the verification system was illustrated, going from on-line training of the voiceprints, to the actual storage on the smart card, over the retrieval of a voice print from the card and the user identity check.

5.8 Results from the public demonstration

The demonstration was shown in public at the IS&N98 Conference in Antwerp and at Voice Europe 98 in London. Even in the adverse conditions as found on a technology fair, the system seemed to perform quite well. People were astonished to see the system work, despite the high level of background noise in the demonstration booth. The audience was very enthusiastic about the idea of enrolling their voice on a smart card, and seemed very pleased with the free choice of passwords. During this test, none of the intruders

managed to impersonate the system, even if they had assisted in the training session. Thus they did not only have full knowledge of the password, but were also able to imitate the bona fide speaker's speaking style.

5.9 Extended study: Feasibility study of an algorithmic split

5.9.1 Security aspect

Although the applicability of a vocal biometric authentication was successfully demonstrated in the scope of the project, a major security issue is still holding the system from commercial application. This emerges from the fact that the smart card is at the moment only used as a storage device, while the verification process and ultimate decision are done locally in the terminal. The terminal can however, not be regarded as a trusted third party. The latter is particular true in a mobile communications environment. A secure operation can only be guaranteed if the ultimate decision is taken by the smart card itself. In the case of a simple PIN number verification, the card does the comparison of the given number to the template stored on the card. For security reasons the card never reveals the stored number to the terminal. A similar approach is out of the question in the case of our vocal biometric check, since the smart card is not powerful enough in order to do all of the necessary processing. Therefore it was decided to extend the work of package 2.7 in order to study the feasibility of the split of the algorithm in a client-server architecture. In this new structure the terminal should be the client and the card the server. The algorithm should be split in such a way that the ultimate decision is taken by the card and not by the terminal, since the card is the only trusted party for the service provider.

5.9.2 Overview of current algorithm

At training time the voiceprint is assembled on the terminal using the features extracted from three input utterances. After going through a compression step, it is transferred to and stored on the smart card. The opposite scenario is seen at verification time: the voiceprint is retrieved from the smart card and transferred to the terminal. After inflation of the models the terminal matches the extracted features of the new voice input sample to the models and calculates a general match. This match will result in a possible accept/reject decision, which is sent to the card.

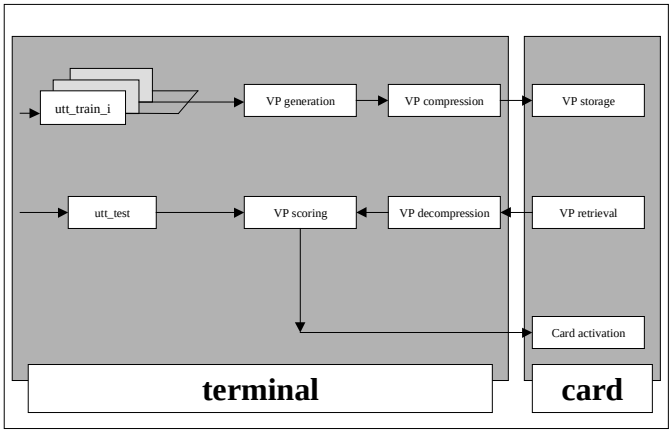


Figure 5.4 - Terminal/Card Functional Split

Since a situation where the training is done on a fake terminal should not be considered, we principally only have to deal with the verification process here. In practise, however, a lot of table data is momentary shared between the training and verification process. We thus have to minimise the unnecessary duplication of Read Only Data blocks on the terminal and card side. Even if we would duplicate all ROM data we would be unable to do the scoring at verification time locally at the smart card.

A situation as of now, where the card can be unblocked by a simple command from the terminal is very insecure. There is no control at all on the usage of fake terminals at verification time. A fake terminal could even send the activation command to the card without requesting any speech input, or needing any additional check.

5.9.3 Specification of current and future Smart card constraints

We consider card configurations for the year 2002:

	High end	Typical
CPU	32 bit	8 bit
ROM	256 K	64 K
RAM	16 K	4 K
EEPROM	128 K	64 K
I/O	115 Kb/s	115 Kb/s

GSM standard : 57.6Kb/s

Current: 9.6K/s

5.9.4 Split of current algorithm

We had to conclude that the current algorithm can not be split over the terminal and the card. First of all the data assembled at training time is stored on the smart card. It is absolutely mandatory that the data sent to the terminal at verification time is not deterministic. Since a major part of the security stems from the knowledge of the password, we cannot reveal the voiceprint to the terminal. Although we think that the chance of assembling correct data input to the card would be small, it is an option that is not absolutely secure.

The decompression and scoring of the voiceprint on the other hand cannot be done on the smart card, since it requires more than one mega byte of ROM storage, which falls beyond card constraints.

5.9.5 Proposal alternative approach

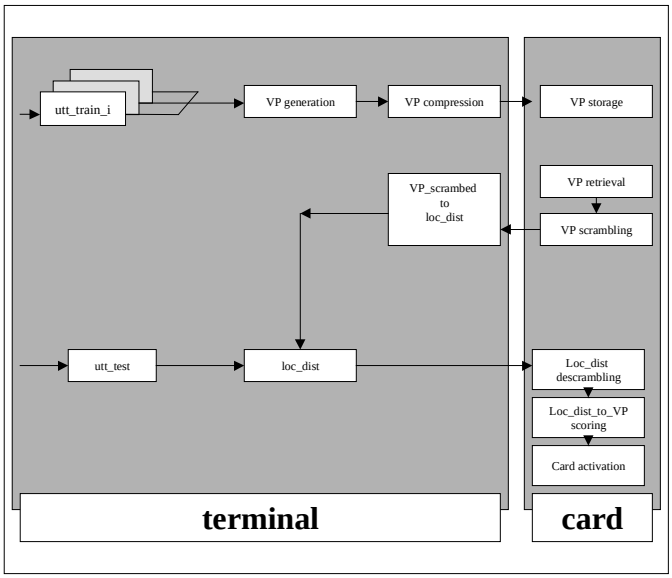


Figure 5.5 - Alternative Terminal/Card functionality split

If we consider the data stream upwards, originating from the data input, there is no danger in extracting all data features on the terminal side, since this processing is totally independent of the stored training data, and consequently could not endanger the security. These features could eventually be sent to the smart card, there is no real I/O problem here.

The ROM data needed in order to score the extracted features to the voiceprint are as high as 260 Kbyte for the basic models and 1 Mbyte for the compression/inflation of the voiceprints. This is far too much in order to be stored on the voiceprint. A first alternative is not to use the compressed models but use models that are about twice as big. A second possibility is that the card polls the terminal for the inflation data. This would need about 2 to 3 K of traffic.

In both options we would need a total of 70000 read operations out of this ROM data per second.

In an alternative embodiment, we would not store the 260 K ROM data on the card, and the compressed VP will be sent to the terminal in scrambled order. The terminal will decompress the voiceprint, without de-scrambling it. Since the actual voiceprint is not known to the terminal, there is no way that the estimated response could be guessed, and the terminal will not be able to make the scoring. It will only get addresses of local data input it needs to read in response to the voice input. If the response stream is sent in 6 bit precision, we would need a 13K byte per second speech. This option looks feasible for a 115K baud communication link. The drawback on this solution now stems from the fact that a hacker would theoretically be able to produce a matching input if and only if he knows the password.

A better solution is the following.

First of all, there is no danger in freeing the anti-password. The anti-password is only needed in order to separate speech from silence and in order to have a rough estimate on the correctness of the utterance. By sending the ATP to the terminal, and scoring the voice input to this anti-password, we could reduce the required data stream by 33%. In the meantime the non-speech part of the input can be dropped. The rest of the voiceprint data should be provided with extra redundant data and will be sent in scrambled order to the terminal. Adding a 50% extra redundant information should not reduce the level of security. The terminal will send all local scores to the smart card. This traffic takes a 13K per second speech.

The estimated averaged delay time is 2 to 3 seconds for a 1 to 2 second speech.

5.9.6 Conclusion algorithmic split

A straightforward algorithmic split will not be able to deal with the constraints put by the specifications of the near future cards. Some alternatives of the current algorithm should enable the card to make the final match and decision, and thus give full password/biometric security.

5.9.7 Additional references

Speech Recognition:

[1] J.C. Jugua & J.P. Haton, Robustness in automatic speech recognition, Kluwer Academic Press. (Overview of speaker recognition based on this work)

Speaker Recognition:

[1] J.C. Jugua & J.P. Haton, Robustness in automatic speech recognition, Kluwer Academic Press.

[2] B.S. Atal : Automatic recognition of speakers from their voices. Proc IEEE, vol 64, no 4, pp 475-487,1976

[3] A.L. Higgings, L.Bahler and J.Porter Speaker verification using ramdomized phrase prompting, Dig Sig Proc, vol 1, pp 869-872, 1986

[4] A.E. Rosenberg & Al. The use of cohort normalised scores for speaker verification. Proc 1992 ICSLP oct 1992, pp 599-602.

[5] F.K.Soong, A.E.Rosenberg,L.R.Rabiner and B.H.Juang. A vector quantisation approach to speaker verification. IEEE 1985, pp. 387-390.

[6] A.E.Rosenberg, Chin-Hui Lee and Frank K Soong. Sub-word Unit Talker verification using Hidden Markov Models. IEEE1990, pp 269-272.

[7] T.Masui and S.Furui. concatenated phoneme models for text-variable speaker recognition. IEEE 1993 pp 391-394

[8] J. Kuo, C.H.Lee and A.E. Rosenberg. Speaker set identification through speaker group modeling. BAMFF ‘92’

[9] M.I. Hannah, A.T. sapeluk, D.I. Damper & I.M Roger, (1993) The effect of utterance length and content on speaker verifier performance. Proc Eurospeech pp 2299-2303, Berlin 1993.

[10] M.E. Forsyth, A.M Sutherland, J.A.Elliott & M.A. Jack (1993). HMM speaker verification with sparse training data on telephone quality speech. Speech Communication , Vol 13, pp 411-416.

6 The use of trusted third parties in UMTS

6.1 Introduction

This Chapter investigates the role of Trusted Third Parties in providing security services in UMTS. Security solutions based on the use of TTPs are seen to be a particularly attractive way of helping to manage the complex trust relationships involved in the provision of end-to-end security. The ASPeCT work therefore aimed to verify the feasibility and acceptability of various TTP-based solutions for end-to-end security by conducting demonstrations and trials.

The ASPeCT work has primarily focused on the role of TTPs in the provision of end-to-end security services such as confidentiality, integrity and authentication over UMTS bearer services. In this context, end-to-end security includes security between any end user of a UMTS bearer service which may include value-added service providers (VASPs) who offer services/information to mobile users.

In our model the end user would register and establish a trust relationship with a TTP who would provide various services to support end-to-end security. This TTP would typically, but not necessarily, be operated by the user's UMTS service provider. In order to establish a trust network, TTPs would then establish trust relationships with each other, perhaps on a loosely hierarchical basis. This would help define trust relationships between end users belonging to different TTPs and thus facilitate end-to-end security. The concept of a trust network may be based on the current network of roaming agreements which exists between GSM operators to facilitate secure inter-network roaming.

Although the focus of the ASPeCT TTP work was on support for security between end UMTS users, the use of TTPs to support security between other UMTS entities was also investigated. For instance, the concept of a TTP can help to model the provision of service-related mutual authentication and key agreement in a UMTS environment where complex trust relationships exist between users, network operators and service providers. Although this work was not directly addressed in the TTP work, the public key mechanism studied and implemented for service-related mutual authentication and key agreement in security migration demonstrator allows a user and a network operator to authenticate each other with the support of an on-line certificate server and off-line certification authorities acting as TTPs.

6.2 Requirements for TTPs in UMTS

The concept of a Trusted Third Party (TTP) with respect to communications security is defined in [ISO10181-1]. It is described as a security authority or its agent, trusted by other entities with respect to security-related activities.

TTP services can be considered to be value-added security services available to various entities in UMTS. TTPs have to be able to offer value with regard to availability, integrity, confidentiality and assurance.

The UMTS role model assumed by the ASPeCT project is defined in [ETSI050103] and is shown in

Figure 6.1 below. It illustrates the relationships between users, subscribers, service providers and network operators.

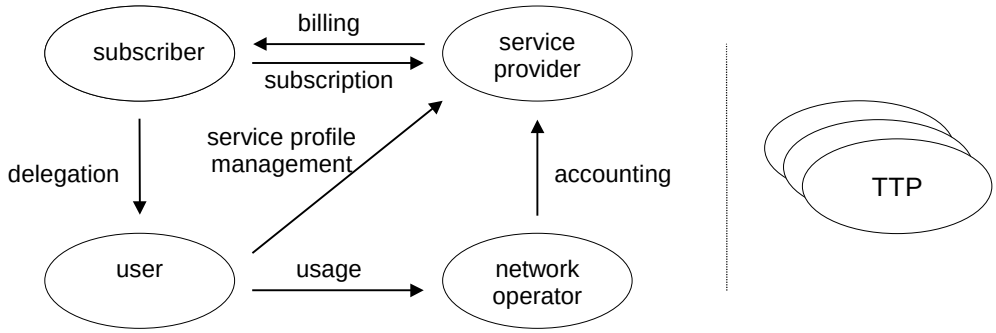


Figure 6.1 - UMTS role model

The TTP(s), shown as external entities to UMTS, may provide security services to any of the entities in the UMTS role model. In addition TTPs may also provide services to national authorities and regulators which interact with all the other parties involved in enforcing the legal and fair operation of the telecommunication service.

One of the advantages of security techniques based on TTPs is that they can help resolve conflicts of requirements between different entities. For example, conflicts exist between the requirements of users for privacy, the requirements of providers for commercial security and the legitimate demand of national authorities for eavesdropping capabilities.

A TTP network for UMTS would have clear benefits in the provision of key management for end-to-end security services between end users by facilitating 'ad hoc' communications between end users with no prior trust relationship or security association. With the emergence of new operators and service providers, there will also be a requirement for well defined trust relationships between the entities such that it is easy for a new organisation to become affiliated with the UMTS system. A TTP network could allow this requirement to be met more easily by eliminating the need for entities to form individual bilateral agreements.

Initially, the motivation to use TTPs in UMTS will be brought about by their use to support security services for which the use of a TTP is essential or at least highly advantageous. A notable example would be the use of a TTP to support the provision of end-to-end encryption services to users, with key escrow functionality to support the demand for warranted interception. The introduction of TTPs to support services like this may increase the perceived benefit of other TTP services and make them more commercially viable.

The full set of TTP functions and services are identified in [D02]. However, not all of these were implemented in ASPeCT demonstrations and trials. Instead the project focused on developing TTP infrastructure to support two security services as outlined below:

- **TTP-based key management for encryption with key escrow** - End-to-end confidentiality is seen as an increasingly important service as users are becoming more likely to transfer commercially sensitive information mobile communication networks. The conflicting requirement of governments to be able to intercept communications demands a suitable key escrow mechanism. The implementation and demonstration of a TTP-based key escrow mechanism was the main focus of the TTP work and is discussed in detail in this Chapter. An outline of a key recovery service is described in [MR98].
- **TTP support for secure billing between users and VASPs** - Secure billing includes overcoming the problems associated with the lack of trust between the participants involved in the billing process. A typical requirement is that a user must be able to find the charging for the services he uses incontestable. This will involve the billing party producing evidence to show the correctness of a bill. TTP infrastructure required to provide a non-repudiation service to support incontestable charging between a user and a VASP was developed in as part of the TTP work. This infrastructure was integrated into the secure billing demonstration and trial. More details on the secure billing work can be found in Section 7.

Some of the more general material in [D02] on the role of TTPs to support secure communications in open networks was submitted to ETSI TC Security. This included a comprehensive set of services which may be provided by TTPs. This information now forms a major part of an ETSI Guide on Requirements for TTP services [ETS97a].

6.3 TTP infrastructure design

The two TTP-based services identified in Section 6.2 and studied in ASPeCT involve the use of TTPs to support security services which are based on public key cryptographic techniques. To provide such support the most fundamental role for TTPs is the generation, distribution and management of public key certificates. Such certification services will become increasingly important in future mobile telecommunication services as public key-based security services become more widespread. For this reason ASPeCT primarily concentrated on the role of TTPs as *certification authorities* (CAs), which certify and manage public keys on behalf of both mobile users and entities they interact with, for example VASPs in the secure billing environment.

Another possibly important role for TTPs in future mobile networks is the support of end-to-end confidentiality services, with key escrow to enable lawful interception or key recovery. Thus ASPeCT also considered the role of TTPs as *key recovery agents* (KRAs), who provide the vital interface between users and interception agents in a key escrow environment.

In the end-to-end encryption demonstrator a protocol based on the JMW architecture [JMW96a, JMW96b] was implemented. In this demonstrator two ASPeCT defined TTPs provide public key certification services to their two users. Based on the certification services, these two users establish a shared key for end-to-end confidential communications and the key shared between two users is escrowed to both TTPs. The key escrow service is implicit in the demonstrator, as either TTP has access to the secret key shared between User A and User B, however no specific interfaces have been provided for an explicit key escrow/recovery service.

In the secure billing demonstrator an on-line TTP provides certificate distribution, provision of certificate chains using cross-certificates, time-stamping, and assurance that a certain certificate was not revoked before a certain time. In addition the TTP provides an assurance to the VASP that the user is authorised by his UMTS service provider to use the VASP's services, such that the risk of the user not paying for the service lies with the user's UMTS service provider and not with the VASP. This authorisation may be implied by assurance of non-revocation. The idea is that a user's certificate will be revoked when he is no longer authorised to use services. Therefore, the user's UMTS service provider can be held liable by the VASP for costs incurred by the user if the VASP can prove that the user's certificate was not revoked and if the user's signature on the charge data can be verified.

In the following subsections we look at some of the TTP infrastructure which is required to support the services provided in both demonstrators.

6.3.1 Certificate design and format

A major part of the work concerning TTP support of security services involves the TTP acting as a CA. An important accompanying task is thus to design and specify suitable certificate formats for use in UMTS. In this section we describe the application of certificates within the ASPeCT demonstrators and describe a special certificate format designed to minimise storage space and use of bandwidth.

6.3.1.1 Certificate applications

In the context of asymmetric cryptography, two entities wishing to communicate securely with one another may need to perform tasks such as verification of the identity of one another, verification of the signature of one another, encryption a message by using the public encipherment transformation of one another, or agreement on a session key by using the public key agreement key of one another. For all these purposes, each entity must obtain the other's public key from a CA. The CA uses a signature algorithm to certify the public key, producing a certificate. This CA needs to have its own digital signature key pair.

A certificate must have the following three properties:

- only an authorised CA can issue a valid certificate;
- any entity with access to the public signature verification key of the CA can recover the public key which was certified;
- no entity other than the CA can modify the certificate without this being detected (certificates are unforgeable).

In the ASPeCT model of future mobile telecommunications systems, the following five types of entity may need to have public keys certified:

- mobile users,
- Network Operators (NOs),
- Service Providers (SPs),
- VASPs,
- TTPs.

ASPeCT TTPs are responsible for the following tasks to support certification of public keys for individual entities:

- generation of a certificate,
- maintenance of a Directory Information Base (DIB), which is used to store certificates,
- management of a Directory Information Tree (DIT), which is used to issue a particular certificate to a particular entity via a suitable certification path,
- revocation of an invalid certificate, and
- generation and maintenance of Certificate Revocation Lists (CRLs).

Here, we are concerned with the generation of certificates, including client certificates, TTP certificates and cross certificates as described below:

- **clientCertificate**: a certificate for the public key of a client (e.g., mobile user, VASP, NO and SP),
- **TTPCertificate**: a certificate for the public key of a TTP (in particular, this TTP is a CA),
- **crossCertificate**: a certificate for another certificate of a public key, which includes both forward certificate and reverse certificate (see Section 6.3.1.5).

For simplicity, we make use of the same data structure for client and TTP private keys as well as for certificates on client and TTP public keys. The main difference with a private key certificate is that it is not necessary to sign a secret key. Instead of the signature, other origin authentication, data integrity and confidentiality techniques will be used.

6.3.1.2 Types of Certificate

We have two types of certificates for ASPeCT, depending on which signature mechanism is used. The leftmost byte of the certificate string is the certificate type identifier.

The first type of certificate makes use of an RSA-signature based on [ISO9796-2] (Draft International Standard). Its certificate type identifier is 00 (hex). Its signed certificate information sequence includes a signed recoverable string and a non-recoverable part.

The second type makes use of an AMV-signature based on [ISO14888-3]. Its certificate type identifier is 01 (hex). Its signed certificate information sequence is the certificate information sequence itself together with an appendix, which is the signature of the sequence.

Each certificate for a public key consists of two parts: certificate type identifier and signed certificate information sequence. Within ASPeCT, these two types of certificates make use of one single certificate information sequence format as described in the following section.

6.3.1.3 Certificate Information Sequence Format

A special certificate format has been designed that minimises the storage space on the smart card and the bandwidth on the air interface. The size of a public-key certificate is less than 200 bytes, which should be compared to 10 Kbytes for typical X.509 certificates [ITU1] (and certificates proposed within IETF). The certificate allows for all necessary information: version number, serial number, issuer identifier, four validity dates (begin and end of validity and two optional dates for usage of the private key), subject identifier, and public key information (algorithm type identifier and a public key value). Other optional fields include key usage, cross certificate attributes, and certificate path attributes. Similar ad hoc certificate formats are being used in the financial sector (e.g., for the EMV specifications [EMV96] by Europay, Mastercard and VISA).

In the detailed description of the certificate information sequence in Table 6.1, *issuer* denotes the entity issuing the certificate authoritatively; and *subject* denotes the entity holding the private key, for which the corresponding public key is being certified.

Field	Contents	Description	Length
1.	Map field	This field gives the map which fields and options will be presented in the certificate. It includes the following 8-bit information:	1 byte binary
		1 public key (1) / secret key (0)	
		2 issuer public key identifier present (1) / not present (0)	
		3 issuer identifier format: hash (1) / plain (0)	
		4 subject private key usage period present (1) / not present (0)	
		5 subject identifier format: hash (1) / plain (0)	
		6 subject key usage present (1) / not present (0)	
		7 cross certificate attribute present (1) / not present (0)	
		8 certification path present (1) / not present (0)	
2.	Version	The version number of the certificate. The version that we will start with is V1.0. This field will therefore contain the value 10 (hex).	1 byte binary
3.	Serial Number	Unique number of the certificate, assigned by the issuer.	12 bytes binary
4.	Public key identifier	Optional. Unique identifier of the public key (e.g., as key updating occurs) to be used to verify the signature on this certificate.	1 byte binary
5.	Issuer Identifier	There are two options for the issuer identifier:	
		1 the binary issuer identifier, like a serial number	16 bytes binary
		2 the plain issuer identifier.	15 + 30 bytes binary
6.	Validity	Including four dates:	
		1 the date before the certificate is not valid,	6 bytes binary
		2 the date after the certificate is no longer valid.	6 bytes binary
		3 optional private key usage period: including the date before and the date after as well. It is used when the subject private key usage period is not the same as the public key validity.	6 + 6 bytes binary
7.	Subject Identifier	There are two options for the subject identifier:	
		1 the binary subject identifier, like a serial number	16 bytes binary
		2 the plain subject identifier.	15 + 30 bytes binary

8.	Subject key usage	The usage of the subject key being certified includes: 0 = digital signature, 1 = data encryption, 2 = key agreement, 3 = key certificate signature, 4 = CRL signature.	1 byte binary
9.	Cross certificate attributes	Optional. Two situations: 1 The public key certified will be used to sign another certificate (issuer identity and certificate serial number). 2 The public key used to signed this certificate was certified by another certificate (issuer identity and certificate serial number).	 30 bytes binary 30 bytes binary
10.	Certificate path attributes	Optional. Two subfields: 1 Path length - the number of related certificates. 2 A list of subject identifiers included in the certificate path.	 1 byte binary 16 byte binary per each subject name
11.	Subject Public key information	An algorithm type identifier plus a public key value for the subject. Subfield 1: algorithm type identifier……..…. 0 = RSA 1 = elliptic curve 2 = Diffie-Hellman other unspecified If algorithm type identifier = 0 Subfield 2: modulus length of key in bits……..….. Subfield 3: exponent length of key in bits……..…. Subfield 4: key value: first modulus, then exponent of key……..….. If algorithm type identifier = 1 Subfield 2: length of x-coordinate of key in bits…… Subfield 3: key value: first x-coordinate, then y-coordinate of key…..….. Subfield 4: parameter set identifier……..... If algorithm type identifier = 2 Subfield 2: length of key value in bits…….... Subfield 3: key value: g^x mod p……..... Subfield 4: parameter set identifier……..	 1 byte binary 2 bytes binary 2 bytes binary (sum of (values of subfields 2 and 3)+7)/8 bytes binary 2 bytes binary 2 * ((value of subfield 2+7)/8) bytes binary 1 byte binary 2 bytes binary (value of subfield 2+7)/8 bytes binary 1 byte binary

Table 6.1 - A certificate information sequence format

6.3.1.4 Signature Mechanism

Within ASPeCT we have two models of signature used to sign a certificate information sequence. The related certificate type identifiers are assigned as 00 and 01, respectively.

6.3.1.4.1 RSA-signature based on [ISO9796-2]

Assume that there exist a public modulus *n* of *k* bits ($512 \leq k \leq 1024$), a private signature exponent *s* and a corresponding public verification exponent *v*. These values may be different for each certificate, and must have the usual properties required for operation of the RSA algorithm, namely that *n* is a product of two

large prime numbers *p* and *q*, that *v* and *(p-1)(q-1)* are relatively prime, and that *sv* is congruent to 1 modulo *lcm((p-1)(q-1))*. Note that the two primes, *p* and *q*, should be discarded, and never revealed after key production.

Suppose that a certificate information sequence *M*, as described in the above section, is of *m* bits. There are two possible cases:

1. the entire message can be recoverable from the signature;
2. the message shall be split into two parts: a non-recoverable part M_x of *x* bytes, where x is a positive integer, and a recoverable part M_r of *m-8x* bits.

We make use of the RIPEMD-160 hash function [DBP96] to compute a hash-code *H* of 160 bits from the entire message *M*. The recoverable string S_r of *n* bits is then constructed as shown in Table 6.2.

Left adaptation	More-data bit	Padding field	Data field	Hash-code field	Trailer
two bits	one bit	one or more bits	m bits or m-8x bits	160 bits	1 or 2 bytes
01	0 if M_x empty 1 otherwise	0…01	*M* if M_x empty M_r	*H*	'BC' or 'XYCC'

Table 6.2 - A recoverable string S_r

If the hash-function in use is either implicitly known or coded inside the message, then the trailer shall consist of one byte set at 'BC'. If the rightmost byte is set at 'CC', then the trailer shall consist of two bytes where the leftmost byte is the hash-function identifier. For example, '31' denotes RIPEMD-160.

Note that some bits of this string have to be changed in the following way.

- The leftmost nibble shall remain unchanged.
- Every subsequent nibble equal to '0', if any, shall be replaced by a nibble set to 'b'.
- The first subsequent nibble not equal to '0' is the border nibble: it carries the border bit; it shall be exclusive-ORed with 'b'.
- The remaining bits shall remain unchanged.

We then sign the recoverable string S_r , in both cases using the signature function under control of secret signature key. The signature is $Sign_s(S_r) = S_r^s \bmod n$.

The signed certificate information sequence shall be either

- the signature alone, if M_x is empty in the first case, or
- the non-recoverable part M_x together with the signature in the second case.

A certificate shall be as shown in Table 6.3.

	Certificate type identifier	*Signature*	*Non-recoverable data*
	one byte	*k* bits	*8x* bits
first case, *k+8* bits	*0*	$Sign_s(S_r)$	
second case, *k+8x+8* bits	*0*	$Sign_s(S_r)$	M_x

Table 6.3 - Certificate format based on RSA-signature

The verification process is a reversed procedure of the signature.

6.3.1.4.2 AMV-signature based on [ISO14888-3]

This signature mechanism is an ElGamal-type signature scheme [ElG85] based on elliptic curves over finite fields.

In the description of the signature mechanism, we make use of the following notation.

ε a finite commutative group

$\#\varepsilon$ the cardinality of ε

p a divisor of $\#\varepsilon$

g an element of order p in ε

X secret signature key, $0 < X < p$

Y public verification key, $Y = g^X$

f a map from elliptic curve points to Z_p

h a hash function

Suppose that a certificate information sequence M is of m bits. The signature equation (the Agnew-Mullin-Vanstone equation) is $RK-SX-H = 0 \bmod p$, with:

K a random number

R the first part of the signature, $R=f(g^K)$

H a hash-code, $H=h(M)$

S the second part of the signature, $S=(RK-H)X^{-1} \bmod p$

Σ the signature, $\Sigma=(R,S)$

The length of the certificate depends on the length of p. Annex C.2 of [ISO14888-3] gives an example with parameter length 129 bits and hash value of length 128 bits. Assume that these numbers can be used in our certificate, the certificate shall be as shown in Table 6.4.

Type of certificate	Length of Certificate information sequence (in bits)	Certificate information sequence	Length of R and S (in bits)	Appendix
one byte	2 bytes	m bits	2 bytes	256 bits
1	L_M	M	L_R	R,S

Table 6.4 - Certificate format based on AMV-signature

The verification process is as follows.

$[g^K]' = Y^{(S/R) \bmod p} G^{(H/R) \bmod p}$

$R' = f([g^K]')$

If $R'=R$, the signature is verified successfully.

6.3.1.5 Cross-certificates

One of the TTP security support services provided by the secure billing demonstrator involves the establishment of cross-certificate chains. Such chains are required when parties in the protocol do not have the same TTP, or when the parties do not have on-line access to their TTP. Here *CertChain(X, Y)* consists of a sequence of certificates, *c0, c1, …, cn*, where the signer of certificate *c0* is the Certification Authority (CA) of entity X, the subject of *ci* is equal to the signer of *ci+1* ($0 \leq i < n$), and the subject of certificate *ci* is entity Y. Such a certificate is verified starting with *c0* (using the public key of the CA of entity X); this guarantees the public key required to verify *c1*, etc. The verification is completed after verification of *cn*. In

order to speed up the verification process and reduce the communication overhead, the CA of entity X might also verify the complete chain, and then create a new certificate for entity Y. However, this provides slightly different guarantees to the entity verifying the cross-certificate. More details on this can be found in the next section.

6.3.2 Certificate management issues

In this section we stray from the TTP scenario implemented in the ASPeCT demonstrators, and consider a number of alternative TTP scenarios, and how they might be implemented. We first take a more detailed look at certificate chains, and then consider how to apply them.

6.3.2.1 Definition of certificate chains

One of the TTP security support services provided by the secure billing demonstrator involves the establishment of (cross-) certificate chains. Such chains are required when parties in the protocol do not have the same TTP, or when the parties do not have on-line access to their TTP. We denote by CertChain(X,Y) a certificate chain that allows X to verify the public key of Y.

There would appear to be two main ways to interpret the definition of the certificate chain CertChain(X,Y).

1. The first interpretation is, as one would expect from the term 'chain', that CertChain(X,Y) consists of a sequence of certificates, c_0, c_1, …, c_n, where the signer of certificate c_0 is CA_X (the Certification Authority of entity X), the subject of c_i is equal to the signer of c_{i+1} ($0 \le i < n$), and the subject of certificate c_n is entity Y. Verification of this certificate chain by X is straightforward:

   X verifies c_0 using the Public Key of CA_X,

   X verifies c_1 using the Public Key recovered from c_0,

   …,

   X verifies c_{i+1} using the Public Key recovered from c_i,

   …,

   X verifies c_n using the Public Key recovered from c_{n-1}, and thereby obtains a verified copy of the Public Key of Y.

2. The second interpretation can only work where the TTP T providing the CertChain is equal to the CA of entity X, i.e. $T = CA_X$. In this case we can provide an alternative interpretation based on possible use of field 10 of the Certificate Information Sequence Format in Table 6.1, and, in this interpretation, the CertChain consists of a single certificate, newly signed by the TTP.

   The TTP first assembles a chain of n+1 certificates, exactly as in the first interpretation immediately above. Note that, since we assume that $T = CA_X$, the first certificate in the chain is one that was signed by the TTP itself. The TTP then verifies the certificate chain (as in the previous interpretation), and thence obtains a verified copy of the public key of Y.

   The TTP now signs a new certificate, the subject of which is the public key of Y. In this certificate, field 10 is defined to contain a list of n names, namely the subjects of certificates c_0, c_1, …, c_{n-1}, or, equivalently, the identities of the signers of certificates c_1, c_2, …, c_n. This certificate is defined to be CertChain(X,Y).

The advantage of the second approach is that the recipient of the CertChain only has to verify one certificate; this may be a non-trivial advantage of the recipient has limited processing power, as may be the case for U in the Payment Protocol. Moreover, no information is lost about the trustworthiness (or not) of the public key, since the information about the certificate chain used in producing this single certificate is recorded in field 10. Of course, the second approach can only be used when $T = CA_X$, but this applies in two important special cases of the Payment Protocol.

It would seem sensible to use the second interpretation wherever possible. This creates additional work for the TTP, but also minimises the certificate verification workload for the user and VASP. It also demonstrates the usefulness of field 10 of the certificate format.

6.3.2.2 Alternative TTP scenarios

In the secure billing protocols implemented in the ASPeCT demonstrators a TTP acts as CA in a communication between a user *U* and a VASP *V*. It is assumed in the implemented protocols that the TTP *T* is the CA of user *U*. From a theoretical point of view this approach seems to be fine. Yet there may be cases where the VASP *V* cannot communicate with the user's TTP (due to communication failure). Also, it may be preferable for *V* to communicate with another TTP (whenever User and VASP are in a completely different domain), which might be the CA that the VASP is certified by, or another TTP. We consider the implications for certificate chains of a number of options.

6.3.2.2.1 Four simple scenarios

Suppose that there exist two TTPs, T_1 and T_2, and that the user and VASP can be registered with either, or neither of these. However, only T_1 is implemented on-line in support of the secure billing protocol. Thus, when T_1 receives a message from a VASP *V* in connection with a user *U*, there are four possible cases to consider:

1. *U* and *V* are both certified by T_1, i.e. $CA_U = CA_V = T_1$. As part of the protocol, *V* will supply Cert*V* (signed by T_1). In such a case the three CertChains supplied by T1 are as follows:
   - CertChain(*U*,*V*) = Cert*V*,
   - CertChain(*V*,*U*) = Cert*U*, (as originally generated by T_1),
   - CertChain(*V*,*T*) = the null string (since $CA_V = T_1$).
2. *U* is certified by T_1 and *V* is certified by T_2, i.e. $CA_U = T_1$ and $CA_V = T_2$. As part of the protocol, *V* will supply Cert*V* (signed by T_2). In such a case the three CertChains supplied by T1 are as follows:
   - CertChain(*U*,*V*) = a single certificate, signed by T_1, created by 'concatenating the two certificates: (CertT_2, Cert*V*), where CertT_2 is a 'cross-certificate' signed by T_1,
   - CertChain(*V*,*U*) = the certificate pair (CertT_1, Cert*U*), where CertT_1 is a 'cross-certificate' signed by T_2,
   - CertChain(*V*,*T*) = the cross-certificate CertT_1 (signed by T_2).
3. *U* is certified by T_2 and *V* is certified by T_1, i.e. $CA_U = T_2$ and $CA_V = T_1$. As part of the protocol, *V* will supply Cert*V* (signed by T_1). In such a case the three CertChains supplied by T1 are as follows:
   - CertChain(*U*,*V*) = the certificate pair (CertT_1, Cert*V*), where CertT_1 is a 'cross-certificate' signed by T_2,
   - CertChain(*V*,*U*) = a single certificate, signed by T_1, created by 'concatenating the two certificates: (CertT_2, Cert*U*)[1], where CertT_2 is a 'cross-certificate' signed by T_1,
   - CertChain(*V*,*T*) = the null string (since $CA_V = T_1$).
4. *U* and *V* are both certified by T_2, i.e. $CA_U = CA_V = T_2$. As part of the protocol, *V* will supply Cert*V* (signed by T_2). In such a case the three CertChains supplied by T1 are as follows:
   - CertChain(*U*,*V*) = Cert*V*,
   - CertChain(*V*,*U*) = Cert*U*, (as originally generated by T_2)

* CertChain(*V,T*) = the cross-certificate CertT_1 (signed by T_2).

One problem that arises with this fourth case is that *U* does not have a valid copy of T_1's public key which needs in order to verify the signature of the TTP (see protocol description in Section ??). So, the VASP should pass to *U* the CertChain(*U, V*) = Cert*V*, and the CertChain(*V, T*) which will enable *U* to verify the signature using a certified, by its associated TTP T_2, copy of the public key of T_1.

Another problem is how T_1 obtains Cert*U* as T_1 in message *M2* receives only the *id* of the user and not its certificate. In that case T_1 has to contact either a central database where Cert*U* might be kept or *U*'s CA. Yet, if the user in the first message passed its Certificate (Cert*U*) to *V* which in turn can pass it to T_1 in message *M2*, T_1 can easily gain possession of the Certificate of the user. This way we avoid the communication that is required, for getting Cert*U*, between T_1 and a database or between T_1 and T_2.

6.3.2.2.2 Other scenarios

A couple of other TTP scenarios have been considered within ASPeCT.

In one variant, we assume that the on-line TTP is that of *V*. *U* passes his Certificate to *V*, who in turn passes it to the *TTP* instead of *CertV*. According to this scheme there is no need for the *TTP* to pass to *V* the CertChain(*V, T*) and there are only two possible cases that we have to examine.

1. U is certified by T_1:

- CertChain(*U, V*) = Cert*V*
- CertChain(*V, U*) = Cert*U*, (as originally generated by T_1),

Both entities are in possession of the public key of the T_1 so there is no need for the TTP to pass any valid certificate to *V* or *U*.

2. *U* is certified by T_2:

- CertChain(*U, V*) = the certificate pair (CertT_1, Cert*V*), where CertT_1 is a 'cross-certificate' signed by T_2.
- CertChain(*V, U*) = a single certificate, signed by T_1, created by concatenating the two certificates: (CertT_2, Cert*U*) where CertT_2 is a 'cross certificate' signed by T_1.

There is no reason for the TTP to construct and send the third CertChain(*V, T*) as *V* is certified by T_1.

In another variant we try to describe the case where the TTP *T* is not the CA of either *U* or *V*. This is likely to happen in cases where *V* can not communicate with its own TTP. That is, in the worst case scenario, where *U* is using these services abroad and *V* cannot communicate with its own TTP (due to communications failure), then *V* would have to communicate with a third TTP which is not the CA of either *U* or *V*.

Because *T* is not the TTP of either *U* or *V*, it should supply *U* and *V* with a valid copy of its public key. This copy should be certified by the CAs of *U* and *V* which makes the whole protocol a rather complicated one. The TTP has to supply *V* with all the Certificate-Chains required by *U* and *V*. Assuming that T_1 is the CA of *U* and T_2 is the CA of *V* the CertChains passed to *V* are as follows:

- CertChain(*U, V*): the certificate pair CertT_2, Cert*V* where CertT_2 is a 'cross-certificate' signed by T_1,
- CertChain(*V, U*): the certificate pair CertT_1, Cert*U* where CertT_1 is a 'cross-certificate' signed by T_2,
- CertChain(*V, T*): the cross-certificate Cert*2T* (signed by T_2),
- CertChain(*U, T*): the cross-certificate Cert*1T* (signed by T_1).

The VASP *V* passes to *U* CertChain(*U, V*) and CertChain(*U, T*) which enables *U* to verify Cert*V* and the signature of the TTP.

This case is much more complicated than the previous described because it requires the possession of cross-certificates among the various TTPs and the transmission of additional data (*CertChain(U, T)*) which may be critical for the performance of our system.

However, it covers all the previous schemes described as:

If $T = T_1$

- CertChain(*U, V*): the certificate pair CertT_2, Cert*V* where CertT_2 is a 'cross-certificate' signed by T_1 (if $T_2 = T_1 = T$ then CertChain(U, V) = CertV because CertT_2 = the null string),
- CertChain(*V, U*): Cert*U* (because CertT_1 = the null string),
- CertChain(*V, T*): the cross-certificate Cert*2T* (signed by T_2) (if $T_1 = T_2 = T$ then this chain can be omitted as CertChain(V, T) = the null string)
- CertChain(*U, T*): the null string (and can be omitted).

If $T = T_2$

- CertChain(*U, V*): Cert*V* (because CertT_2 = the null string),
- CertChain(*V, U*): the certificate pair CertT_1, Cert*U* where CertT_1 is a 'cross-certificate' signed by T_2 (if $T_2 = T_1 = T$ then CertChain(*V, U*) = Cert*U* because CertT_1 = the null string),
- CertChain(*V, T*): the null string (and can be omitted),
- CertChain(*U, T*):. the cross-certificate Cert*1T* (signed by T_1) (if $T_1 = T_2 = T$ then this chain can be omitted as CertChain(*U, T*) = the null string)

6.3.3 TTP Software architecture

In this section we give an overview of the software architecture adopted in the design and development of the TTP services in the ASPeCT demonstrators.

6.3.3.1 Overview of software architecture

Figure 6.2 illustrates the high-level architectures of the TTP server and client software used to construct the demonstrators.

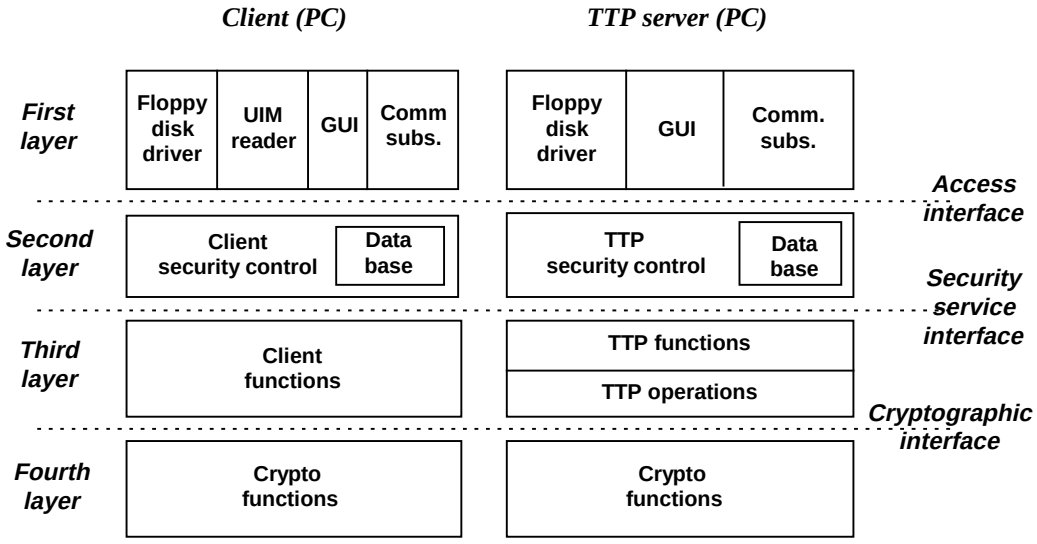


Figure 6.2 - Overview of a TTP server and its client

Note that the internal structure of the client suggested here is outside the scope of the TTP definition.

Note also that from the software architecture point of view, there may exist direct interfaces between the TTP security control block and the client security control block, and between two TTP server's security control blocks. These interfaces allow two security control blocks have an access to each other.

There are three types of application interfaces between a TTP server and its client and between two TTP servers, namely a **local link** (or called a portable link) including *floppy disk driver - floppy disk driver* and *GUI-GUI*, a **remote link** including *communication subsystem - communication subsystem*, and a **security service access link** which is *security control - security control*.

6.3.3.2 Architecture of TTP server software

The architecture of the TTP server software has the following four layers.

(1) The first layer is the ***external communication layer*** consisting of three functional blocks.

- *Floppy disk driver*. This block includes reading data from and writing data to a floppy disk.
- *GUI*. This block includes both the graphical client interface package and the graphical security manager interface package.
- *Communication subsystem*. One example of this block is the communication application running over TCP/IP.

(2) The second layer is the ***TTP security control layer*** consisting of two functional blocks.

- *TTP database*. As defined in Section 9 of [D02], the TTP database should securely store the following three types of security related information:
  1. cryptographic keys, including private signature key(s) for the TTP, and secret keys shared with other TTPs (e.g. used for establishing shared session keys between users);
  2. user information, including key escrow information for a user (possibly including secret or private keys belonging to a user), account/credit information for a user, policy information for a user, and access control information for a user; and
  3. event information, e.g. as used to support audit, fraud detection and alert management functions.
- *TTP security control*. This block will respond to a client's security requirements, by managing a group of TTP security functions, and preparing related security information using the TTP data base.

(3) The third layer is the ***TTP function and operation layer*** consisting of two functional blocks.

- *TTP functions*
- *TTP internal operations*

(4) The fourth layer is the ***cryptographic function layer*** with one functional block.

- *Security algorithms*. This package is implemented behind a Generic Cryptographic Service - Application Programming Interface [GCS-API].

Between the first and second layers, there is a TTP service access interface. This interface will support interactions between the TTP security control and the floppy disk driver, the GUI and the communication subsystem.

Between the second and third layers, there is a *TTP security service interface*. This interface takes into account [GSS-API], Section 9 of [D02], [ISO 11770-1] and [TR 14516-2].

Between the third and fourth layers, there is a *Cryptographic Interface*. This interface is based on [GCS-API].

6.3.3.3 Architecture of TTP client software

The architecture of the TTP client may also have four layers.

The first layer is the *external communication layer* consisting of four functional blocks, namely the three described above together with the UIM reader.

The second layer is the *client security control layer* consisting of two function blocks, the client data base and client security control.

The third layer is the *client security function layer* with one functional block.

The four layer is the *security algorithm layer* with one functional block.

Between the first and second layers, there is a *client service access interface*.

Between the second and third layers, there is a *client security service interface*.

Between the third and four layers, there is a *cryptographic service interface*.

However, the main concern here is the TTP, and the protocols between it and its clients.

6.4 Key management for encryption

In this section we overview the work conducted within ASPeCT on key management for end-to-end encryption. We concentrate on the sensitive issue of providing key escrow (recovery) services through the assistance of a TTP. We present a general framework for assessing key escrow schemes and then discuss in detail one particular proposal, the JMW protocol. We also describe the integration of key escrow within the security architecture of UMTS and consider some alternative escrow options.

6.4.1 Lawful interception in UMTS

The provision of key escrow (recovery) services is a sensitive issue that has attracted recent attention and debate. In any cryptographic system that intends to offer end-to-end confidentiality, there may be a need to permit certain parties under special circumstances to access keys used to encrypt communications. Such parties and circumstances could be law enforcement or security organisations in the event of a criminal investigation, or could be local security managers in the event of loss or damage to previously used cryptographic keys. In either case, the access management of cryptographic keys is conducted by a *key recovery agent* (KRA), which in the UMTS scenario we simply regard as a TTP. This TTP is generally trusted by both users and parties involved in the recovery process.

With respect to key recovery within UMTS, the starting premise within ASPeCT is that it may be the case that legal requirements at both national and international level dictate that key recovery services must be provided within a mobile telecommunications network. The main objective of ASPeCT was thus to demonstrate the provision of key escrow services through the assistance of TTPs, *given that such a service is required*. It was not a central objective of ASPeCT to add to the debate over whether key recovery is a desirable service, although technical results obtained during the establishment of the working demonstrators may contribute to this wider discussion.

6.4.2 Framework for evaluating key escrow schemes

In order to identify properties and permit comparison between different key escrow schemes it is necessary to identify an evaluation framework. Such a framework will necessarily be somewhat abstract and subjective, but in the absence of any framework it is virtually impossible to conduct any worthwhile analysis.

We thus provide two lists that jointly provide a framework within which to analyse proposed escrow protocols. The first list contains *parameters* of an escrow protocol, which describe the relationship between entities in the protocol and particular properties of the protocol. The second list contains *requirements* of the protocol, which are necessary outcomes of the protocol. Note that while the requirements are distinct from the parameters of the protocol, some of the requirements are specified in terms of the protocol parameters. The two lists are partially compiled from lists in [JMW96a] and [VKT97]. The detailed taxonomy in [DB96] and the list of criteria in [D95] are also of interest, although somewhat broad to be of direct use in ASPeCT.

In the following by *users* we mean parties wishing to communicate securely through the use of end-to-end encryption, and by *interception agents* we mean parties wishing, through some authorised means, to access

messages sent between two users. By a *warrant* we mean a legal document giving interception agents authority to request access to certain types of communication.

6.4.2.1 Escrow parameters

The following parameters should be identified when proposing or analysing an escrow protocol.

1. **Communication structure:** *Who talks to whom?* This parameter includes a description of which entities are involved in the protocol, and what type of communication channel (if any) exists between them.
2. **Trust relationships:** *Who trusts who?* This parameter describes the degree of trust that entities involved in the protocol have for one another. When two entities partially trust one another the nature of this partial trust should be precisely described.
3. **Interception safeguards:** *Which communications can be intercepted, when and by whom?* This parameter details the scope of interception permissible with respect to precision of target and time length. It also describes which TTPs can assist in each type of interception.
4. **Escrow type:** *What type of escrow?* This parameter identifies whether session keys are to be escrowed and whether the choice of encryption algorithm is to be fixed, or both.
5. **Cryptographic flexibility:** *How flexible?* This parameter describes how cryptographically flexible the protocol should be. In particular, how much choice should there be with respect to key update policies and choice and use of trusted third parties.
6. **Communication Type:** *What type of communication?* This parameter describes the communication scenario that the key escrow protocol is to be applied to. For example, whether communication is one-way or two-way, for national or international networks.
7. **Implementation:** *What implementation restrictions exist?* This parameter describes any relevant implementation restrictions that exist for the protocol environment. For example whether the solution is for hardware or software (or both), or whether public or secret key algorithms can be supported.

6.4.2.2 Escrow requirements

Any key escrow protocol should satisfy the following requirements.

1. **User completeness:** *Honest users succeed.* By following the protocol, honest users will succeed in establishing a session key for encrypting messages.
2. **Agent completeness:** *Honest agents succeed.* By following the protocol, an interception agent, in possession of an appropriate warrant, will be able to obtain plaintexts of any messages subject to the specifications for such interception detailed by the interception safeguards.
3. **User soundness:** *Dishonest users do not benefit.* Any user activity that is designed to misuse the protocol should at least be detectable. This includes using the framework of the protocol to establish a session key by some other means or encrypting by techniques not specified in the protocol.
4. **Agent soundness:** *Dishonest agents do not benefit.* Any agent activity that is designed to misuse the protocol should at least be detectable. This includes any activity not specified by the interception safeguards such as release of information by a TTP not designated to do so by the safeguards, release of information not specified by the safeguards, and release of information to an interception agent not in possession of an appropriate warrant.
5. **User acceptability:** *User approval.* The protocol should be acceptable to users. Factors that are likely to lead to acceptability include expert approval of the protocol, use of well-known cryptographic techniques, cryptographic flexibility and compatibility, efficiency of use and visible benefits of use.
6. **Agent acceptability:** *Agent approval.* The protocol should be acceptable to interception agents. The acceptability factors largely overlap those of user acceptability, however emphasis is different for some cases. For example *efficiency* in this case refers to efficiency of interception.

7. **Legality:** *Within appropriate laws.* The protocol should satisfy all relevant legal restrictions, including those concerning interception policy and cryptographic algorithm use and export. The protocol should also protect the relevant constitutional rights of all participating entities.

Note that while we claim that the above requirements are necessary for any escrow protocol, it may not always be possible to verify that they all hold. For instance, to verify agent soundness we must ensure that interception agents can not present forged warrants. This lies outside the scope of ASPeCT. Note also that *acceptability* is not well-defined. It may be the case that some participants find the parameters of the protocol unacceptable, before even considering the protocol itself. For a more detailed separation and discussion of issues concerning acceptability and legality see [VKT97]. Note that in assessing protocols we omit discussion of the last requirement, legality, as this lies somewhat beyond the technical scope of ASPeCT.

6.4.2.3 Desirable parameters for UMTS

We will now consider a key escrow scenario regarded as appropriate to a UMTS application and identify a suitable set of parameters. This scenario has been selected bearing in mind the possible problems of requiring access to messages in different UMTS domains (possibly different nation states) in which legal requirements may differ and cross-domain co-operation may be costly or limited.

1. **Communication structure:** Two UMTS users A and B, register with separate home TTPs, denoted TA and TB respectively. User A and TA share a secure link, as do user B and TB. The TTPs TA and TB have access to a secure link but use of this link is restricted as it is regarded as expensive. Users A and B communicate over an insecure link. Interception agents can communicate with either TA or TB.
2. **Trust structure:** Users and interception agents trust both TA and TB. Users and interception agents do not trust one another directly but rather trust the TTPs to act honourably in dealings between them. The TTPs do not need to trust the users or interception agents however they have some partial trust, at least to the extent that if either of these entities regularly abuse TTP services then the TTPs may lose their *trusted* status. This also applies to trust between TA and TB. Users A and B partially trust one another, at least not to subsequently reveal shared session keys.
3. **Interception safeguards:** Interception agents have the potential to access any message sent between A and B (if an appropriate warrant is obtained). Interceptions should however be *targeted* and *time-bounded.* A targeted interception specifies precisely whether *all* messages from (or to) a specific user can be intercepted, or whether only messages from (or to) other specifies users can be intercepted. Time-bounded interceptions cover only a specified time, marked by dates or time-stamps. All messages sent outside the specified interception period should remain fully protected.
4. **Type of escrow:** Only session keys are to be escrowed. Any encryption algorithm can be used.
5. **Cryptographic flexibility:** Users should be able to generate (or request) fresh keys as often as possible. The protocol should permit any symmetric encryption algorithm to be used for the encryption of messages. In this basic scenario users have one fixed home TTP.
6. **Communication type:** One-way or two-way communication.
7. **Implementation:** No restrictions for the basic scenario.

6.4.3 The basic JMW protocol

The protocol implemented in the first ASPeCT TTP demonstrator is based on the *JMW protocol* [JMW96a, JMW96b]. This protocol has received widespread attention and has sometimes been referred to as the *Royal Holloway* protocol. Several variants of this protocol have appeared including those in [UKC96] and [CGM96]. We note also that in [AR97] a variant of the JMW scheme is referred to as the *GCHQ* protocol. All variants of the JMW scheme are based on the Diffie-Hellman key exchange protocol [DH76].

6.4.3.1 Detailed protocol description

The protocol is in the context of a pair of users, where one user wishes to send the other a confidential message and needs to be provided with a session key to protect it. The start situation for the protocol is as follows.

1. Four participants, namely two users (one as a sender, say A, and the other as a receiver, say B) and two TTPs (one for each user, say TA and TB), are involved.
2. Two TTPs have agreed between them values *g* and *p*. These values must have the usual properties required for operation of the Diffie-Hellman key exchange mechanism, namely that *g* must be a primitive element modulo *p*, where *p* is a large prime and *p*-1 can be divided by another large prime *q*. These values have been passed to the two users.
3. Each TTP has an asymmetric signature verification key pair. The private signature key is known only to himself, and an authenticated copy of the public verification key can be accessed by his own user and the other TTP.
4. Each user and his TTP have access to a protected channel between them, which provides origin authentication, data integrity and confidentiality.

Figure 6.3 shows the message exchanges of the protocol among the four participants. The protocol includes four parts, each of which can optionally be run separately.

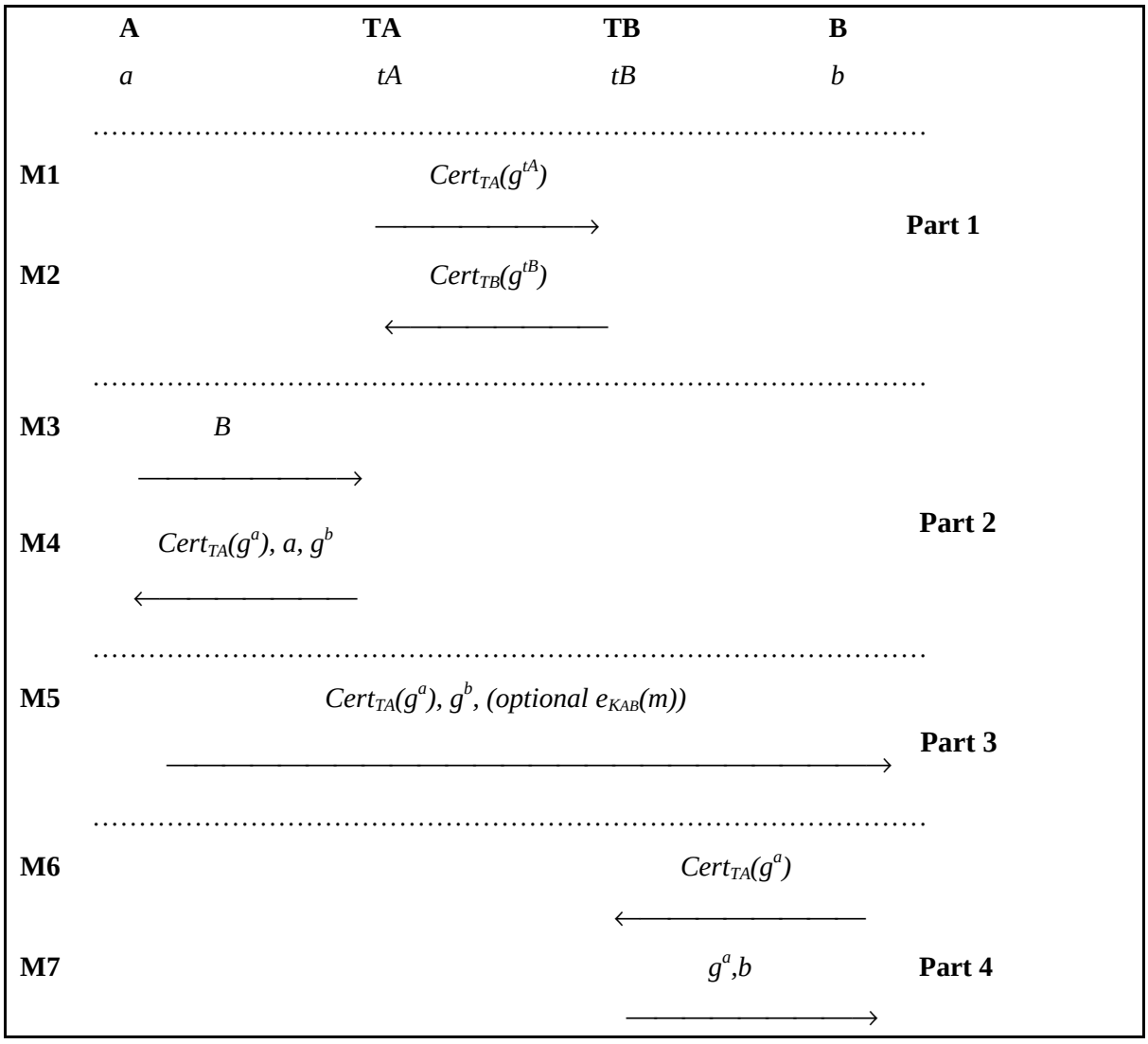


Figure 6.3 - The JMWprotocol as implemented in the Demonstrator

Part 1: Share a secret between two TTPs

1. Each TTP, *TA* or *TB*, generates a private and public key agreement key pair (tA, g^{tA}) or (tB, g^{tB}).
2. Each TTP sends his public key agreement key signed using his private signature key to the other TTP in M1: $Cert_{TA}(g^{tA})$ or in M2: $Cert_{TB}(g^{tB})$.
3. Each TTP verifies the received public key agreement key using an authenticated public signature verification key of one another.
4. Each TTP computes a shared secret, $K_{TATB} = g^{tAtB}$, using his own private key agreement key and the other's public key agreement key in Diffie-Hellman key establishment algorithm.

Part 2: Certificate generation in A's domain

1. *A* sends *TA* a request in M3 including *B*'s name.
2. *TA* generates a random number *a* as *A*'s private send key and a corresponding public send key, g^a.
3. *TA* makes a certificate for *A*'s public send key, $Cert_{TA}(g^a)$.
4. *TA* computes *B*'s private receive key, $b = h(K_{TATB}, B)$, and the corresponding public receive key, g^b.
5. *TA* sends *A*'s public send key certificate, *A*'s private send key and *B*'s public receive key to *A* in M4.
6. *A* computes a shared key, $K_{AB} = g^{ba}$, using his own private send key and *B*'s public receive key.

Part 3: message transmission from A to B

A sends the following information to *B* in M5:

- his public send key certificate issued by *TA*,
- *B*'s public receive key, and
- an optional message encrypted by the shared key, K_{AB}.

Part 4: Certificate generation in B's domain

1. After receiving M5, *B* sends *TB* a request including *A*'s public send key certificate issued by *TA*, and *B*'s public receive key sent by *A*.
2. *TB* computes *B*'s private receive key, $b = h(K_{TATB}, B)$, and verifies g^b.
3. *TB* verifies *A*'s public send key certificate using the public signature verification key of *TA*.
4. *TB* sends *A*'s public send key and *B*'s private receive key to *B*. Note that if *B* has already got his current private receive key, this key does not need to be sent in M7.
5. *B* computes the shared key, $K_{AB} = g^{ab}$, using his own private receive key and *A*'s public send key.

After running the protocol successfully, a shared key K_{AB} will be established between the sender and receiver, and this key will be escrowed by both TTPs.

Note on the above interpretation as implemented for the Demonstrator:

There are actually two cases to be considered in Part 4

(i) where B has had previous dealings with clients of TA (this is the general case):
B has the private receive key *b* (for all clients of TA) and TA's public verification key .

1. B verifies $Cert_{TA}(g^a)$. using TA's public verification key.
2. B computes shared key $K_{AB} = g^{ab}$.

(ii) where B has not had previous dealings with any client of TA:
(this happens only once per TA; it is going to be comparatively rare and should not be seen as typical)

1. B sends TB a request with identity of TA.

2. *TB* computes *B*'s private receive key, $b = h(K_{TATB}, B)$.
3. TB sends *b* and TA's public key to B.
4. B verifies $Cert_{TA}(g^a)$. using TA's public verification key.
5. B computes shared key $K_{AB} = g^{ab}$.

6.4.3.2 Interception options

Once the JMW protocol has been run successfully, it is possible for interception agents to approach either of the TTPs to make a request for access to communication between the users. There a two different possible methods by which this access could be given:

6.4.3.2.1 TTP releases escrowed keys

Three different keys can be released by TTPs involved in a particular communication:

- TA could release the private send key *a* of the sender, which would allow all messages sent from the sender to be decrypted during the lifetime of *a*.
- Either TA or TB could release the private receive key *b* of the receiver, which would allow all messages sent to the receiver from all users of the sender's TTP to be decrypted during the lifetime of *b*.
- Either TA or TB could release the session key *K*, which would allow all messages from the sender to the receiver to be decrypted during the joint lifetime of *a* and *b*.

The release of private send or receive keys allows some efficient interceptions to be made. However, care must be taken to ensure that these keys do not allow the agent to access communications that are not covered by the warrant. For some types of warrant it may not be permissible to release the private send key or private receive key. Instead, the TTP must release the session key for each message covered by the warrant.

In order to investigate the acceptability of the scheme to both users and agents, we consider some of the possible types of warrant that may be presented. First we note that the scheme supports both node-based interception (in which all communications involving a particular target can be decrypted) and edge-based interception (in which only communications between two targets can be decrypted). Both types of interceptions are now discussed in turn. For each type of interception we explain how warranted interception can be provided. We consider four possible types of node-based interception:

N1. A TTP is warranted to provide access to all outgoing communications from a user for which it acts.

The TTP can provide the private send key(s) for the targeted user.

N2. A TTP is warranted to provide access to all incoming communications to a user for which it acts.

The TTP can provide the private receive key(s) for the targeted user.

N3. A TTP is warranted to provide access to all incoming communications (from users for which it acts) to a user for which it does **not** act.

The TTP can provide the private receive key(s) for the targeted user.

N4. A TTP is warranted to provide access to all outgoing communications (to users for which it acts) from a user for which it does **not** act.

The TTP can provide the session key(s) for outgoing communications (to users for which it acts) from the targeted user.

Interceptions N1, N2, and N3 can be provided with good agent acceptability, since the release of private send and receive keys increases the efficiency of interceptions. Interception N4 has a lower agent acceptability, since individual session keys must be obtained. However, although interception N4 is less efficient, it is likely to be much less common in practice than N1 and N2 (and possibly N3). Note also that in each case, user acceptability and agent soundness are maintained by ensuring that the agent can only access those communications covered by the warrant. We consider two possible types of edge-based interception:

E1. A TTP is warranted to provide access to communications from a particular user for which it does act to a particular user for which it does **not** act.

The TTP can provide the session key(s) for communications between the two users.

E2. A TTP is warranted to provide access to communications from a particular user for which is does **not** act to a particular user for which it does act.

The TTP can provide the session key(s) for communications between the two users.

For edge-based interception, only individual session keys can be released. These can be obtained from either TTP and only provide access to communications between the two targeted users. Thus, user acceptability and agent soundness is maintained.

6.4.3.2.2 TTP performs decryption

If the TTP performs decryption then it is easy to ensure that interceptions are targeted, since the actual communications to be decrypted are presented to the TTP. Thus, agent soundness and user acceptability are maintained, and many of the problems associated with releasing keys are avoided. The main disadvantage of this approach is the increased amount of communication required between the TTP and the intercepting agent, and the potential delay in accessing communications. However, this disadvantage could be partially overcome if the key established in the protocol was used as a key encrypting key used to encrypt the actual session key. In this case the agent would present encrypted session keys to the TTP for decryption, rather than whole messages.

6.4.3.3 Evaluation of protocol against parameters and requirements

We now check if the protocol matches the parameters of our basic UMTS scenario. A quick check reveals that it *almost* does. The only parameter that is not agreed upon is the interception safeguards. As can be seen from the types of interception available, while precise targeted interception is possible by the third of the three interception types, there is no provision for time-bounded interception (except between private send and receive key updates). We discuss variants that permit time-bounded interceptions in Section 6.4.4.1.

We now check if the protocol matches our list of requirements.

1. **User completeness:** Yes.
2. **Agent completeness:** Yes.
3. **User soundness:** Several problems exist that relate to the possibility that users use the established public keys in a different public key encryption system to the one proposed, or use the established session key to generate an alternative session key not available to the TTPs. These problems are however general problems that apply to many different key escrow schemes. For further discussion, see [KP96].
4. **Agent soundness:** No. Although this is a difficult requirement to ensure there should at least be provision for time-bounded interceptions before this can be satisfied.
5. **User acceptability:** No. Certainly not acceptable without clarification of agent soundness. The protocol does use established techniques however and does allow users to request fresh private send keys at any time. Private receive key generation is a more complex operation as updating of keys requires communication between TA and TB.
6. **Agent acceptability:** Needs clarification of the user soundness. Probably more acceptable to agents than users. Some efficient interceptions are possible through the release of private send or receive keys, however these types of interception should only be permissible when an interception is targeted in such a way that all affected users are covered by the warrant.

6.4.4 Protocol variants

In the light of the remarks on the suitability of the JMW protocol we consider a number of variants and extensions which could improve the acceptability of the protocol in a UMTS environment.

6.4.4.1 Time-bounded interceptions

In order to increase the agent soundness and user acceptability of the protocol it is worth incorporating the option of time-bounded interceptions. We consider a number of proposals.

6.4.4.1.1 Variant I

This variant was proposed in [JMW96b]. The protocol is identical to the JMW protocol except that the private receive key of B is regularly updated. Hence the key b of the JMW protocol becomes a *permanent* receive key, and a second key generating function G is applied to b and a date-stamp d to compute a *temporary* receive key $b' = G(b,d)$. Time-bounded interceptions are now possible by the following types of interception:

- TA or TB releases b'. All messages from users with home TTP TA to B can be read for the validity of date-stamp d.
- TA or TB releases $g^{ab'}$. All messages from A to B can be read for the joint lifetime of a and validity of date-stamp d.

6.4.4.1.2 Variant II

This variant is based on ideas in [UKC96]. This protocol is very similar to JMW variant II except that the private send key of A is also regularly updated. Thus the key a of the JMW protocol becomes a *permanent* send key, and a key generating function G is applied to a and a date-stamp d to compute a *temporary* send key $a' = G(a,d)$. For simplicity we assume that the date-stamp is updated with the same frequency for both temporary send and temporary receive keys, although this need not be the case. Time-bounded interceptions are now possible by the following types of interception:

- TA releases a'. All messages from A can be read for the validity of date-stamp d.
- TA or TB releases b'. All messages from users with home TTP TA to B can be read for the validity of date-stamp d.
- TA or TB releases $g^{a'b'}$. All messages from A to B can be read for the validity of date-stamp d.

6.4.4.1.3 Variant III

The JMW protocol is used to establish a session key g^{ab}. User A takes a hash function H and a date-stamp d, and computes a *session key of the day* $H(g^{ab},d)$ that is employed as the session key for encrypting messages from A to B. Time-bounded interceptions are now possible by the following type of interception:

- TA or TB releases $H(g^{ab},d)$. All messages from A to B can be read for the joint lifetime of keys a, b and validity of date-stamp d.

Note that it is most likely that the lifetimes of keys a and b span several date-stamps and so a time-bounded interception over an extended time period is likely to require the release of $H(g^{ab},d)$ for all date-stamps d corresponding to the time covered by the court order.

6.4.4.1.4 Variant IV

Proceed as in JMW variant II except replace date-stamp d with date-stamp D, where D lasts for a greater time period than d. User A then proceeds to establish the session key as in JMW variant II but then incorporates the ideas of JMW variant III. Thus user A takes a hash function H and date-stamp d and computes a session key of the day $H(g^{a'b'},d)$, which is then employed as the session key for encrypting the message from A to B. Time-bounded interceptions are now possible by the following types of interception:

- TA releases a'. All messages from A can be read for the validity of date-stamp D.

- TA or TB releases b'. All messages from users with home TTP TA to B can be read for the validity of date-stamp D.
- TA or TB releases $H(g^{a'b'}, d)$. All messages from A to B can be read for the joint validity of date-stamps d and D.

Thus in JMW variant IV time-bounded interceptions for short time intervals and precise targets can be conducted by release of session keys by either TTP. For longer interceptions that apply to wider targets it is possible to release the private temporary send or receive keys that are date-stamped by all D covered by the appropriate court order. The exact granularity of the two different date-stamps can be adjusted to fit the application.

6.4.4.2 Two-way communication

Although we have discussed the basic protocol (and variants) in terms of one-way communication, there is no reason why session keys established by the protocol cannot be used for two-way communication. If it is preferred that both users should contribute to the establishment of a two-way session key then the one-way protocol can be performed twice, once in each direction, and the two resulting session keys could be combined. Note that if a session key has been agreed upon for two-way communication between user A and user B then, when a warrant is issued for interception of communication from A to B, it is unavoidable that all communication from B to A will also be intercepted. If this situation is not acceptable then such two-way keys should not be established and communication between A and B should take the form of two separate streams of one-way communication (one from A to B, and the other from B to A) with a different one-way session key protecting the information flow in each direction.

In [D14] various merits of several variants of the basic one-way communication protocol extended to two-way communication are considered. These are based on several simplified protocols that had previously appeared in ASPeCT related documents. All these protocols are Diffie-Hellman [DH76] based protocols for establishing a two-way session key. The following general conclusions resulted from this study.

- *Combining two one-way keys does not seem profitable.* There does not seem to be any significant advantages in combining the two one-way keys as suggested as an option in [JMW96a] as this appears to result in a protocol that does not offer many obvious gains over a simple Diffie-Hellman key exchange [DH76] with escrowed secret keys..
- *It is possible to use "one-way" keys for "two-way" communication.* User security does not seem to be affected by the adoption of a one-way key for two-way communication. This allows the benefits of efficient warranted interception in some one-way protocols to be extended to two-way protocols. The basic one-way escrow scheme of [JMW96b] is the most efficient in this regard.
- *Directional targeting can not be done with a two-way key.* This somewhat obvious comment is made to highlight the fact that if sent and received communication is to be separated then a protocol that establishes a two-way key should not be used. Rather, a one-way key should be established in each direction and these keys should not be combined, but used separately, one for communication in each direction (the basic two-way version of [JMW96b]).

6.4.4.3 Escrow in multiple domains

The basic UMTS scenario considered assumes that each user belongs to a domain and only directly communicates with a *home TTP*, which is a TTP associated with the user's domain. Interception agents in either the sender's or the receiver's home domain can then access the communications by approaching the appropriate TTP under their jurisdiction.

However, in a multiple domain environment it may be required that interception agents in other domains, other than the home domain of the sender and receiver, have access to the communications. Consider for example a potential scenario in UMTS where two UMTS users A and B, who communicate with each other using end-to-end encryption, are citizens of countries C and D, respectively, work in countries E and F, are registered with two networks in countries G and H, and are roaming in two countries I and J. Their traffic

might conceivably need to be intercepted by agents in any of the countries (domains) C-J. In this scenario the secret confidentiality key may need to be escrowed to TTPs in all the countries C-J.

Key escrow in multiple domains is easily catered for in key escrow schemes that are not combined with key distribution; keys are simply escrowed to all the required TTPs. In the JMW scheme, where key escrow is combined with key distribution, multiple domains can be catered for using a conference key distribution scheme, which allows the TTPs in the required domains to independently gain access to an escrowed key and make an updated contribution to the key without communication with TTPs in any other domain [CM96]. The use of key escrow in multiple domains may be used to increase agent acceptability in some scenarios.

6.4.4.4 Split escrow

In the demonstration a user can only register with one TTP. However, in practice a user may want the extra reassurance offered by having their keys shared between a number of independent keys. This idea is sometimes called split escrow. In order to support split escrow, a user *X* must be able to register with a set of TTPs $\{TX_i\}$, which the user is prepared to trust collectively, but perhaps not individually. Two suggested methods for enhancing the JMW protocol in this way were suggested in [JMW96b] and [CGM96]. These protocols were generalised and made more efficient in [M97].

6.4.4.5 Increased cryptographic flexibility

It is assumed that *g* and *p* are system wide parameters. However, in practice different integers *g* and *p* could be agreed between each pair of TTPs in the network. If this approach was adopted then each user must receive the value of *p* before it can carry out the required cryptographic operations. This could be achieved by ensuring that the appropriate TTP sends the value of *p* to the user as part of the protocol.

6.4.5 Alternative schemes

At the commencement of the ASPeCT project, key escrow was an extremely active a research area with many protocols and schemes being proposed. For an introductory overview, see [MH98]. It is not practical to compare the protocol proposed within ASPeCT with every other alternative proposal, and so we restrict attention here to three carefully selected escrow proposals. The first protocol is chosen because it also offers a range of possible interception options, in a similar way to the JMW scheme. The second protocol represents a major proposal from a well-known commercial computing company. The third proposal represents an entirely different approach to providing key escrow. We present brief description of the alternative proposals here and then make a comparison between them.

6.4.5.1 LWY protocol

The following protocol is referred to as the *LWY protocol* and was proposed in [LWY95]. Let p and q be large primes, such that q divides $p-1$, and let g be an element of order q (modulo p). Let $S(x), P(x)$ be the secret and public keys of X, where $P(x) = g^{S(x)}$ (modulo p). Let E be a secure block cipher and H be a one-way hash function. Values p, q, g, E and H are all public.

In advance of communications

1. User A selects a TTP in each domain that requires access to the future communication. Let A select TTP TA, and let B select TTP TB.
2. User A generates $S(a), P(a)$, while user B generates $S(b), P(b)$.
3. $S(a)$ is escrowed to TA, while $S(b)$ is escrowed to TB.

Note that in step 2 we assume that the users generate their own secret and public keys. However, in [LWY95] it is not specified who generates these keys.

At time of communication

1. A computes the following:

$K(a,b,d) = H(P(b)^{S(a)},d)$; $S(a,d) = H(S(a),d)$; $S(a,b,d) = H(S(a,d),P(b))$

2. B computes the following:

$K(b,a,d) = H(P(a)^{S(b)},d)$; $S(b,d) = H(S(b),d)$; $S(b,a,d) = H(S(b,d),P(a))$

3. A sends $E_{S(a,b,d)}(K(a,b,d))$ to B.

4. B sends $E_{S(b,a,d)}(K(b,a,d))$ to A.

Users A and B can now exchange messages using session key $K(a,b,d) = K(b,a,d)$.

Key recovery

Several types of interception are possible using the LWY protocol:

- TA releases $S(a)$. All messages from or to A can be read for the lifetime of $S(a)$.
- TA releases $S(a,d)$. All messages from or to A can be read for the joint lifetime of $S(a)$ and date-stamp d.
- TA releases $S(a,b,d)$. All messages between A and B can be intercepted for the joint lifetime of $S(a), S(b)$ and date-stamp d.
- TB releases $S(b)$. All messages from or to B can be read for the lifetime of $S(b)$.
- TB releases $S(b,d)$. All messages from or to B can be read for the joint lifetime of $S(b)$ and date-stamp d.
- TB releases $S(b,a,d)$. All messages between A and B can be intercepted for the joint lifetime of $S(a), S(b)$ and date-stamp d.

6.4.5.2 IBM protocol (SecureWay)

The infrastructure of SecureWay [IBM97] relies on the existence of a number of *key recovery service providers (KRSPs)* in each domain. In the documentation an amount of stress is placed on suggesting that these service providers are not TTPs. We note however that the KRSPs do play a very similar role to TTPs in other protocols. In each case TTPs (KRSPs) hold some secret information that is on its own necessary but not sufficient to determine the session key. Assume that user A wishes to send a message to user B. We briefly describe how to send a message using SecureWay.

In advance of communication

1. User A selects a subset of KRSPs in each domain that requires access to the future communication. For simplicity let A select two KRSPs in each of the sending and receiving domains; label these SPA1, SPA2 and SPB1, SPB2.
2. For each designated KRSP, users A, B and the KRSP mutually agree on a "random" number. This random number (one for each designated KRSP in each domain) is fixed over a number of different communication sessions.

Note that it is suggested that Step 2 could be achieved by means of a "three-way" Diffie-Hellman key exchange. Another option is for A and B to exchange a random seed that is then used to pseudorandomly derive the random numbers, and for them to pass them on to the KRSPs using a public key encryption method.

At time of communication

1. User A selects a symmetric encryption algorithm E.
2. User A selects a session key K.

3. For each random number $r_{A1}, r_{A2}, r_{B1}, r_{B2}$, user A computes a secondary parameter (by some publicly known procedure that takes as input the random number and, possibly, date, address details, encryption method etc.). Denote the results $k_{A1}, k_{A2}, k_{B1}, k_{B2}$.

4. A sends to B: $E_{k_{A2}}(E_{k_{A1}}(K)),\ E_{k_{B2}}(E_{k_{B1}}(K)),\ E_K(m),$ where m is the message.

On receiving

1. User B decrypts the message. To do this B must already have a copy of session key K. It is not precisely specified how B computes this key, but one option is by a key agreement exchange with A prior to communication.

2. User B checks the key recovery information (B can do this as B knows each of the random number seeds from which the secondary parameters are computed).

Key recovery

The process of key recovery is identical in each domain. For example, for the sending domain:

1. Interception authority approaches SPA1 and SPA2 with an interception warrant and details of the communication to be intercepted.

2. SPA1 computes and releases k_{A1}, SPA2 computes and releases.

3. Interception authority decrypts (in reverse order) the nested encryption of K.

4. Interception authority decrypts the message.

6.4.5.3 VKT protocol (Binding cryptography)

The concept of *binding cryptography* was discussed in [VKT97] and [VT97]. Binding cryptography can be thought of as a type of escrow protocol and it has some attractive properties that make it worth considering for ASPeCT.

Let p be a large prime and G be the multiplicative group $\mathbf{Z}_p^*$. Let g be a generator of $\mathbf{Z}_p^*$. Let $S(x), P(x)$ be the secret and public keys of user or TTP X, where $P(x) = g^{S(x)}$ (modulo p). The parameters p and g are public. The protocol is based on the ElGamal public key encryption system [ElG85].

In advance of communications

1. User A selects a TTP in each domain that requires access to the future communication. Let A select TTP TA, and let B select TTP TB.

2. User agrees ElGamal parameters with its TTP.

At time of communications

1. User A generates a random r

2. User A sends the following to user B:

3. $(g^r, P(b)^r K)$; $(g^r, P(ta)^r K)$; $(g^r, P(tb)^r K)$; *binding data*

Values 1, 2 and 3 represent ElGamal encryptions of K using the public keys of B, TA and TB respectively. The *binding data* is the text of a zero-knowledge proof that enables anyone to test that the three encryptions have all been performed using the same parameter r (and thus represent valid ElGamal encryptions for the three entities). Details of the construction of the binding data can be found in [VT97].

On receiving

1. User B decrypts the session key using his secret ElGamal key

2. User B checks the key recovery information (B can do this as B knows the public ElGamal keys for the TTPs TA and TB)

Key recovery

Only one type of interception is possible:

- TA or TB releases K. All messages from A to B can be intercepted for the lifetime of K.

6.4.5.4 Parameters of alternative protocols

6.4.5.4.1 Communications structure

Protocol	Description of parameter
LWY	An off-line link is required between A and TA and between B and TB in order to escrow secret key with TTP. In addition, an off-line link is required between TA and TB such that all entities can obtain (certified copies of) the public keys of A and B. During interception the agents needs to obtain relevant information from the TTPs in their domain in order to intercept the communications. Co-operation or agreements between TTPs are not required (although TTPs need to be able to obtain public keys of users belonging to other TTPs).
IBM	An off-line link is required between A and its KRSPs and between B and its KRSPs in order to agree on a random number, which is fixed over a number of sessions. During interception the agents must operate very closely with the relevant KRSPs to obtain the information necessary to decrypt each session key. Co-operation or agreements between TTPs are not required.
VKT	An off-line link is required between A and TA and between B and TB in order to let users obtain the relevant (certified) public ElGamal keys. During interception the agents needs to obtain relevant information from the TTPs in their domain in order to intercept the communications. Co-operation or agreements between TTPs are not required.

6.4.5.4.2 Trust relationships

Protocol	Description of parameter
LWY	The users and TTPs in this scheme have the same trust relationships as the users and TTPs in the JMW scheme.
IBM	The users and KRSPs in this scheme have similar trust relationships as the users and TTPs in the JMW scheme. The difference is that the basic IBM scheme allows for key splitting. Thus, the user need not trust its KRSPs individually, but must however trust them collectively.
VKT	The users and TTPs in this scheme have the same trust relationships as the users and TTPs in the JMW scheme.

6.4.5.4.3 Interception safeguards

Protocol	Description of parameter
LWY	TTPs hold information which can be used to generate the session key.

	Time-boundedness is incorporated into the basic scheme. Many types of interceptions are possible. These include efficient mechanisms for performing node-based interception.
IBM	KRSPs hold information which on its own is not sufficient to generate the session key. Instead, the agent needs to recover information from the messages sent from/to the targeted user(s). The scheme naturally incorporates key splitting, thus the agent may have to approach more than one KRSP before being able to intercept the communication. Time-bounded interceptions are possible, but at the expense of a greater complexity of interception. Only one type of interception possible - KRSPs release information which can be used to help compute the session key. Thus, efficient node-based interceptions are not possible. The receiver can check the correctness of the key recovery information. However, this process is more complex than VKT.
VKT	TTPs hold information which on its own is not sufficient to generate the session key. Instead, the agent needs to recover information from the messages sent from/to the targeted user(s). Key splitting is not featured in the basic scheme. Time-bounded interceptions are possible, but at the expense of a greater complexity of interception. Only one type of interception possible - TTPs release information which can be used to help compute the session key. Thus, efficient node-based interceptions are not possible. The receiver can check the correctness of key recovery information. Before an interception, a third party can verify that key recovery information has been correctly computed.

6.4.5.4.4 Escrow type

Protocol	**Description of parameter**
LWY	Session key only
IBM	Session key only
VKT	Session key only

6.4.5.4.5 Cryptographic flexibility

Protocol	**Description of parameter**
LWY	Session key is jointly generated by A, B, TA and TB.
IBM	Session key generated by user. Flexible with regard to key recovery demands of different domains.

	Symmetric algorithm used to compute key recovery information can be different in each domain.
VKT	Session key generated by user. Flexible with regard to key recovery demands of different domains.

6.4.5.4.6 Communications type

Protocol	Description of parameter
LWY	Two-way communication (thus no directional targeting).
IBM	Two-way communication (thus no directional targeting).
VKT	Two-way communication (thus no directional targeting).

6.4.5.4.7 Implementation

Protocol	Description of parameter
LWY	No restrictions.
IBM	No restrictions.
VKT	ElGamal public key encryption used to compute key recovery information.

6.4.5.5 Evaluation of protocols against escrow requirements

We briefly collectively summarise the degree to which these three protocols meet the requirements suggested for a UMTS scenario. Clearly all three protocols offer both user and agent completeness.

1. **User soundness:** The general problems of user soundness mentioned in Section 6.4.3.3 apply in all three cases. However for binding cryptography, a third party can detect that key recovery information has been correctly computed without actually performing key recovery. This may discourage abuse of the scheme.
2. **Agent soundness:** None of the protocols guarantee agent soundness, although this is a difficult requirement to meet.
3. **User acceptability:** This is not high for the LWY protocol. Users do not have good flexibility over key management (changing their secret and public keys is difficult). There is no built in mechanism for split escrow. A possible advantage is that communication between TTPs is not required. User acceptability is higher for SecureWay. Users have good flexibility over key management (they can change their session key), and cryptographic algorithms used. Split escrow can be catered for. The proposal is flexible with regard to key recovery demands of different domains. However, the initialisation stage is relatively complex. User acceptability is also high for binding cryptography. Users can change session key at any time. Split escrow can be catered for and key recovery demands of different domains are flexible.
4. **Agent acceptability:** For all three protocols this needs clarification of the user soundness. In the LWY protocol many types of interception are possible, including efficient means of performing node-based interceptions. TTPs can also generate information required for interception in advance, therefore interceptions do not necessarily require computations to be performed by the TTP. In SecureWay node based interceptions are inefficient. Agents may need to co-operate with many KRSPs and a reasonable degree of computation is required by a KRSP in order to recover a session key. For binding cryptography node-based interceptions are inefficient. There is a degree of computation required by a TTP in order to recover a session key. However, some user abuse of the scheme can be detected without having to perform interception.

6.4.5.6 Comparative evaluation of UMTS escrow protocols

As should have become clear from the analysis of the JMW and alternative protocols, it is very difficult to technically design an escrow protocol that fully fits the suggested UMTS escrow parameters and requirements. It is felt nonetheless that the JMW protocol, and in particular several of the variants suggested, fits these requirements a little closer than many of the alternative options. There are however many grave reservations about the technical aspects of actually implementing such a protocol in any open environment such as UMTS. Further details can be found in, for example [AAB97] and [KM97].

6.5 Demonstration of key management for encryption

In this section we give an overview of the ASPeCT key management for encryption demonstration which implements a TTP-based key escrow protocol known as the JMW protocol [JMW96a, JMW96b].

6.5.1 Overview of demonstration architecture

The entities involved in this TTP architecture are: TTP servers, two types of clients, namely mobile users and Interception Authorities (IAs), and three types of administrators, namely system administrators, security managers and auditors.

The logical and physical connections between the above entities in the demonstration are shown in Figure 6.4, below.

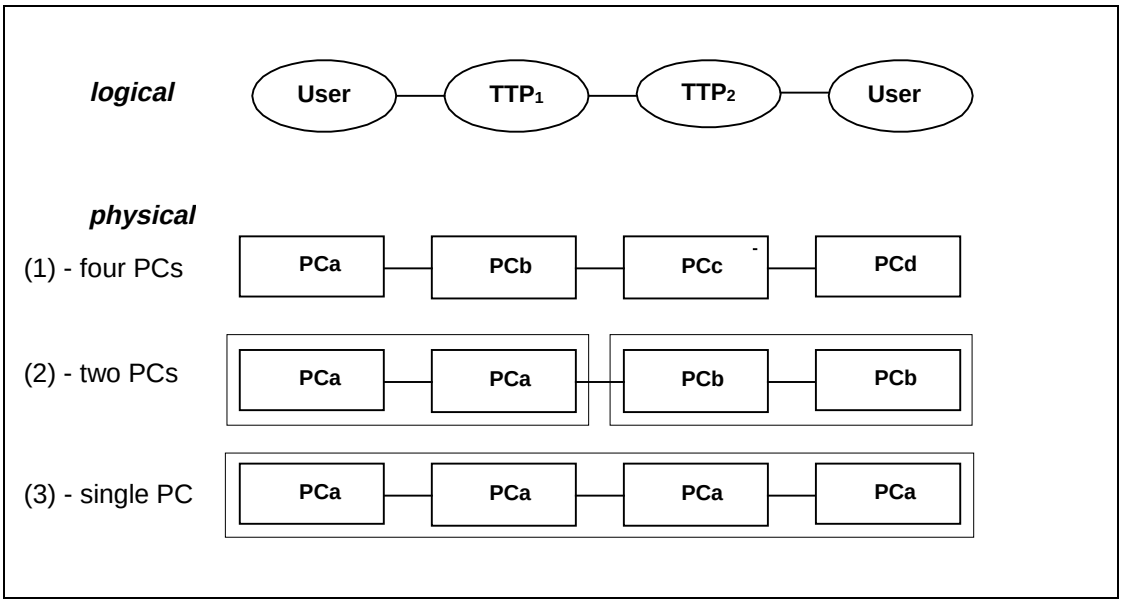


Figure 6.4 - The logical and physical connections between the entities involved in the demonstration

In the demonstrator, each of the main logical entities, expect the IAs, are implemented using PCs. The interconnections between PCs are implemented by using external communication interfaces as shown in Figure 6.5.

The following abbreviations are used in Figure 6.5:

- **SM** denotes a security manager,
- **UIM** denotes a user identity module,
- **GUI** denotes a graphical user interface, and
- **Comm. subs**. is a communication subsystem.

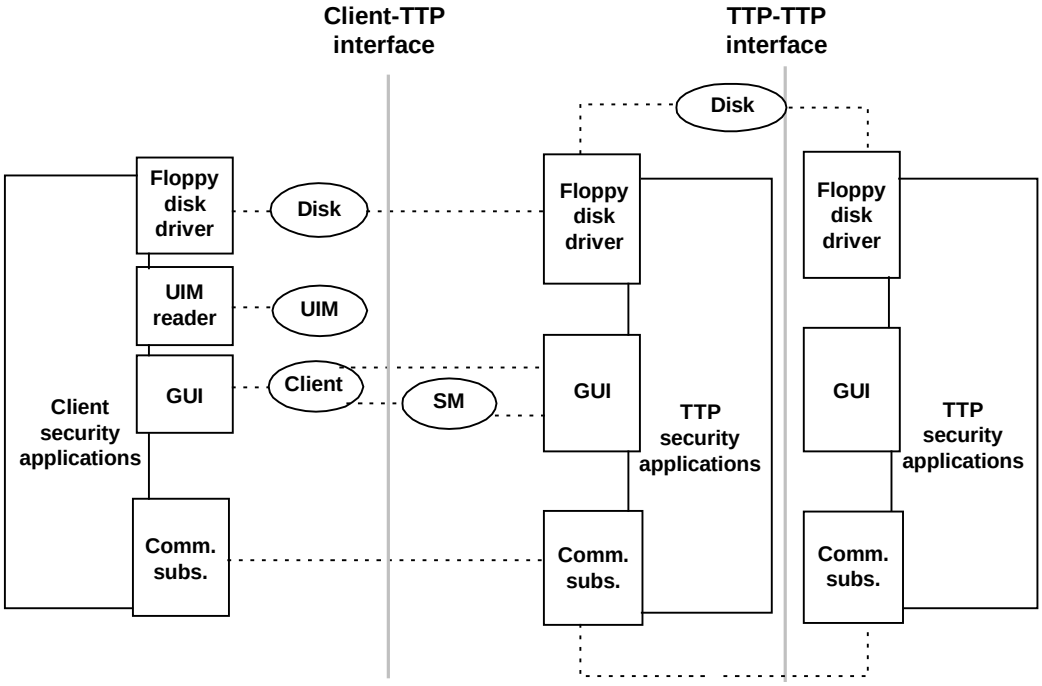


Figure 6.5 - External communication interfaces between the entities

In the TTP server's PC, there are three interfaces for external communications:

- a floppy disk driver,
- a GUI, and
- a *communication subsystem*, which is used to implement the communications application running over TCP/IP.

In the client's PC, there are four external interfaces:

- a floppy disk driver,
- a UIM reader,
- a GUI, and
- a communication subsystem.

A client can work with its PC in the following three ways:

- by using a floppy disk,
- by using a UIM (for a mobile user), and
- by interacting with a GUI of the PC.

The client can communicate with a TTP server in the following four ways:

- by using a floppy disk,
- by interacting with a GUI of the TTP,
- via a SM who can talk with the GUI of the TTP, and
- via a network interface between the TTP's PC and the client's PC (e.g. using an Ethernet connection and the TCP/IP protocol).

The communications between TTP servers implemented on different PCs will either be in the form of off-line transfers (e.g. using floppy disks) or via a network interface (e.g. an Ethernet connection and the TCP/IP protocol).

6.5.2 Demonstration configuration

The number of PCs used in the TTP Demonstrator could be chosen during setup. There are three main options:

- using only one PC for the four participants,
- using two PCs, one for A and TA, and the other for B and TB,
- using four PCs, one for each participant.

The main objective of the demonstration was to present the procedure necessary for communication, supporting end-to-end encryption, through the use of TTP services. It comprises two clients, User A and User B, located in different domains and two TTPs, TTP A and TTP B, one for each domain. User A communicates securely with User B with the intervention of their respective TTPs which collaboratively perform the role of providing the users with key management services for confidentiality.

The GUI provides drop-down menus and prompt boxes which guide the user through the demonstration.

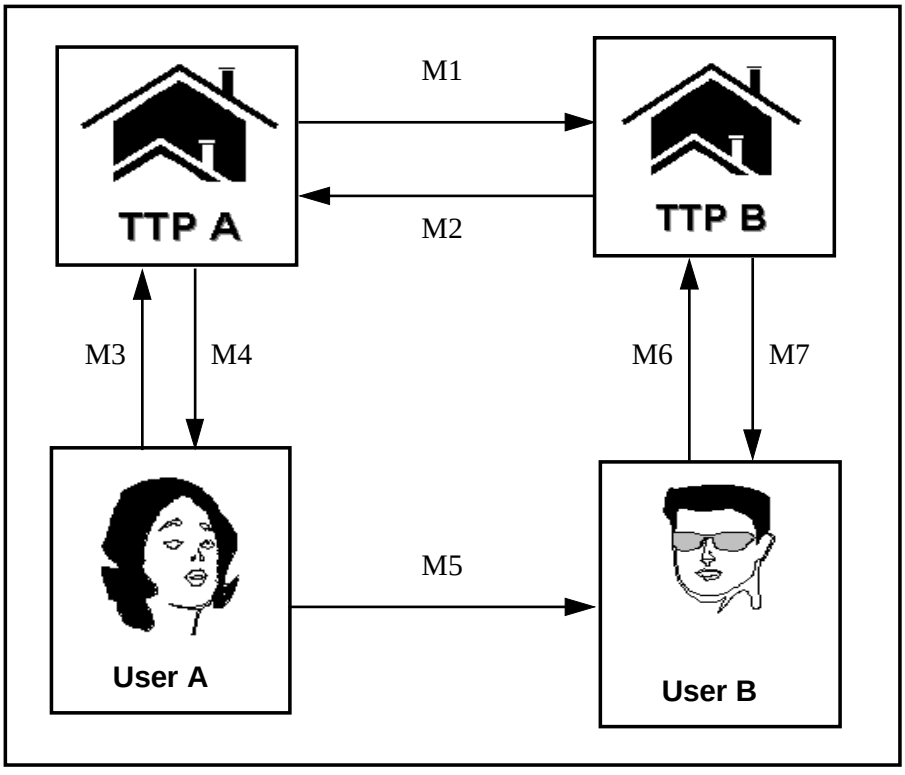


Figure 6.6 – Message exchanges in demonstrator

The procedure is initiated and conducted by the demonstration users who can act as the demonstrated entities. Every entity is entitled to a set of specific actions that the user may select from the appropriate Action Menu. For example, the demonstration user that acts as User A, will be prompted to write the encrypted message he wants to transmit to User B. The whole process is viewed through two options: the Tracer, that displays the details of the actual messages exchanged, and the Monitor, that provides a high level visual representation of the procedure.

The end-to-end confidential communication establishment is done via a message exchange process according to the protocol described in section 6.4.3, above. The message exchange of the protocol, initiated from the Action menu, includes four parts:

- The **initial TTP / TTP set-up** in which the two Trusted Third Parties exchange secret keys in a reliable and secure way to support end-to-end encryption for their respective clients.
- The **initial TTP / sender set-up** in which the sender requests and obtains a private key from the corresponding TTP.
- The **direct sender / receiver communication** in which the encrypted message transmission takes place.
- The **TTP / receiver set-up** in which the receiver requests and obtains the required secret and public key information in order to compute the sender / receiver shared key.

Through the GUI the user may conduct and observe the message exchange process either in a detailed or in a more general way. The sequences of events occurring at each of the four involved entities are presented in the following diagrams. The notation is based on the protocol description from section 6.4.3.

A1.	send M3: B
A2.	wait for M4: $Cert_{TA}(g^a),a,g^b$ – received
A3.	verify $Cert_{TA}(g^a)$ – pass or reject
A4.	compute K_{AB}
A5.	(optional) input a message – m (which will be enter by PC's keypad and display in the window)
A6.	(optional) encrypt m to get $e_{KAB}(m)$
A7.	form M5: $Cert_{TA}(g^a),g^b$, *(optional* $e_{KAB}(m)$*)*
A8.	send M5

Figure 6.7 - Sender A - sequence of events

TA1.	choose *tA*
TA2.	check *tA*
TA3.	compute g^{tA}
TA4.	sign g^{tA}
TA5.	form M1: $Cert_{TA}(g^{tA})$
TA6.	send M1
TA7.	wait for M2: $Cert_{TB}(g^{tB})$ – received
TA8.	verify $Cert_{TB}(g^{tB})$ – pass or reject
TA9.	compute K_{TATB}
TA10.	wait for M3: *B* – received
TA11.	choose *a*
TA12.	check *a*
TA13.	compute g^{a}
TA14.	sign g^{a}
TA15.	form $Cert_{TA}(g^{a})$
TA16.	compute *b*
TA17.	compute g^{b}
TA18.	form M4:$Cert_{TA}(g^{a})$, *a*, g^{b}
TA19.	send M4
TA20.	store *a* and *b*

Figure 6.8 - TTP TA - sequence of events

TB1.	choose *tB*
TB2.	check *tB*
TB3.	compute g^{tB}
TB4.	sign g^{tB}
TB5.	form $Cert_{TB}(g^{tB})$
TB6.	wait for M1: $Cert_{TA}(g^{tA})$ –received
TB7.	verify $Cert_{TA}(g^{tA})$ – pass or reject
TB8.	sending M2:$Cert_{TB}(g^{tB})$
TB9.	compute K_{TATB}
TB10.	wait for M6:$Cert_{TA}(g^{a})$, g^{b} – received
TB11.	compute *b*
TB12.	verify g^{b}
TB13.	verify $Cert_{TA}(g^{a})$ – pass or reject
TB14.	form M7: g^{a}, *b*
TB15.	send M7
TB16.	store *b* and g^{a}

Figure 6.9 - TTP TB - sequence of events

B1.	wait for M5: $Cert_{TA}(g^{a})$, g^{b}, $e_{KAB}(m)$ – received
B2.	send M6: $Cert_{TA}(g^{a})$, g^{b}
B3.	wait for M7: $Cert_{TB}(g^{a})$, *b* – received
B4.	compute K_{AB}
B5.	(optional) decrypt $e_{KAB}(m)$ (*m* will be displayed in the window)

Figure 6.10 - Receiver B - sequence of events

6.6 Evaluation of demonstration

The demonstrator was fully described in [D09] and publicly exhibited at the IS&N conference on the 28th May 1997 in Como, Italy. The demonstrator explicitly shows the use of TTPs in establishing confidentiality keys between users to support an end-to-end encryption service. The demonstrator also implicitly provides a key recovery service through the same TTPs. A full evaluation of the demonstrator is contained in [D14].

6.6.1 Technical feasibility

In order to help evaluate the technical feasibility of the solution, the performance of the demonstrator was evaluated by measuring the processing delays involved during execution of the protocol. The protocol was divided into four parts and the processing delays involved for each part were treated separately. Where timings were appropriate, estimates of the processing delays were made by measuring the delays between fixed measuring points. These measuring points are indicated in Figure 6.11 and Figure 6.12. Each timing was measured five times and an average measurement was computed and taken as a basic estimate of the processing delay. The results are shown in Tables 1 to 6.

All timings were made using an INTEL 486 33,4 Mhz.

6.6.1.1 Estimates of processing delays

Part 1: Share a secret between two TTPs

Part 1 can be conducted off-line, and thus no secure communication timings were taken for this part of the demonstrator operation.

Part 2 : Certificate generation in A's domain

The measurement points for Part 2 are indicated in Figure 6.11.

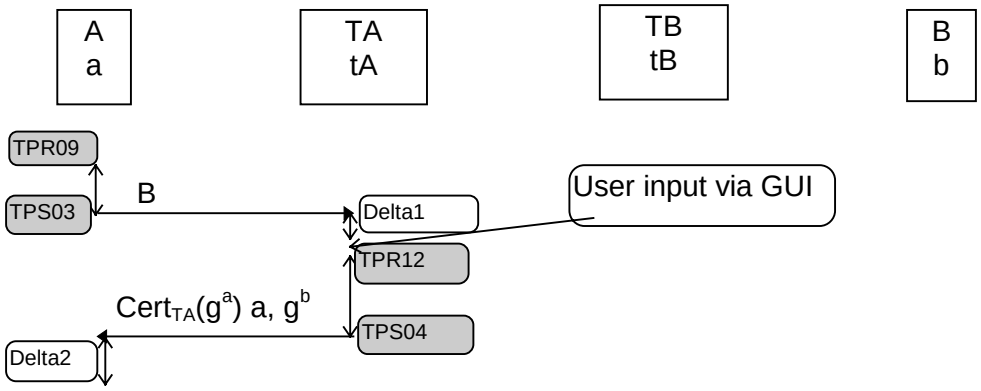


Figure 6.11 - Measurement points for Part 2

Two measurements are provided for Part 2 of the protocol. Firstly user A prepares a request to send to TA (TPR09 - TPS03) and then TA generates a certificate and key values for A. Due to a minor technical problem it was not possible to obtain estimates of the full processing time of the certificate generation process during Part 2. However most of this process was measured (TPR12 - TPS04) and the remaining time Delta1 represents the time taken to generate a user window in the demonstrator, and can be considered as negligible. A similar technical problem prevented a processing delay estimate being calculated for the generation of the session key at user A. This measurement is represented in Figure 6.11 as Delta2 and corresponds to one exponentiation.

The measured delays during Part 2 are shown in Table 6.5 (seconds.milliseconds).

TPR09	TPS03	Time at A	TPR12	TPS04	Time at TA	Time at A and TA
19.570	19.630	0.060	26.990	37.640	10.650	10.710
21.380	21.380	0.000	50.760	01.470	10.710	10.710
04.310	04.310	0.000	15.510	26.220	10.710	10.710
18.620	18.620	0.000	41.360	52.070	10.710	10.710

Table 6.5 - Measured delays during Part 2

The average processing delay during Part 2 (excluding timings Delta1 and Delta2) is thus 10.710 seconds.

Parts 3 and 4 : Message transmission from A to B and certificate generation in B's domain

The measurement points for Parts 3 and 4 are indicated in Figure 6.12.

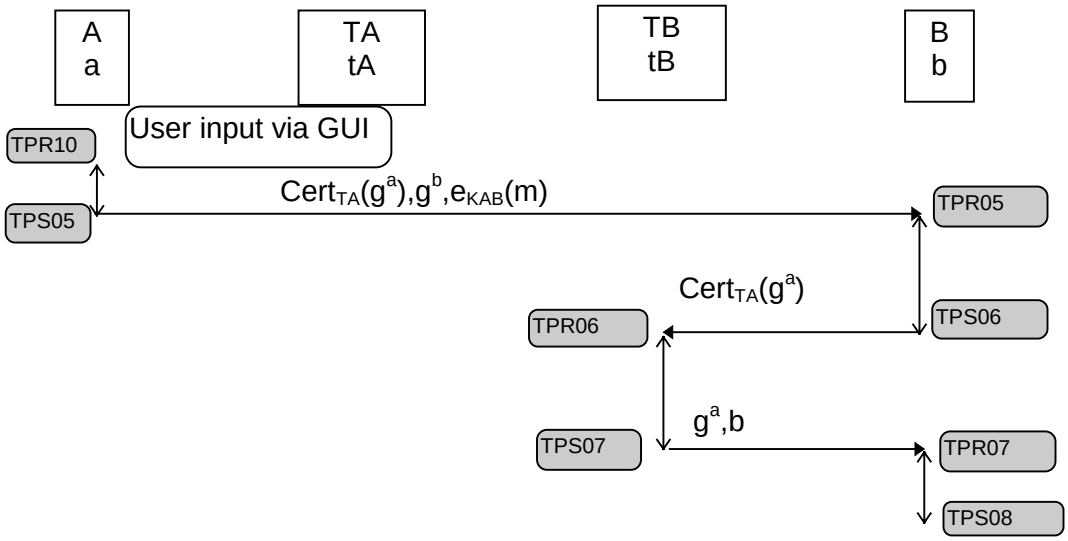


Figure 6.12 - Measurement points for Parts 3 and 4

The first measured delay was that of the sending user (user A) as they prepare the message to be sent to B (TPR10 - TPS05). This processing includes one encryption and the delays are shown in Table 6.6 (seconds.milliseconds).

TPR10	TPS05	Time at A
01.530	01.960	0.430
11.970	12.410	0.440
17.570	17.950	0.400
37.870	38.250	0.470
35.320	35.650	0.330

Table 6.6 - Delay at A during Parts 3 and 4

The average processing delay at A during Parts 3 and 4 is thus 0.414 seconds.

At the receiving end there are two sources of processing delay - user B and TB. The delay at TB (TPR06 - TPS07) is largely taken up by a certificate verification and the measured delays (seconds.milliseconds) are shown in Table 6.7.

TPR06	TPS07	Time at TB
03.610	14.930	11.320
13.950	25.320	11.370
19.490	30.860	11.370
39.950	51.380	11.430
37.300	48.670	11.370

Table 6.7 - Delay at TB during Parts 3 and 4

The processing delay at TB during Parts 3 and 4 is thus 11.372 seconds.

The processing delay at B consists of two stages. Firstly B extracts a certificate for the message received from A and forwards this to TB (TPR05 - TPS06). Then later B computes the session key from information received from TB and decrypts the message (TPR07 - TPS08). The measured processing delays at B are indicated in Table 6.8 (seconds.milliseconds).

TPR05	TPS06	Time at B	TPR07	TPS08	Time at B	Total time at B
02.020	03.500	1.480	15.200	19.930	4.730	6.210
12.410	13.900	1.490	25.540	30.320	4.780	6.270
17.950	19.380	1.430	31.080	35.800	4.720	6.150
38.360	39.850	1.490	51.490	56.320	4.830	6.320
35.710	37.190	1.480	48.830	53.560	4.730	6.210

Table 6.8 - Delay at B during Parts 3 and 4

The average processing delay at B during Parts 3 and 4 is thus 6.232 seconds.

Table 6.9 indicates the total processing delay involved at the receiving end (TPR05 - TPS08). This delay include processing at both user B and TB.

TPR05	TPS08	Time at B and TB
02.020	19.930	17.910
12.410	30.320	17.910
17.950	35.800	17.850
38.360	56.320	17.960
35.710	53.560	17.850

Table 6.9 - Total delay at B and TB during Parts 3 and 4

The average processing delay at B and TB during Parts 3 and 4 is thus 17,896 seconds.

6.6.1.2 Analysis of processing delays

From Tables 1 to 5 in the previous section we can calculate the total measured processing delays at each of the entities involved in the demonstrator. These are exhibited in Table 6.10.

A	TA	TB	B
0.414	10.700	11.372	6.232

Table 6.10 - Total processing delays at each entity during Parts 1 to 4

There has been no attempt to provide best possible timings by running the demonstrator on suitably fast machines. These timings represent a first measurement of processing time and their main relevance is as an indication of the balance of the demonstrator performance. It is suggested that running the demonstrator on an INTEL Pentium 75 Mhz would immediately reduce the measured timings by about 40%.

There is a marked difference between the total measured processing time at A and B. This difference may be explained by the combination of a couple of factors:

- The total processing time at user A shown in Table 6.10 does not include the omitted time Delta2 during Part 2. Delta2 includes one exponentiation and is thus a significant measurement.
- The processing delay at B of approximately 1.480 seconds between TPR05 and TPS06, shown in Table 6.8, is not caused by the execution of any cryptographic primitives. It is possible this processing delay can be significantly reduced.

6.6.2 User acceptability

The demonstrator allows the user to configure the communication database and to observe and guide the end-to-end confidentiality key establishment process. In order to estimate the user acceptability of the demonstration several aspects are considered:

The overall appearance: The GUI provides a simple visual representation of the demonstrated protocol through the Monitor bitmaps and an accurate record of all interactions through the Tracer window.

The flexibility: The demonstration is flexible regarding potential misuse by the users. There are messages that appear when the user makes inappropriate selections, in order to help them define the proper sequence of actions.

The sufficiency of the displayed features: The user has the option of configuring the communication database and the entities involved and watching the message flow in an abstract (Monitor option) or concrete way (Tracer option).

The degree of detail of the displayed information: The Tracer window enables the user to observe and understand the internal actions that are not presented in the main GUI window (key generation, certificates, etc.). Also, detailed information on the involved entities' features is included in the Status windows for the user's convenience.

The degree of guidance of the user on the steps he must follow: The menu bar is designed in a way that all the actions can be selected sequentially.

The security functions impact as perceived by the user: The duration of the security functions execution is not perceived by the user.

6.6.3 Summary of suggested enhancements

The TTP software for end-to-end encryption met the specifications of [D07] and [D09] and performed well in the demonstrations. There are however a number of suggested enhancements. These have been detailed in [D14]; we provide a brief summary here.

- ***Key escrow protocol***. Possible enhancements to the basic JMW protocol have been examined in Section 6.4.4, and alternatives reviewed in Section 6.4.5 above. JMW with some of the suggested enhancements appears to be the best of current protocol proposals. Research on key escrow protocol development will continue through work being conducted by the academic partners in the project. Serious reservations concerning technical aspects of practical implementations of any key escrow protocol in an open environment such as UMTS were noted, referring to [AAB97] and [KM97].
- ***Smart card based security.*** The security functionality in the demonstration is implemented in software on the demonstration PCs. However, the security of the protocol relies on a trustworthy implementation, which protects certain cryptographic information. In order to achieve a higher degree of security, parts of the security functionality on the user's terminal could be implemented on a separate, trusted, tamperproof security module or smart card.
- ***Security services***. More TTP services, functions and internal operations could be provided to support a wider range of end user security services.
- ***Architecture***. The implemented architecture could be developed to provide closer alignment general structure outlined in [D07] to enable standardised interfacing to a variety of TTP and client environments. A more complete and flexible TTP security information storage configuration could also be provided as well as a TTP Service API.
- ***Demonstrator appearance***. Suggested enhancements include the addition of an on-line help, improvements in Monitor message display, extra Tracer options to permit off-line viewing and an automatic protocol execution option. Note that other minor refinements to the demonstrator could include an explicit demonstration of key recovery (rather than an implicit one) and an option to slow

down certain screen operations (such as message arrow display) to ease visual comprehension of the protocol message sequence.

6.7 Conclusions

The general aim of the ASPeCT TTP work was to identify and develop relevant TTP services within UMTS by specification and development of working demonstrators. The main achievements can be summarised as follows:

- identification of key TTP services relevant to UMTS;
- analysis and specification of a possible UMTS key escrow service;
- design and development of a demonstrator providing a UMTS key escrow service;
- specification of a compact public-key certificate format for UMTS;
- analysis and specification of several CA services for UMTS;
- design and development of a demonstrator providing a TTP-assisted secure billing service;
- conducting a trial of the TTP-assisted secure billing service.

There is no doubt that the ability to implement public key cryptographic techniques will greatly increase the number of security services in UMTS with respect to previous systems that relied on symmetric key cryptography. The vast majority of these new services are impossible without a public key infrastructure in place, and using TTPs in the role of CAs, to act as guarantors of public keys is becoming a well accepted component of such an infrastructure. The technical ASPeCT work on certificate design, cross certificate usage, and protocol specification, all further add to the increasing knowledge-base concerning use of TTPs to support public key cryptography in the provision of security services. The development of the secure billing demonstrator shows that providing an incontestable charging service within UMTS through an on-line TTP is both realistic and achievable in an efficient manner.

It is worth re-emphasising the ASPeCT project premise that ASPeCT would demonstrate the provision of key escrow services through the assistance of TTPs, *given that such a service is required*. It is worth repeating that it was not a central objective of ASPeCT to directly add to the debate over whether key recovery is a *desirable* service. The technical ASPeCT work on key escrow has shown that it is indeed theoretically possible to develop an escrow mechanism that is applicable to UMTS, and an amount of analysis has been conducted on how best to design a protocol that would be satisfactory to most of the involved entities. It has however proved very difficult to determine exactly what technical and implementation restrictions a UMTS key escrow would be required to operate within, since at the time of writing there is little consensus as to the practicalities, and indeed legalities, of providing such services in a telecommunications environment.

The work has shown that TTPs can, and undoubtedly will, be key security service providers for a wide variety of services within UMTS.

7 Security and integrity of billing in UMTS

7.1 Introduction

The objective of the ASPeCT secure billing work was to examine how UMTS can meet the requirements of users, service providers and network operators regarding the security and integrity of billing information. Various billing scenarios were investigated and the precise requirements of the involved parties were stated in order to help identify the important security issues. The ASPeCT work selected one particular secure billing scenario and aimed to verify the feasibility and acceptability of a particular secure billing solution for this scenario by conducting demonstrations and trials.

Existing networks address the issue of secure billing to various degrees. A weak form of ensuring the integrity of billing, that may help to settle some disputes, is itemised billing. Another measure that may help to instil the confidence of a subscriber in a bill is advice of charge. In GSM more sophisticated security measures (e.g. authentication based on cryptographic procedures), which are used to counter fraud, may further increase the confidence of subscribers in the correctness of the bill.

However, none of these measures produces hard evidence that may allow a third party (such as a judge) to settle a dispute between a subscriber and a service provider. Also, these measures do not address the aspects of secure billing between network operators and service providers. For these purposes, stronger forms of protection are required in the future. Stronger forms of protection may also contribute to increased confidence in the proper working of the system by all parties involved and may greatly reduce the number of cases where disputes arise.

A detailed investigation into the security issues and requirements arising from various envisaged UMTS billing scenarios was presented in [D02]. In this report threat and risk analyses were performed for the scenarios, taking into account the views of all the parties involved. A semi-formal concept of obligations in open telecooperation was then introduced in order to capture the requirements on the non-repudiation services needed to secure the scenarios. This was followed by an examination of the trust relations between the involved parties. This allowed the requirements on secure billing services in the considered scenarios to be derived. These are stated in terms of the evidence which needs to be provided to the involved parties in the course of the provision of a service so as to make the billing of the service secure.

The ASPeCT work has primarily focused on one particular security billing scenario involving the provision of premium rate services by a value added service provider. This scenario was chosen since it allowed the project to cover some of the most important security issues associated with billing scenarios in UMTS. More specifically it allowed the project to investigate suitable methods for increasing the confidence of users in the correctness of the billing process in a situation where trust among operators, providers and users can no longer be taken for granted. This scenario was also chosen because it was felt that it had the greatest potential for innovation.

For the selected secure billing scenario the requirements amount to the ability of the user to generate proof that he used a certain service in a certain way. In the ASPeCT project this proof was provided through the use of non repudiation security services based on digital signatures and micropayment techniques. These services were provided with the support of TTP infrastructure to manage and certify the various signature keys.

7.2 Secure billing for value-added information services

In this section we review the model for payment of value-added services adopted by ASPeCT and the micropayment charging approach on which the resulting protocols are based. We then describe in more detail the protocols designed by ASPeCT. We also conduct some comparative analysis of the ASPeCT solution with respect to other related proposals.

7.2.1 A payment model for value-added information services provision

The principal roles in the ASPeCT payment model are those of the mobile user (*buyer*), the VASP (*vendor*) and the UMTS service provider (*broker*). The relationships between these roles in the ASPeCT payment model are indicated in Figure 7.1.

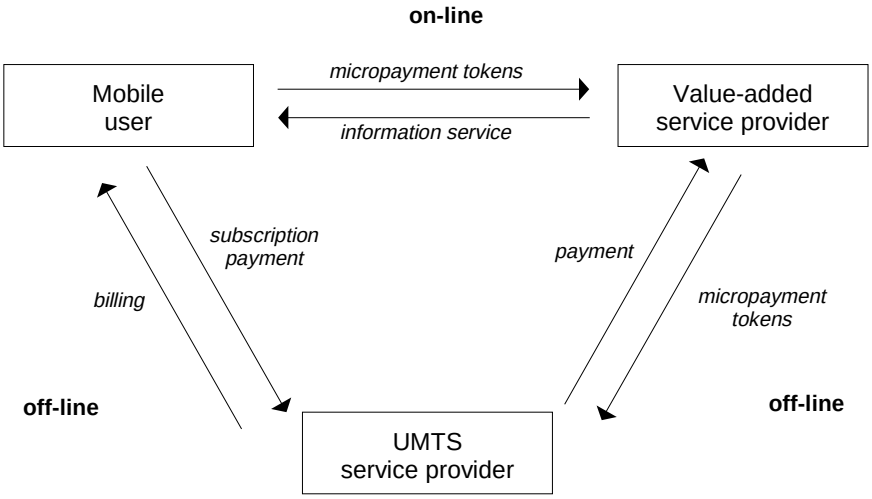


Figure 7.1 - Charging model

In the ASPeCT charging model the only on-line communication required in the charging procedure is that between the mobile user and the VASP. The VASP provides information to the user and sends charge requests. The user pays by sending micropayment tokens as described in Section 7.2.3. The VASP is able to check the validity of these tokens based on a certificate on the user's credentials issued by the UMTS service provider, or a TTP acting on behalf of the UMTS service provider.

The VASP forwards the information proving the claims on the user to the user's UMTS service provider off-line, who in turn bills the user off-line. The UMTS service provider takes care of the payments to the network operators involved in providing the required bearer service.

No previous contact between the VASP and the mobile user's UMTS service provider is required as long as the VASP can satisfactorily verify the user's certificate. However, an existing business arrangement between the VASP and the UMTS service provider would be advantageous in facilitating clearance of the payment. Payment clearance would typically take place at regular intervals, perhaps daily or weekly.

The UMTS service provider plays the role of a broker. He provides the user with a means to pay electronically and vouches for the credit-worthiness of the user by issuing a certificate for him. New certificates could be issued periodically, perhaps monthly. If a bill is not paid the validity of the current certificate can be withdrawn. The UMTS service provider bills the user and is paid by the user through established telecommunications billing procedures. The UMTS service provider then forwards the due share to the VASP.

A major advantage of implementing this type of payment scheme in a telecommunications environment is that the broker/banking infrastructure for billing the user and paying the vendor is already in place.

7.2.2 Specific requirements on the charging scheme

Suitable payment systems for UMTS undoubtedly belong to the general class of payment systems suitable for the Internet or the World-Wide Web, more specifically. It is assumed that the protocol used to retrieve the information from the value-added service provider is almost certainly the HTTP protocol and that the information is structured in a compatible way with the formats used on the World-Wide Web. However, among these Internet oriented systems only a small subclass are suitable for use in a mobile environment. The reasons for this are that there are a number of specific requirements within the UMTS environment that restrict the type of payment system that can be adopted. These include the following:

- The value of a particular piece of information retrieved by a user from a VASP at any one time may be quite small. Charging schemes should thus not require a large financial overhead in order to process the charge.
- Selected protocols and cryptographic mechanisms must be chosen in such a way that they are particularly suited to the low bandwidth of mobile systems and low computational capabilities on the user's smart card. The payment protocol itself must therefore be very light-weight.
- For reasons of efficiency it is highly desirable that payment for value-added services and payment for basic telecommunication services are efficiently combined by integrating the initialisation of the payment process with the call set-up procedure in UMTS.
- It should be possible for the roles in the payment scheme to be played by parties already active in today's mobile networks, namely mobile users, mobile (GSM, UMTS) service providers and value-added service providers. Their existing business relationships, in particular the existing infrastructure for billing users, may also be used for the new payment scheme. No additional clearing network of financial institutions like banks or credit card organisations is needed.
- The payment scheme should work off-line to avoid creating additional signalling load.
- The charging for today's VASs consists of a basic charge for the basic service and a premium for the added value. Both charges are based on the duration of the call. In the future, due to the greater variety of services on offer more *flexible* charging schemes for the premium would be desirable. Flexibility relates to the parameters which determine the charge (in addition to the duration of the call, the charge could depend on the amount of data transferred), to the variety of different possible tariffs and to the ease with which a certain tariff can be changed.
- The scheme should have a performance compatible with the requirements of a mobile system. In short, the charging scheme must be *efficient.*
- It is expected that the evolution of current mobile systems towards UMTS will also see the emergence of many new network operators, UMTS service providers and VASPs, which may have serious implications for the trust relationships between them. Thus, the charging scheme must be *secure* against cheating (such as overcharging by the VASP or underpaying by the user), and the parties involved should have the assurance that justified claims relating to charges can be proved and that unjustified claims cannot be successfully made. This is often called *incontestable charging.* More precisely:

  From the payer's point of view:

  - a payment in his name can be made only by him;
  - the amount of the payment is exactly what the payer has specified;
  - only the payee specified by the payer can receive the payment.

  From the payee's point of view:

  - a payee can verify the correctness of a payment;
  - the payer cannot deny having made a verified payment;
  - the payee can be certain of being credited for verified payments by the broker.

  From the broker's point of view:

  - the broker can verify the correctness of a payment.

ASPeCT designed and demonstrated a proposed charging scheme for VASs in UMTS that satisfies almost all of the above requirements. The charging scheme is implemented as a credit-based payment scheme using *micropayments* (see Section 7.2.3), although it is also possible to use the scheme on a pre-paid basis.

7.2.3 Review of micropayment approach

Micropayment schemes are electronic payment schemes that are proposed explicitly for the payment of items costing very small amounts. The interest in such schemes is largely based on the need to develop efficient methods of electronically paying for items such as information on the web or units of telephone connection time. In the last few years there have been many new proposals for micropayment schemes, many of which are currently being developed and put on trial as commercial products.

In most micropayment schemes there are three entities. A *user* purchases goods from a *vendor*, and the transaction may or may not also require the participation of a *broker* (or *acquirer*) who may be in contact with a *bank*, or indeed in some cases be a bank. In some of the schemes both the user and the vendor have their own servers who communicate with one another, or both communicate directly with a bank. In our payment model, the mobile user is the buyer, the VASP is the vendor and the UMTS service provider, or a TTP acting on its behalf, is the broker.

There are a couple of general design assumptions that characterise micropayment schemes.

1. The cost of communication and processing costs of a micropayment should be kept as low as possible, since otherwise it may not be economical to collect the charge for a micropayment at all.
2. Since potential losses over short periods in a micropayment scheme are low, it may be possible to sacrifice the full security requirements of a payment protocol in order to gain increases in efficiency of operation and decreases in cost. Thus large scale abuse of a micropayment scheme should be prevented, but limited small scale fraud may be worth tolerating.

Several techniques can be used to reduce communication and processing costs. These include limiting the use of (expensive) public key cryptographic techniques (such as digital signatures), processing (clearing) transactions off-line, and processing (clearing) transactions in batches.

Several techniques are used to reduce the need for tight security requirements. These include only checking for user overspending on a daily basis, leaving vendor fraud to be dealt with by the market forces, and setting daily limits on user spending, but being prepared to write off user fraud up to that limit.

A wide variety of micropayment schemes have been recently proposed. In ASPeCT we chose to implement a scheme based on the core idea often attributed to [P95]. In [P95] each unit micropayment was referred to as a *tick*. The "tick" concept uses chains of pre-images of a one-way function F to make micropayments. This idea was independently suggested by several authors. Similar and related schemes to [P95] include PayWord [RS96], NetCard [AMS96], Micro iKP [HSW96], and an extension of the idea in PayTree [JY96]. The basic idea in all of these schemes is very simple. The user starts with a starting value α, commits to the n-th image $F^n(\alpha)$ of the starting value, and then pays for the i-th micropayment (or tick) by releasing the pre-image $F^{n-i}(\alpha)$. This is essentially the concept first developed by Lamport [L81] for one-time passwords, and by Winternitz (see Merkle [M90]) for one-time signatures. The major advantage of this technique is that, as required by the charging scheme requirements listed in Section 7.2.2, computationally expensive public key signatures are avoided during the payment mechanism, and replaced by computations of a one-way function.

7.2.4 Protocol design

In this section we give a brief overview of the secure billing protocols in ASPeCT. We will describe the central protocol implemented in the secure billing demonstrator and trial. Detailed design analysis and several variants of the protocol have appeared in a number of publication [HHM98], [HP98], [M98], [MPM98] to which we will periodically refer for further details (see Section 7.2.4.4).

7.2.4.1 Structure of charging process

The charging consists of two phases:

- In the *authentication and initialisation phase*, the user and the VASP authenticate one another, and the user commits to a starting value for the micropayment scheme and a certain tariff by performing a digital signature on corresponding data.
- In the *data transfer phase*, the user pays by releasing the pre-images of the starting value, so-called "ticks", which represent unit charges. The value of one unit charge is agreed upon in the initialisation phase. The "ticks" serve as proof to the VASP that the user incurred certain charges, because only the user could have generated them. They will be presented off-line by the VASP to the user's UMTS Service Provider (via the Network Operator) to settle the charges.

There are two variants of the payment process: *on-line*, meaning that a TTP is involved on-line in the payment process, in addition to the user and the VASP; *off-line*, meaning that the user and the VASP are the only entities involved on-line in the payment process. There are three versions of the authentication protocol which is run in the initialisation phase: versions A, B and C. Versions A and B are very similar. They realise the off-line version of the billing protocol as implemented in the demonstrator and are described in [D16], [HP98], and [M98]. Authentication protocol version C, which we describe here, realises the on-line variant of the payment process.

7.2.4.2 Authentication and initialisation phase

7.2.4.2.1 Goals

The goals to be achieved at the end of a successful run between user *U* and VASP *V* are:

1. mutual entity authentication of U and V;
2. agreement of a secret key between U and V (with joint key control);
3. mutual implicit key authentication;
4. mutual key confirmation;
5. mutual assurance of key freshness;
6. non-repudiation by U of data sent to V;
7. confidentiality of the identity of U over the air interface;
8. initialisation of the payment mechanism;
9. distribution of certified public keys for U (to V) and V (to U),
10. assurance for U and V that the certificates on the public keys of V and U (respectively) are valid and have not been revoked.

7.2.4.2.2 Prerequisites and choice of cryptographic parameters

For the **authentication and initialisation of payment protocol**, we have the following prerequisites and choice of cryptographic parameters:

- There is an elliptic curve cryptosystem E over $GF(p)$ whose parameters p (prime defining the field), q (size of a cyclic subgroup of the curve), g_x and g_y (co-ordinates of a generator g of the cyclic subgroup of the curve), a and b (coefficients of the defining equation) are configurable. As default values, the parameter values in [ISO14888-3] Annex C.2 are taken, where the cardinality q (of the cyclic subgroup of the elliptic curve) is in the order of 2^{129}.
- There is a function f mapping E onto the numbers in the range $[0..q-1]$. It is $f(Z) = Z_x \bmod q$.
- V has long-term secret and public key agreement keys v and g^v respectively, where v is a number in the range $[0..q-1]$, and $g \in E$ is as above.

- *U* possesses an asymmetric signature system with secret signature transformation Sig_U, secret key KU^- and public key KU^+. It is an AMV signature system based on the above elliptic curve *E*, as described in [ISO14888-3]. $Sig_U(M)$ denotes only the appendix.
- *U, V* and *T* possess a symmetric encryption function, where $\{ M \}_K$ is the encryption of message *M* with key *K*. We assume that the encryption algorithm is resistant against known cryptanalytic attacks such as code book attacks and chosen plaintext attacks. This function is DES-CBC.
- *U* and *V* possess a (pseudo-)random number generator. This RNG uses DES-OFB.
- *U* and *V* possess functions *h1*, *h2* and *h3* with the following properties:

  The functions *h1*, *h2*, and *h3* are compression functions (i.e. they map inputs of arbitrary finite lengths to fixed length outputs) which are easy to compute and satisfy:

  1. *h1* is a partial-preimage resistant, weakly computation resistant and weakly pseudo-random function.
  2. *h2* is a partial-preimage resistant, weakly computation resistant function.
  3. *h3* is collision resistant.

  (A function *h* is *weakly computation resistant (weak MAC-property)* if it is computationally infeasible to find a pair *(x, h(K, x))* without knowing *K*, provided that no other pair *(x', h(K, x'))* is known. A function *h* is *weakly pseudo-random* if, for secret random key *K* not used before and for known random *x* the output *h(K, x)* is indistinguishable from a random output.)

  The choices are as follows:

  h1 = h3 = RIPEMD-128, *h2(x)* = trunc*(40, h3(x))* where trunc*(n,y)* returns the *n* least significant bits of *y*.
- The identity *idV* of *V* is assumed to be known to *U* at the start of the protocol.
- There are parameters *ch_data*, *TV*, α_T and *IV* with the following significance: *ch_data* is data sent from *V* to *U* describing the applicable tariff, *TV* is a time-stamp generated by *V* (the UTC time of *V*), α_T and *IV* are random values generated by *U* for use in the payment protocol.
- The TTP possesses an asymmetric signature system with secret signature transformation Sig_T, secret key KT^- and public key KT^+. If this is an asymmetric signature system with appendix then $Sig_T(M)$ denotes only the appendix.
- The TTP possesses the function *h3*, as defined above.
- The TTP possesses a public key agreement key g^w.
- *CertChain (A, B)* is a certificate chain on the public key of *B* which can be verified by an entity in possession of the public key of CA_A. (*A, B* may take the values *U, V; B* may also take the value *T*). If CA_A and CA_B coincide then *CertChain (A, B) = CertB.*
- *TT* is a time-stamp issued by the TTP (the UTC time of TTP T).
- *cidA* is a unique identifier of a certificate on a public key of user *A*, e.g. a combination of the identity of the issuer and the serial number.
- It is assumed that the user's identity is sufficient for the TTP to retrieve the appropriate certificate and that the user is in possession of the public key KT^+, and of the public key agreement key g^w.
- There is a public system parameter *T* which gives the maximum number of ticks to which the user can commit himself by one signature. *T* is equal to at most 2^{16}. For performance reasons, *T* may be set to a lower value. The default value for *T* is 2^{10}.

- There is a public family F of length-preserving one-way functions FIV: {0,1}n → {0,1}n, where n is a public system parameter and IV is an initialisation vector. (To be more precise, the function FIV needs to be one-way on T iterates [KMP98].)

The choices for the first and second secure billing demonstrators are $n = 64$ and $F_{IV}(x) = \text{trunc}(64, h(IV||x))$, where h is RIPEMD-128.

7.2.4.2.3 Authentication and initialisation of payment protocol

The protocol is executed between a user *U* and a VASP *V*, and between the VASP *V* and a TTP *T*. It is assumed that *U* and *V* do not have authentic copies of one another's public keys or valid certificates on their own public keys. The on-line TTP provides the certificates to the user and the VASP. For stronger starting assumptions (variants A and B) see [D16], [HP98] or [M98].

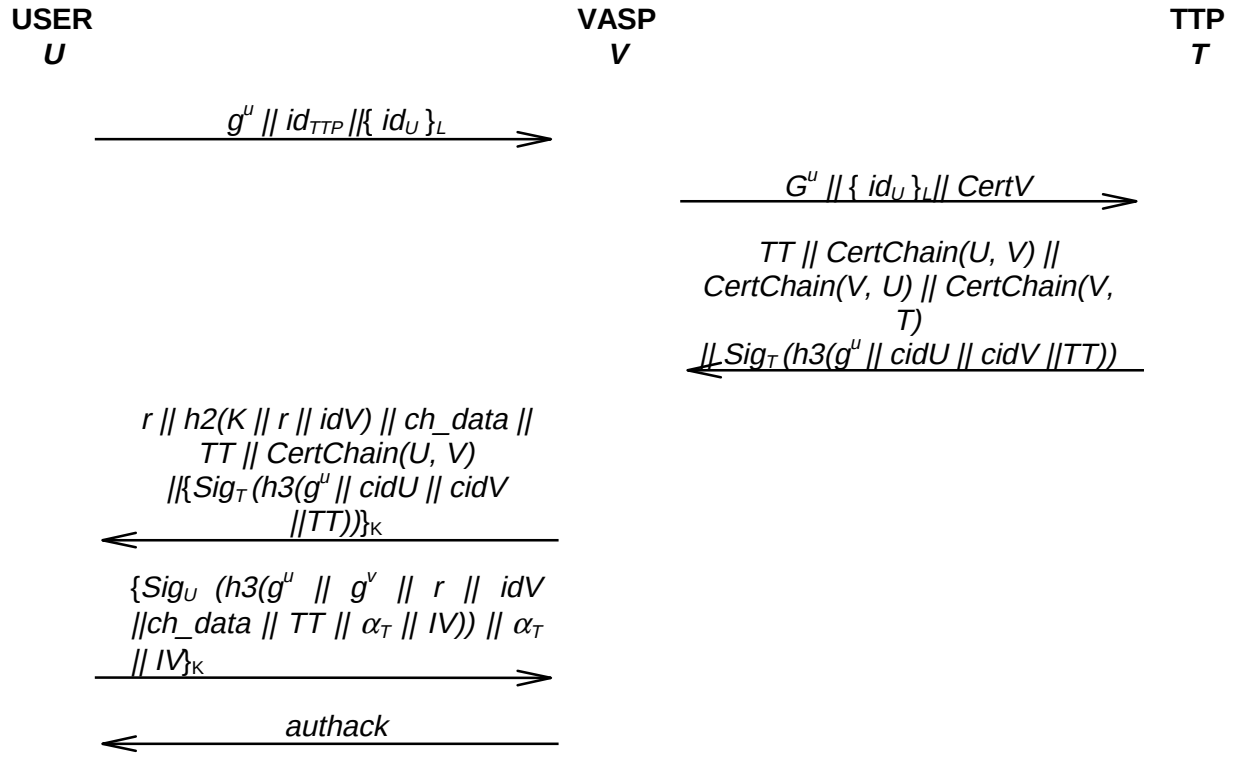


Figure 7.2 - Authentication and initialisation of payment protocol

Operations at *U*:

1. generate random number u
2. compute g^u
3. compute $L=g^{uw}$
4. compute $\{ id_U\}_L$
5. send message $U \rightarrow V$: authreq: $g^u \,||\, id_{TTP} \,||\, \{ id_U\}_L$
6. store u, g^u for later use in the protocol.

Operations at *V*:

7. send message $V \rightarrow T$: ttpreq: $g^u \,||\, \{id_U\}_L \,||\, CertV$ to TTP with identity id_{TTP} sent in the first message.
8. store $cidV$, g^u

Operations at *T*:

9. decrypt $\{ id_U\}_L$

10. generate time-stamp *TT*
11. generate certificate chains *CertChain(U, V), CertChain(V, U)* and *CertChain(V, T)*
12. generate signature $Sig_T(h3(g^u \,||\, cidU \,||\, cidV \,||TT))$
13. send message $T \rightarrow V$: ttpresp:
    $TT \,||CertChain(U, V) \,||\, CertChain(V, U) \,||\, CertChain(V, T) \,||\, Sig_T(h3(g^u \,||\, cidU \,||\, cidV \,||TT))$

Operations at *V*:

1. verify *CertChain(V, U)*
2. retrieve *CertU* from *CertChain(V, U)* and *cidU* from *CertU*
3. retrieve public key of *T* from *CertChain(V, T)*
4. verify $Sig_T(h3(g^u \,||\, cidU \,||\, cidV \,||TT))$
5. generate random number *r*
6. compute $K := h1(f((g^u)^v)||r)$
7. compute $h2(K \,||\, r \,||\, idV)$
8. compute $\{Sig_T(h3(g^u \,||\, cidU \,||\, cidV \,||TT))\}_K$
9. send message $V \rightarrow U$: authcont: $r \,||\, h2(K \,||\, r||\, idV) \,||\, ch\_data \,||\, TT \,||CertChain(U, V) \,||\, \{Sig_T(h3(g^u \,||\, cidU \,||\, cidV \,||TT))\}_K$
10. store *TT, ch_data, K, CertU*

Operations at *U*:

11. verify *CertChain(U, V)*
12. decrypt $\{Sig_T(h3(g^u \,||\, cidU \,||\, cidV \,||TT))\}_K$
13. verify $Sig_T(h3(g^u \,||\, cidU \,||\, cidV \,||TT))$
14. compute $K := h1(f((g^v)^u)||r)$
15. compute $h2(K \,||\, r \,||\, idV)$
16. display *ch_data* on the screen in a separate window; display warning on the screen if the difference *TT-TU* (*TU* = UTC time of *U*) is greater than a pre-defined *delta_t*; authcont_check = OK if received $h2(K \,||\, r \,||\, idV)$ equals $h2(K \,||\, r \,||\, idV)$ computed in step 28 and if *ch_data* is confirmed by human user via GUI (can be switched off).
17. generate random α_0 and random *IV* and compute $\alpha_T = F_{IV}{}^T(\alpha_0)$
18. compute $Sig_U(h3(g^u \,||\, g^v \,||\, r \,||\, idV \,||\, ch\_data \,||\, TT \,||\, \alpha_T \,||\, IV))$
19. compute $\{Sig_U(h3(g^u \,||\, g^v \,||\, r \,||\, idV \,||\, ch\_data \,||\, TT \,||\, \alpha_T \,||\, IV)) \,||\, \alpha_T \,||\, IV\}_K$
20. send message $U \rightarrow V$: authresp: $\{Sig_U(h3(g^u \,||\, g^v \,||\, r \,||\, idV \,||\, ch\_data \,||\, TT \,||\, \alpha_T \,||\, IV)) \,||\, \alpha_T \,||\, IV\}_K$
21. store *ch_data, K,* α_T*, IV*

Operations at *V*:

22. decrypt $\{Sig_U(h3(g^u \,||\, g^v \,||\, r \,||\, idV \,||\, ch\_data \,||\, TT \,||\, \alpha_T \,||\, IV)) \,||\, \alpha_T \,||\, IV\}_K$
23. retrieve KU^-. If $Sig_U(h3(g^u \,||\, g^v \,||\, r \,||\, idV \,||\, ch\_data \,||\, TT \,||\, \alpha_T \,||\, IV))$ verifies, set authresp_check = OK
24. store $Sig_U(h3(g^u \,||\, g^v \,||\, r \,||\, idV \,||\, ch\_data \,||\, TT \,||\, \alpha_T \,||\, IV))$, *idU,* $g^u \,||\, g^v \,||\, r \,||idV \,||ch\_data \,||TT \,||\, \alpha_T \,||\, IV$
25. set $j \leftarrow T$, *tck_cnt* $\leftarrow 0$, $\alpha \leftarrow \alpha_T$
    / Initialisation of parameters of tick payment protocol. */*

26. send message $V \rightarrow U$: authack: *{ }*
    / message contains no user data */*

Operations at *U*:

27. set $j \leftarrow T$, $tick\_total\_U \leftarrow 0$
    / Initialisation of parameters of tick payment protocol. */*

Remarks:

1. Slightly simpler variants of this protocol are possible if the identity of the user is not encrypted in the first message and it is assumed that the UMTS air interface is protected by underlying mechanisms. These have been implemented in previous demonstrators [D16], [D19].
2. The purpose of the signature sent in the third message differs for user and VASP: The VASP gets assurance that the user's certificate was not revoked at the time given by the time-stamp. He does not interpret the unique identifier of his own certificate nor the parameter g^u. It is assumed that the VASP has a reliable clock with which he can compare the time-stamp. The user, on the other hand, cannot generally be assumed to have access to a reliable clock (the protocol may be executed on a smart card). By including g^u in the signature, the user gets assurance that the time-stamp was created during the current protocol run and that the VASP's certificate was not revoked before the start of the current protocol run.
3. It is assumed that the user and the VASP know the unique identifiers of the certificates on their own public keys, so that they are able to verify the signature. If this may not be assumed then this unique identifier number has to be included in the third and the fourth messages in the clear. *(If cidU and cidV are of the form described in the section on "prerequisites" above then cidV is contained in CertChain(U, V) and cidU is contained in CertChain(V, U), so there would be only be a need to additionally include cidU in the fourth message*)
4. A short protocol can be run to re-initialise the payment once the maximum number of ticks has been exceeded. Details can be found in [D16].

7.2.4.3 Data transfer phase

7.2.4.3.1 Goals

See remarks on incontestable charging in Section 7.2.2.

7.2.4.3.2 Prerequisites

See Section 7.2.4.2.2.

7.2.4.3.3 Charge ticks protocol

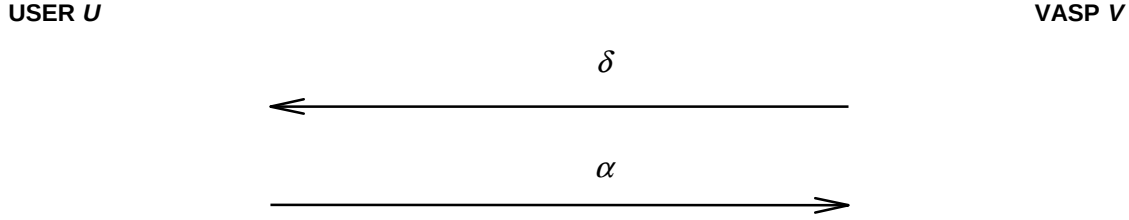


Figure 7.3 - Charge ticks protocol

Recall that initialisation of parameters is done at the end of the authentication and the re-initialisation protocols respectively.

Notation at *U*:

/ T - j is the number of ticks already sent by U; tick_total_U is the number of ticks which has to be paid by U - according to the count of U; δ is the number of ticks whose payment is requested by V in the current run of the tick payment protocol. */*

Notation at *V*:

/ T - j is the number of ticks requested by V; tck_cnt is the number of ticks correctly received by V;. α is the last tick received by V. */*

Operations at *V*:

If $j \geq \delta$ then only the charge ticks protocol is run. If $j < \delta$ then V splits the request for δ ticks in two requests: First the charge ticks protocol is run and V requests j ticks to spend the complete agreed set of ticks. Then the re-initialisation of payment protocol is run, and then again the charge ticks protocol is run, and V requests $\delta - j$ ticks.

If $j \geq \delta$ then the charge ticks protocol is run, else the re-initialisation of payment protocol is run.

1. Set $\alpha' \leftarrow \alpha$
2. Set $j \leftarrow j - \delta$
3. Send message $V \rightarrow U$: ticks: δ

Operations at *U*:

If $(j \geq \delta)$ and $(tick\_total\_U + j - T \geq \delta)$ then chtickreq_check = OK
/ On the user side, $j < \delta$ should not occur in a correct protocol run because the VASP also checks $j \geq \delta$. */*

4. Set $j \leftarrow j - \delta$
5. Set $\alpha \leftarrow F^j(\alpha_0)$
6. Send message $U \rightarrow V$: chtickresp: α

Operations at *V*:

If $(\alpha' = F^\delta(\alpha))$ then chtickresp_check = OK

7. Set $tck\_cnt \leftarrow tck\_cnt + \delta$
8. Store α, *tck_cnt*
   / α, tck_cnt should be continuously stored, not only at the end of the session, otherwise the paid ticks are lost when the program crashes. Previous values are overwritten. */*

7.2.4.4 Variants and references

In this section we discuss further references to publicly available documentation concerning the ASPeCT secure billing protocols. It should be noted that the flexibility of the ASPeCT authentication and initialisation protocol has led to a number of slight variants existing in different documentation. We clarify the differences here.

Firstly, detailed descriptions of all the protocols relating to the first and second demonstrators can be found in [D16] and [D19] respectively. The variant appearing in [D19] is a simplified version of that appearing in Section 7.2.4.2.3. This simplification relates to the fact that it is assumed that anonymity of the air interface is protected by other than cryptographic means (see Section 7.2.4.2.3, Note 1).

A general overview of the protocol incorporated in the second demonstrator can be found in [MPM98], and another overview that incorporates configurations for the trial is [HHM98]. Both of these papers describe a minor variant of the protocol in Section 7.2.4.2.3 which uses the key *L* to protect the message from the VASP to the user against *content verification attacks* (see Section 7.2.5.2.8). The protocol described in Section 7.2.4.2.3 achieves this in a similar, but simpler, way.

A more detailed discussion of the choice of some of the cryptographic parameters, and explanation as to why the protocols achieve their goals are given in [HP98]. Again, this paper contains a minor variant of the protocol in Section 7.2.4.2.3. The variant of [HP98] has three modifications:

1. *CertChain(V,U)* and the signature in the message from the TTP to the VASP are encrypted with key *L* .
2. The encryption of the signature in the message from the VASP to the user is with key *L* (not key *K*).
3. Key *L* is included in the signature sent from the user to the VASP.

These changes are all made to protect against *bad VASP replay attacks* (see Section 7.2.5.2.9) and also to introduce the additional feature of confidentiality over the VASP-to-TTP interface. This extra confidentiality is not obviously a UMTS requirement, and was not specified in Section 7.2.4.2.1 as a protocol goal, but the existence of such a variant is still worth noting. A further variant on the protocol of Section 7.2.4.2.1 which protects against bad VASP replays attacks without offering confidentiality over the VASP-to-TTP interface is described in Section 7.2.5.2.9.

Detailed discussion of the use and design of suitable one-way functions for use in the charge ticks protocol can be found in [KMP98] (see also Section 7.2.5.2.7). A comparison of the ASPeCT protocol with other related protocols has been conducted in [HMM98] (see also Section 7.2.5.3).

7.2.5 Theoretical evaluation and alternative schemes

In this section we describe theoretical evaluation work of the ASPeCT protocol suite and discuss alternative protocols existing in the scientific literature.

7.2.5.1 Theoretical evaluation work

In general it is very difficult to practically prove the security of a cryptographic protocol, although this is an increasingly active research area, for example [BJM97]. Even when a model can be developed within which the security of a protocol can be proven, this does not guarantee that there are no practical attacks on the protocol that can not be described within the parameters of the chosen security model. This does not mean that such security proofs are not worth pursuing, but it does mean that there is still no really practical substitute for gaining confidence in a protocol than exhaustive testing, careful study and continual analysis as threat models and new attack methods develop.

The basic authentication protocols, from which the ASPeCT protocol suite developed, were first published in 1995 [ETSI95a]. Since then the protocols have been developed, enhanced and scrutinised, and variants have been published [HHM98], [HP98], [M98], [MPM98]. This alone does not prove that the protocols fully meet the goals of Sections 7.2.2 and 7.2.4.2.1, but each part of the process greatly increases confidence that the protocols are indeed robust and fully achieve their target properties.

Two fundamental components of this evaluation process are the accumulation of possible attacks against the protocols, and comparison of the proposed solution with alternative proposals in the literature. We provide a summary of some of this work in the following sections.

7.2.5.2 Possible protocol attacks

We give brief descriptions here of a number of possible attacks against the ASPeCT protocols. Several of the protocol features have been specifically designed to ensure that these attacks are not possible.

7.2.5.2.1 Signer verification attack

During the authentication and initialisation stage it is important that the signature of U is not sent to V in the clear. Doing so can lead to a possible compromise of user anonymity. To understand this potential problem, we consider in turn the two types of signature discussed in the ASPeCT documents.

- **Elliptic curve signature 'with appendix'** (no message recovery). This attack only works in the case that an attacker has access to the message that was signed (or the hashed version of the message that was signed). If the attacker has access to a large set of public verification keys then each key can be used along with the signature and the hashed message in order to see if the verification process is satisfied. If the verification process is not satisfied then it can be immediately deduced that the owner of the public verification key used is not the user U.

- **RSA-type ISO/IEC 9796-2 signature**, [ISO9796-2], (partial message recovery). In this case the same type of attack can be applied, i.e. possible verification keys can be successively applied to the signature. However, in this case it will always be possible to distinguish between the correct and incorrect signature keys, using the built in redundancy of the signature scheme. Thus the only thing necessary for a successful attack in this situation is access to the public keys of the set of likely users.

7.2.5.2.2 Partial chosen key attack

It is tempting to claim that all variants of the protocol offer joint key control since both U and V contribute random values to the computation of session key *K*. It can be claimed however that V has a little more control over the key *K* than the user U. On receipt of message 1, and before replying with message 2, V could generate 2^s variants for the value *r*, and compute $K=h1(g^{uv}||r)$ for each variant. By choosing the one he 'likes best', V can select approximately *s* bits of the key *K*. Of course, the value of *s* is very much constrained by the computing resources of V, and the time interval the user is prepared to wait for a response to message 1.

The most practical way to avoid a partial chosen key attack would be to force V to send a commitment (a hashed value) of his random component before seeing the component chosen by U. Unfortunately this almost certainly involves a redesigning of the protocol with at least one further pass. This problem is not currently regarded as serious enough to merit such action, but the problem is nonetheless worth noting here. Note also that the partial chosen key attack as described here is applicable to most key agreement mechanisms currently proposed in ISO standards.

7.2.5.2.3 Time-memory trade-off attacks

It is important that at no point during any of the protocols is a key hashed on its own and transferred in the clear. To see this, suppose that the value $h2(K)$ is sent from one entity to the other, where *K* is a session key computed using the hash-function $h1$ (and hence *K* might typically have 128 or 160 bits). For the sake of our discussion here, suppose the key *K* contains *k* bits.

A cryptanalyst, who wishes to discover the value of such a key *K*, may choose to compute and store the values $h2(K)$ for a large number of selected values of *K*. For example if the interceptor intends to compute 2^r such values, then he could choose all those whose most significant *k-r* bits are set to some fixed pattern (e.g. all zeros). When the cryptanalyst intercepts a value $h2(K)$, he compares this value with all the pre-computed values, and if a match is found then he has a candidate value for the key *K*. If the length of the hash-output is greater than or equal to $\log_2$(key space size), then there is a reasonable chance that this candidate is the correct key; to simplify our discussion here we suppose that this is the case.

The probability that a match is found will be 2^{r-k}, and thus the cryptanalyst will need to intercept 2^{k-r} messages for every key *K* which is discovered by this method. There is thus a trade-off here between the amount of pre-computation and storage, and the proportion of keys which will be found. This is an example of a *time-memory trade-off*, since the greater the memory (and pre-computation) which the cryptanalyst is prepared to devote to the attack, the shorter the time he will have to wait to find a key *K* . Such attacks against block ciphers were investigated in [BPV98] and [H80].

To avoid such an attack, instead of sending $h2(K)$, a value such as $h2(K \| r)$ should be sent instead, where r is a variable value which will change every time the protocol is executed. This makes it impossible to pre-compute the type of table discussed above.

7.2.5.2.4 Codebook attacks

In general a *Codebook attack* describes the situation where the same data is encrypted twice using the same key, yielding the same ciphertext. If this is observed by an interceptor, then the interceptor can deduce that the two sets of plaintext are the same, even if the plaintext value is not known. Of course, if the plaintext is known for one of the two ciphertexts, then it is immediately known for the other. The computation of the session key K in the ASPeCT protocol completely rules out codebook attacks since the probability of a particular key *K* being computed on two different protocol runs is extremely small.

7.2.5.2.5 Source substitution attacks

The name *Source substitution* attack usually refers to an attack where a malicious entity, *C* say, takes another entity's public key and manages to obtain a certificate stating that the key belongs to *C*. Entity *C* can now take any data string signed by the genuine owner of the public key, and claim that it was signed by *C*. Of course, such an attack can always be avoided by having the Certification Authority (CA) verify that the claimed owner of a public key also owns the corresponding private key. This can be achieved by having the certificate requester create a 'self-signed' certificate for the public key. Such an approach has become widely used in recent years. However, to ensure that a protocol is a secure as possible, it is good practice to ensure that it is not subject to an attack of this type, since one cannot always be sure that the CA will follow recommended best practice.

In the ASPeCT authentication and initialisation protocol such an attack is prevented by including an identifier for the VASP within the scope of the hash-function *h*2.

7.2.5.2.6 Bill theft attacks

The signed string sent from the user to that VASP is intended for use by the VASP as evidence to obtain payment for service provided to the user. If this signed string does not explicitly contain anything which identifies the VASP then, at least in theory, a bad VASP could take the signed string together with tick values released by the user, and claim payment for the service actually provided by another VASP.

Such an attack, which we refer to as a *Bill theft* attack, is prevented by including in the signed string an identifier for the VASP who is actually providing the service

7.2.5.2.7 Payment scheme attacks

The term *Payment scheme attack* is intended to refer to any attack on the tick payment scheme itself (Section 7.2.4.3.3). The payment scheme involves the user generating and signing a commitment α_T, where $\alpha_T = F^T(\alpha_0)$, and F is a one-way function. In fact, the function *F* is chosen from a family of such functions, with the choice of function being specified by an Initialisation Vector *IV*.

If one member of the family of functions proves to be insecure, then there is the possibility that the VASP could claim payment for services not provided. That is, suppose the VASP has found a weak function *F** from the family of possible functions. Suppose also that the user has signed the commitment α_T using a different (secure) function *F*. The VASP now finds a value α^* which satisfies $\alpha_T = (F^*)^T(\alpha^*)$, this being possible because of the insecurity of *F**. The VASP, when claiming payment, produces the signed commitment and α^*, and claims payment for the full *T* 'ticks' of service.

Such an attack is ruled by the inclusion of the initialisation value *IV* in the string signed by the user. This removes any possible ambiguity about the choice of the function *F*. For a more detailed discussion of how to use and design one-way functions in such a protocol, see [KMP98].

7.2.5.2.8 Content verification attack

This attack is very similar to a signer verification attack except that it is applicable to situations where a signature has been intercepted, the signer is known, the message being signed is almost known, and there are a limited number of possibilities for the remainder of the message being signed. In such a case the interceptor can repeatedly verify the signature by guessing the unknown part of the message, hashing it with the known part of the message, and comparing the result with the hash of the actual message obtained by verifying the signature using the known public verification key. Such an attack is applicable to the signature of the TTP sent in the third and fourth messages, where the only unknown variable is *cidU*, a value which could be guessed from a limited number of possibilities and knowledge of which almost certainly reveals the identity of U. One way of countering this attack is to encrypt the signature in the fourth message using a key that is available to both the user and the VASP, which is why this is done in the protocol described in Section 7.2.4.2.3.

7.2.5.2.9 Bad VASP replay attack

In this attack a bad VASP V′ observes a communication session between U and V and then attempts to determine the identity of U by replaying the opening message from the previous communication session to the TTP. The TTP responds to V′ quite legitimately, but because of the inclusion of $\{id_U\}_L$ in the replayed message, it assumes that V′ is acting on behalf of U. The TTP will respond by sending V′ a certificate (chain) for the public key of U and so the identity of U is revealed to V′.

It is not clear how practical a bad VASP replay attack is, and the protocol variant described in Section 7.2.4.2.3 does not protect against such an attack. It is however straightforward the protocol to prevent this sort of attack from occurring. One method of avoiding this attack is to use encryption to make sure that the certificate (chain) for the public key of U is not revealed to any VASP until that VASP has been authenticated. A variant of the ASPeCT protocol that does precisely this was published in [HP98]. Another possibility is to bind the identity of V to the identity of U in the opening protocol message. The most appropriate way of doing this is by sending a MAC on the joint identities. The TTP is then faced with the extra task of checking this MAC before proceeding with the protocol.

7.2.5.2.10 Oracle attacks

Oracle attacks are a very general class of attacks on cryptographic protocols. The basic idea of such an attack is for a cryptanalyst to engage in a protocol with the intention of making the other party reveal the results of a cryptographic calculation of potential value to the cryptanalyst. Examples of such attacks include situations where an entity is persuaded to release a signature on data chosen by a third party. The party persuaded to perform a calculation and release the result is said to be made to 'behave as an oracle'.

Again it is not clear exactly how practical an oracle attack is in the UMTS environment, but we note that previously published variants of the protocol [HHM98], [MPM98] are potentially subject to a special type of Oracle attack, where a VASP makes the TTP *T* behave as an Oracle. The problem arises when, in the second message, the VASP sends the TTP a value g^u, and the TTP responds in the third message with a collection of data items including the key *L*, equal to g^{uw} (where *w* is a long-term secret of the TTP). Thus, the TTP can potentially be persuaded to compute and release the value x^w, for any value *x* chosen by the VASP.

To see why this could be a problem, suppose the VASP performs this attack a large number of times, letting *x* range through all the primes less than *B*, for some positive integer *B*, and stores all the results. The 'bad' VASP can now trivially compute y^w for all *B*-smooth integers *y*. For a definition of *B*-smooth see, for example p.92 [MOV97] (essentially, a number is *B*-smooth if and only if all its prime factors are smaller than or equal to *B*). Of course, if *y* is randomly chosen and sufficiently large, then the chances that it is *B*-smooth are very small. The bad VASP can improve his chances a little by successively testing *y*, *y*+*p*, *y*+2*p*, *y*+3*p*, ..., *y*+*tp* for smoothness, where *p* is the Diffie-Hellman modulus in use, although his chances of success are probably rather small given *p* is sufficiently large. It might be interesting to try and find a strategy which the bad VASP could follow to find a *t* such that *y*+*tp* is *B*-smooth.

In any event, this problem is avoided in the protocol variant described in Section 7.2.4.2.3.

7.2.5.3 Alternative protocols

There have been a large number of different proposals for possible authentication protocols suitable for mobile computing and telecommunications environments. We provide a list of other protocols identified in the literature in Section 7.2.5.3.1, and provide here a few comments on just a few of these many proposals. We note that alternatives to the payment protocol were briefly mentioned in Section 7.2.3.

7.2.5.3.1 List of mobile computing authentication protocols

The following is a list of references relating to mobile authentication protocols. Note that this list does not include documents relating to the ASPeCT project (see main reference list) and that not all the following references actually propose a protocol. In the subsequent discussion of alternative protocols, references relate to papers on this list.

[AD94]	A. Aziz and W. Diffie, Privacy and authentication for wireless Local Area Networks. *IEEE Personal Communications,* 1(1) (1994) 25-31.
[AS93]	R. Akiyama and S. Sasaki. Authentication and encryption in a mobile communication system. *Proceedings of 43rd IEEE Vehicular Technology Conference,* Secaucus NJ, 1993.
[ASK98]	M. Aydos, B. Sunar and C.K. Koc. An elliptic curve cryptography based authentication and key agreement protocol for wireless communication. *Proceedings of 2nd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications,* (1998).
[BCY91]	M.J. Beller, L.-F. Chang and Y. Yacobi. Privacy and authentication on a portable communications system. *Proceedings of GLOBECOM '91.*
[BCY93]	M.J. Beller, L.-F. Chang and Y. Yacobi. Privacy and authentication on a portable communications system. *IEEE Journal on Selected Areas in Communications*, 11 (1993), 821-829.
[BY93a]	M.J. Beller and Y. Yacobi. Fully-fledged two-way public key authentication and key agreement for low-cost terminals. *Electronics Letters* 29 (1993) 999-1001.
[BY93b]	M.J. Beller, Y. Yacobi: "Authentication and key agreement protocol for PCS", Joint experts meeting on privacy and authentication for PCS, P&A JEM/93-012, Nov 8, 1993.
[BY94]	M.J. Beller and Y. Yacobi.. Minimal asymmetric authentication and key agreement schemes. October 1994, unpublished manuscript.
[B95]	D. Brown. Techniques for privacy and authentication in Personal Communication Systems. *IEEE Personal Communications,* August 1995.
[C94]	U. Carlsen. Optimal privacy and authentication on a portable communications system. *ACM Operating Systems Review*, 28 (1994), 16-23.
[CFT98]	C. Carroll, Y. Frankel, Y. Tsiounis, Efficient key distribution for slow computing devices, *Security & Privacy*, Oakland, 1998.
[DOW92]	W. Diffie, P.C. van Oorschot and M.J. Wiener. Authentication and authenticated key exchanges. *Designs Codes and Cryptography*, 2 (1992) 107-125.
[ETSI92]	ETSI ETS 300175-7. DECT Common Interface, Part 7: Security Features. October 1992
[ETSI94]	ETSI ETS GSM 03.20. European Digital Cellular Telecommunications System (Phase 2); Security-related network functions. Version 4.2.4, September 1994.
[ETSI95a]	ETSI SMG SG DOC 73/95. A public key based protocol for UMTS providing mutual authentication and key agreement, September 1995.
[ETSI95b]	ETSI SMG SG TD36/95. P. De Rooij. Public key authentication for UMTS.
[ETSI96]	ETSI SMG SG TD80/96. DeTeMobil. A new authentication mechanisms for UMTS.
[FHK95]	Y. Frankel, A. Herzberg, P.A. Karger, H- Krawczyk, Ch. A. Kunzinger, M. Yung: Security Issues in a CDPD Wireless Network. *IEEE Personal Communications*, August 1995.
[LH93]	H.-Y. Lin and L. Harn. Authentication in wireless communications. *Proceedings of GLOBECOM '93.*
[LH96]	H.-Y. Lin and L. Harn. Authentication protocols for personal communications systems. *Computer Communication Review*, 25(4) (1996) 256-261.
[MC99]	C.J. Mitchell and L. Chen. "Security in future mobile multimedia networks", in *Insights into Mobile Multimedia Communication*, Academic Press, 1999.

[M96]	S. Mohan, Privacy and Authentication protocols for PCS, *IEEE Personal Communications*, Oct 1996.
[MST94a]	R. Molva, D. Samfat and G. Tsudik. An authentication protocol for mobile users. *Colloquium on Security and Cryptography Applications to Radio Systems*, (1994) 62-??.
[MST94b]	R. Molva, D. Samfat and G. Tsudik. Authentication of mobile users. *IEEE Network*, 8(2) (1994) 26-34.
[MV96]	Y. Mu and V. Varadharajan. One the design of security protocols for mobile communications. *Information security and privacy*, Lecture Notes in Computer Science 1172 (1996) 134-145.
[P97a]	C.-S. Park, On certificate-based security protocols for wireless mobile communication systems, *IEEE Network*, September/October 1997, 50-55.
[P97b]	S. Patel, Weakness of North American Wireless Authentication protocol, *IEEE Personal Communications*, June 1997.
[P97c]	S. Pütz, Zur Sicherheit digitaler Mobilfunksysteme, *Datenschutz und Datensicherheit* 21 (1997) 6, 321-327.
[PKO93]	C. Park, K. Kurosawa, T. Okamoto and S. Tsujii. On key distribution and authentication in mobile radio networks. Presented at *EUROCRYPT '93 rump session.*
[SM94]	D. Samfat and R. Molva. A method providing identity privacy to mobile users during authentication. *Proceedings of Workshop on Mobile Computing Systems and Applications*, (1994) 196-199.
[SMA95]	D. Samfat, R. Molva and N. Asokan. Untraceability in mobile networks. Proceedings of MobiCom '95 (1995).
[VM96]	V. Varadharajan and Y. Mu. Authentication of mobile communications systems. Proceedings of IFIP World Conference on Mobile Communications (1996).
[W95]	J. Wilkes. Privacy and authentication needs of a PCS. *IEEE Personal Communications*, August 1995.
[YOL98]	X. Yi, E. Okamoto and K.Y. Lam. An optimized protocol for mobile network authentication and security. *Mobile Computing and Communications Review*, 2(3) (1998) 37-39.
[Z96]	Y. Zheng. An authentication and security protocol for mobile computing. Proceedings of IFIP World Conference on Mobile Communications (1996) 249-257.
[ZL98]	J.Zhou and K.-Y.Lam. Undeniable Billing in Mobile Communication, *Proceedings of MobiCom 98.*

7.2.5.3.2 Comparison of selected protocols

First we note that the bulk of the protocols in Section 7.2.5.3.1 are exclusively authentication protocols and do not specifically initialise a payment mechanism. We assume that to incorporate a payment mechanism of the type used in ASPeCT it will be necessary for the user to compute at least one digital signature when committing to the initialisation data. Hence any protocol which does not involve the user signing data will need to have such a computation added to the existing protocol. We also note that we are considering other protocols in the light of the requirements specified in Section 7.2.2and the goals of Section 7.2.4.2.1. The fact that a particular protocol fails these requirements does not suggest that it is flawed in itself, just that it is unsuitable for the UMTS scenario considered in ASPeCT.

We consider a number of protocols from Section 7.2.5.3.1.

7.2.5.3.2.1 The Park protocol

The Park protocol was proposed in [P97a]. The Park protocol uses a simple Diffie-Hellman variant to establish a secret key between user and VASP. Unfortunately the Park protocol does not satisfactorily achieve several of the goals of Section 7.2.4.2.1, most notably mutual entity authentication. This is because an opponent who discovers a previously used session key can easily masquerade as the VASP in a subsequent communication run. Details of this weakness can be found in [HMM98].

7.2.5.3.2.2 BCY related protocols

A number of protocols have been published that are all based on the [BCY93]. These include [C94] and [MV96]. Again Diffie-Hellman techniques are used to establish the common session key, but the Modular Square Root public key system [W80] is used to create certificates. It is fairly straightforward to extend such a protocol (that of [MV96] for example) to include the extra signature and verification needed to initialise a payment mechanism. This establishes a protocol that appears to meet all the goals of Section 7.2.4.2.1. The disadvantage is that such an extended protocol results in the VASP having to conduct an extra (on-line) verification in comparison with ASPeCT (see [HMM98]).

7.2.5.3.2.3 STS related protocols

The Station-to-station protocol (see for example [DOW92]) is a well established authentication mechanism which appears to satisfy the authentication goals of Section 7.2.4.2.1 and is easily extended to include an initialisation of payment mechanism. Unfortunately the STS protocol was not explicitly designed for entities with asymmetric computing capabilities, and the user has to compute one extra on-line verification with respect to ASPeCT, and the VASP at least one extra signature. Details can again be found in [HMM98].

7.2.5.3.2.4 Aziz-Diffie protocol

The Aziz-Diffie protocol [AD94] has similar overhead requirements to the STS protocol and fails to meet the goals of mutual key confirmation and user confidentiality over the air interface. This protocol has the interesting property that the mutually agreed key is not established by a Diffie-Hellman variant. Further discussion can be found in [HMM98].

7.2.5.3.2.5 The ZL protocol

The ZL protocol [ZL98] is of particular interest, because like ASPeCT it is designed to initialise a payment mechanism which is based on micropayments using one-way chains. Although the background is similar, and the user overhead is very low (lower than in ASPeCT), there are a number of features that make the ZL protocol unsuitable for the ASPeCT UMTS environment. Most notably, the goals of the ZL protocol are less ambitious than the goals of the ASPeCT protocol. The ZL protocol thus does not have several of the ASPeCT security features, including mutual entity authentication, a feature deemed critical by ASPeCT. The ZL protocol also only offers one way implicit key authentication, and one-way key confirmation. The prerequisites of the ZL protocol are very different, and most notably the need for the VASP and the home TTP of the user to share a secret key seems most unsuitable to the ASPeCT scenario, which is based entirely on the use of public key cryptography for key management purposes. An interesting feature of the ZL protocol is the de-coupling of the service request initialisation from registration in the ZL protocol. There are certain situations where this could be an advantage.

7.2.5.3.2.6 Other protocols

In the course of the ASPeCT work a number of protocols have been studied that are proposed for mobile computing environments but appear to fall well short of the goals that ASPeCT proposes necessary for mutual authentication purposes. These include the protocols of [ASK98] and [YOL98]. There are two main reasons why most of these protocols do not satisfy ASPeCT's requirements:

1. The goals of the protocols are not clearly defined (and hence achieved) or the goals are less ambitious than those defined for ASPeCT in Section 7.2.4.2.1.
2. The protocols are not designed to initialise a payment mechanism and thus lose any proposed efficiency advantages when forced to incorporate this extra feature.

Thus, while further analysis of the ASPeCT protocols is both welcome and necessary (see remarks in Section 7.2.5.1) it is felt that the ASPeCT solution is a satisfactory proposal and compares favourably with existing alternative proposals.

7.3 Demonstrations and trials of secure billing for value added information services

7.3.1 Overview

The results of the ASPeCT implementation work on trusted third parties and secure billing were presented in three steps:

- two first demonstrators (demo 1);
- one joint second demonstrator (demo 2);
- a field trial in which the ASPeCT second demonstrator was tested by a group of external users;

Two separate first demonstrators for trusted third party (TTP) services and for secure billing services were completed in February 1997. They were documented in deliverables [D09] and [D10] respectively. Enhanced versions of the demonstrators were shown at the conference IS&N 97 in Como, May 1997.

The first demonstrators were combined in a joint TTP and secure billing second demonstrator in Quarter 1, 1998. The purpose of combining the first demonstrators was to show that the entities involved in secure billing can make on-line use of TTP services. The joint second demonstrator was shown at the conference IS&N 98 in Antwerp, April 1998.

The purpose of the 'TTP and secure billing field trial' was to show that the system works when used by real users, and to measure and analyse the technical feasibility of the system in absolute terms and the acceptability of system performance and quality of service as perceived by users, and to obtain feedback from users regarding the functionality of the application.

7.3.2 First demonstrator

7.3.2.1 Description of demonstration

7.3.2.1.1 Introduction

The value-added service (VAS) used in the demonstration is a service whereby a UMTS user can retrieve information from a remote server. The value-added service provider (VASP) application is based on Web infrastructure, where the remote information server is a Web server, and the user has a Web client, which can be used to browse information on the remote server.

In a typical scenario the UMTS user connects to the Web server on his UMTS terminal. He then uses his Web client to access information on the VASP's Web server. The Web client presents the UMTS user with hypertext information which he may wish to browse on-line, or save locally for viewing off-line. Secure billing is based on the amount of data transferred to the user over the connection and the value of the information transferred as specified by an agreed charging rate. A secure payment protocol provides an incontestable charging service whereby neither the UMTS user nor the VASP can later deny that a specified amount of data was transferred at a specified charging rate.

In a demonstration exhibited at the IS&N '97 conference in Como, the ASPeCT project showed how mobile users can pay for access to information services in a flexible, efficient and secure way. The method has potential application to charging for any telecommunications service.

7.3.2.1.2 Description of the demonstration from the observer's point of view

7.3.2.1.2.1 Authentication and initialisation of payment

After each entity has been initialised with personalised security information, the VASP can offer a secure billing service to users who wish to browse its Web server. To initiate secure billing the user will execute

the authentication and initialisation of payment protocol with the VASP, as described in section 7.3 above. This execution is triggered by using the Web browser, there is no need for the user to execute it explicitly.

Each entity first has to obtain a certificate of the other entity's public key. Variant B of the authentication and initialisation of payment protocol is executed if certificates have not been exchanged previously, whereas Variant A is executed if the user and VASP have obtained certificates in a previous run of Variant B.

Part of the authentication and initialisation of payment protocol involves the VASP supplying the variable *ch_data*, which is used to specify the charging rate for the subsequent payment protocol.

Optionally, the value of *ch_data* may be displayed to the user in a Window of the Graphical User Interface, and the execution of the protocol is only continued when the user manually confirms that he agrees with the tariff given by *ch_data*. This feature can be switched off by setting the corresponding security parameter.

The result of the successful execution of the authentication and initialisation of payment protocol is mutual authentication between the user and the VASP, an agreement of a session key and an initialisation of the tick payment scheme.

The session key may be used for end-to-end confidentiality or integrity services. However, only the key management for these services is demonstrated in the first demonstrator.

Error conditions are flagged to the user as appropriate via the GUI of the secure billing application.

7.3.2.1.2.2 Data transfer and charging

After the authentication and initialisation of payment protocol has been completed, the VASP commences the transfer of the appropriate HTTP response message associated with the first HTTP request. During the data transfer the VASP continually asks the user to release a payment parameter for a number of payment ticks. Providing the user responds appropriately, the data transfer continues. The secure billing application can suspend data transfer if a commitment to pay was not received.

7.3.2.1.2.3 Re-initialisation of payment protocol

If the total number of ticks required in the payment protocol exceeds the value of the system parameter *T*, then a commitment to a new series of tick payments has to be given. In this case the re-initialisation of payment protocol is used.

The protocol offers the possibility to agree a new charging rate between the user and the VASP and will commit the user to another series of tick payments.

7.3.2.1.2.4 Tracing

Throughout all interactions, the observer is able to monitor the message flows using the tracer. The tracer provides a detailed analysis of the execution of the protocol and allows recording of the protocol messages, so they can be viewed off-line. The tracer is an optional feature which can be turned on for demonstration and debugging purposes.

7.3.2.1.2.5 The secure billing application GUI

The user or VASP interacts with the secure billing application via its GUI in order to supply and receive management information relating to the service being offered. For example, the UMTS user can get accurate and reliable information about how much he has been charged for the service received so far. In this respect the GUI supplies the user (and VASP) with evidence generated by the payment protocol which confirms the accuracy of the charging information.

At the end of the call, the GUI presents the user or VASP with a summary of the relevant data collected during the call.

7.3.2.2 Architecture

7.3.2.2.1 *Protocol architecture*

Regarding the protocol architecture, it is particularly worth emphasising that our concept permits the use of existing applications (WorldWideWeb client and server) as well as of existing communication stacks (TCP/IP). Most applications useful in our context are based on a standardised interface, namely sockets, or, in a Windows environment, more specifically Windows sockets. There are three protocol layers (cf. Figure below): An application layer, a communication layer and a security layer in between, realizing the security protocols as described in section 7. The security layer provides Windows sockets to the application layer. There is no need to modify the application and there is no need for a security interface extending Windows sockets. The security layer uses Windows Sockets to access the communication stack. In this way, the security layer is independent from particular applications and is transparent for the application. This is seen as a major advantage of our approach.

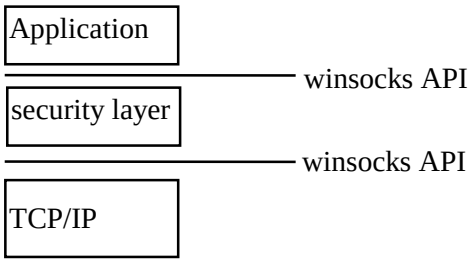


Figure 7.4 - Protocol stack

At the highest level, the application layer implements the functionality of the Web server on the VASP's site, and of the Web client on the user's site. The server and client communicate at the application level using the hypertext transfer protocol (HTTP). In practice the Web client requests information by specifying a uniform resource locator (URL), which uniquely identifies the file to send.

At the lowest level, the transport system layer is implemented as a TCP/IP stack. The Web server and client communicate at the transport system level using the transmission control protocol (TCP). The transport system layer provides a DLL which provides a Winsocks interface to the layer directly above it.

The ASPeCT security layer exists between the application layer and the transport system layer. As such, it must have a Winsocks interface at the upper and lower levels. The Web server and client communicate at the security control level using ASPeCT-developed secure billing protocols.

The ASPeCT security layer additionally interfaces with separate software modules which implement the secure billing demonstrator.

7.3.2.2.2 Architecture of the ASPeCT security layer

The ASPeCT security layer consists of the following principle modules:

The finite state machines (FSMs) module

This module implements the secure billing protocols as specified in section 7.3.

The security control module

The security control module analyses Windows sockets calls from the Web application in order to trigger certain events in the secure billing protocols. It also stops transmission of further messages from the server if payment is not duly received from the user.

The cryptographic functions module

It consists mainly of the Siemens cryptographic library ACRYL which provides a uniform Application Programming Interface to access the cryptographic functions.

The tracer module

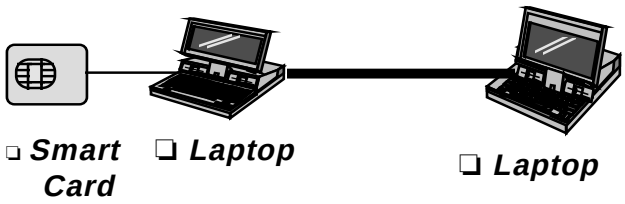
This module traces the events of other modules (if specified there) and displays them in a window of the GUI.

The secure billing application Graphical User Interface (GUI).

It consists of two parts. The first part displays the tracer messages, the second part provides an interface to manage the security parameters of the system.

7.3.2.2.3 Hardware configuration

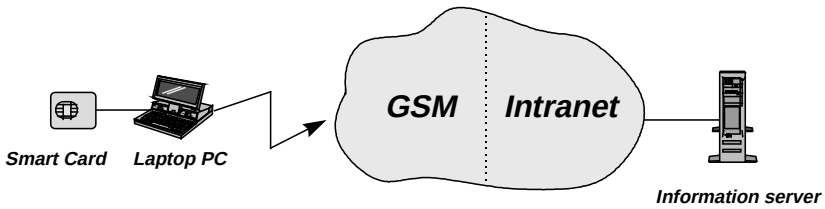
Both versions of the first demonstration are based on PCs, a laptop PC representing the UMTS user and a desktop or laptop PC representing the VAS provider. The user's smart card is attached via a card reader to the user's laptop PC. The two versions differ in the way in which connectivity is provided between the user's laptop PC and the VAS provider's PC. In the first version, completed in February 1997, connectivity is provided by an Ethernet link. This is depicted in the following figure:



Mobile User Value Added Service Provider

Figure 7.5 - Physical configuration of version 1

In the second version, presented at the IS&N97 in May 1997, connectivity is provided by GSM data connections. This is depicted in the following figure:



Mobile User Value Added Service Provider

Figure 7.6 - Physical configuration of version 2

The network operator is not represented in the demonstration set-up.

When the demonstration is run over a GSM connection (version 2), the smart card attached to the PC is different from the SIM card needed to operate the GSM mobile phone. In the future, it is envisaged that - together with the advent of mobile communicators with integrated mobile terminal functionality - there will be multi-application smart cards which integrate SIM or UIM functionality with e. g. payment functions such as the ones needed for secure billing.

The smart card chip is the Siemens SLE44CR80CS cryptocontroller which features a crypto co-processor for modular arithmetics which is needed to carry out public-key based cryptographic mechanisms efficiently.

The card reader is attached to the laptop via the PCMCIA slot.

7.3.2.2.4 Software configuration

The Windows 3.11 operating system is used on both demonstration laptop PCs.

The smart card chip runs a proprietary Siemens card operating system.

A TCP/IP based communications sub-system provides a means to interface the two PCs. The communications sub-system provides a network programming interface based on the Berkeley Sockets paradigm. This interface is the Windows sockets 1.1 application programming interface. The applications on the user and VASP side are compatible with this interface.

The Windows Web server HTTPd V1.4c for Windows 3.1, is used on the VASP laptop PC.

The Web client residing on the user laptop PC is Microsoft Internet Explorer in the version 2.01.

The demonstration software was written in ANSI C.

7.3.3 Second demonstrator

7.3.3.1 Description of demonstration

The joint TTP and secure billing demonstrator (second demonstrator) realises the same application as the first demonstrator, access to mobile value added services, using WorldWideWeb technology, and on-line payment for these services. Hence the corresponding description in the section on the first demonstrator also applies here. Secure billing protocols developed by ASPeCT for this purpose are used between the user and the VASP. The main difference between the second demonstrator, completed in the first quarter of 1998, and the first secure billing demonstrator, completed in 1997, is that the second demonstrator makes use of on-line Trusted Third Party services, and hence that a third entity is involved on-line in the demonstration. More specifically, the Value Added Service provider (VASP) contacts a trusted third party (TTP) on-line during the session establishment with the user and retrieves information to certify that the user and the VASP are both entitled to use the underlying payment system.

Another noticeable difference is the enhancement of the Graphical User Interface.

To show relevant information to observers of the demonstration a common tool was developed, the character oriented tracer.

7.3.3.1.1 Description of the demonstration from the observer's point of view

The on-line use of TTP services is transparent to the user. Hence, the main difference he will notice is the enhanced GUI. Otherwise, the demonstration from an observer's point of view is very similar to what was described in the corresponding section for the first demonstrator. We therefore do not repeat this description here. A more detailed description of a run through the second demonstrator when used in the field trial is given below in the section on the field trial.

7.3.3.2 Protocol architecture and software configuration

7.3.3.2.1 Global structure

The on-line information service used in the demonstrator is a WorldWideWeb (WWW) service. On the user PC a WWW browser is installed, exchanging html documents via the http protocol with the WWW server on the VASP PC. Any WWW browser may be used. In the version used in the field trial, the browser was MOSAIC. In the latest version of the demonstrator, the browser is Netscape communicator version 3.01. The server is W4-Server version 2.6a. On the user PC a local catalogue of the information available from the server is installed. The user opens the local file in the browser and chooses from the list shown there. The server has all the information offered in the user's local catalogue available on disk.

The PCs all run the Windows NT 4.0 operating system.

The smart card chip runs a proprietary Siemens card operating system.

The demonstration software was written in ANSI C.

The global structure of the system is shown in the following Figure 7.7:

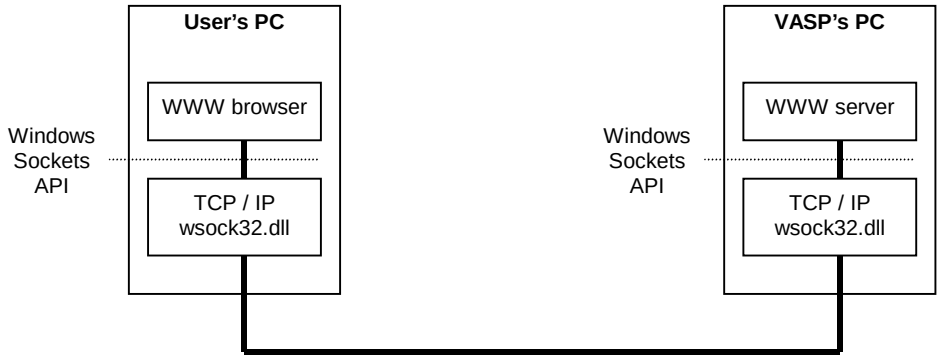


Figure 7.7 - Global structure of WWW service

The secure billing / TTP software consists of several parts:

- Intermediate layers in the communication stack of the user's WWW browser and the VASP's WWW server monitoring the communication between browser and server and reacting to specific events.
- Separate applications (called FSMs) on the user's PC and the VASP's PC handling the secure billing protocols triggered by the monitoring layers.
- Another FSM application running on the TTP's PC participating in the security protocols, contacted by the VASP's FSM.

7.3.3.2.2 The intermediate layers

The central part of the secure billing software are the intermediate layers embedded in the communication stack of the user's WWW browser and the VASP's WWW server. These layers reside between the WWW application and the TCP/IP stack. The figure below shows this structure:

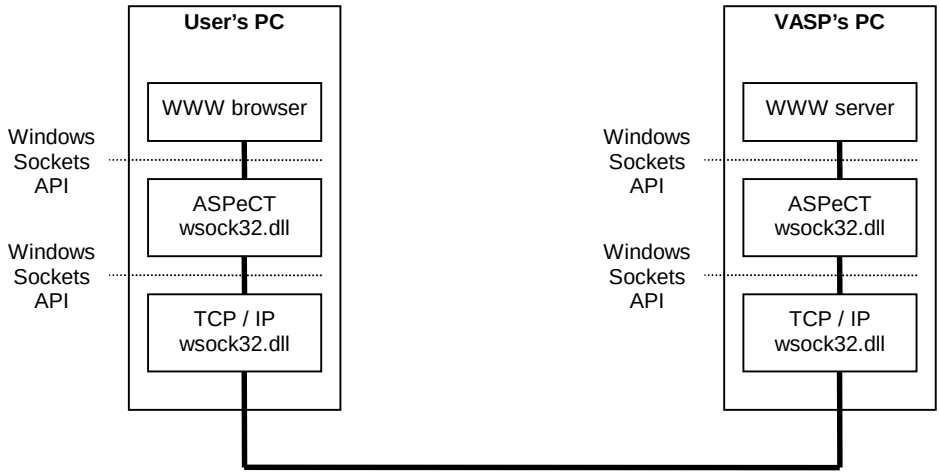


Figure 7.8 - Global structure with intermediate layers

These intermediate layers monitor all the communication between the browser and the server. A basic functionality offered by the ASPeCT `wsock32.dll` is tracing the communication between the browser and the server and displaying it with different levels of detail in the tracer. To realise the secure billing functionality the intermediate layers communicate with the FSMs which execute the secure billing protocols. This mechanism is shown in Figure 7.9

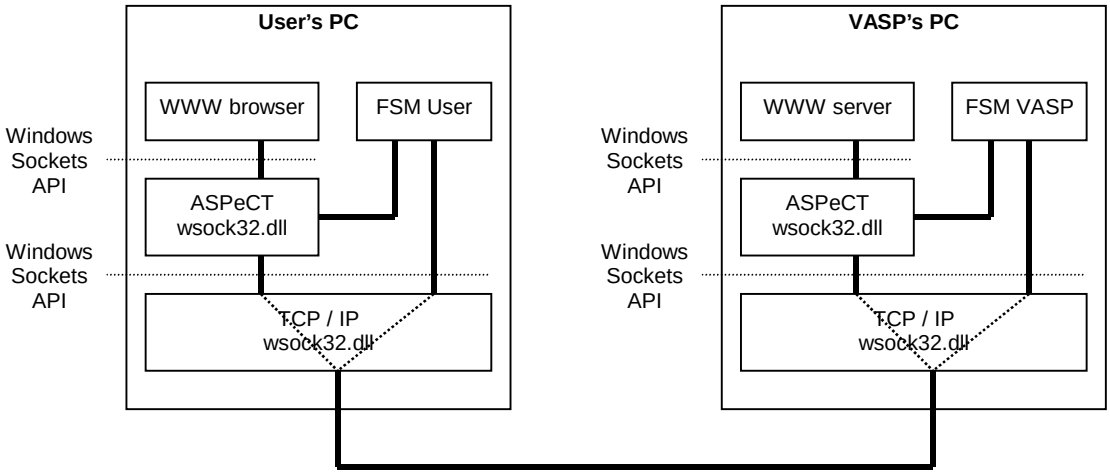


Figure 7.9 - Global structure with FSMs

7.3.3.2.3 The FSMs

The secure billing protocols are executed by applications of its own called finite state machines (FSMs). The FSMs on the user's PC and on the VASP's PC offer an interface to other applications based on an interprocess communication mechanism to send requests to the FSMs. In the demonstrator the intermediate layer on the user's PC uses the interface offered by the user's FSM and the intermediate layer on the VASP's PC uses the interface offered by the VASP's FSM. When a user connects to a VASP and there is no secure billing connection yet between them, the VASP's FSM connects to the TTP's FSM to get currently valid certificates for the user and the VASP. Some cryptographic functions (digital signatures) are executed on a smart card where also the cryptographic keys are stored. The smart card is accessed by the User's FSM.

The following Figure 7.10 shows the general structure of the demonstrator including the TTP. The smart card is not shown separately in the figure.

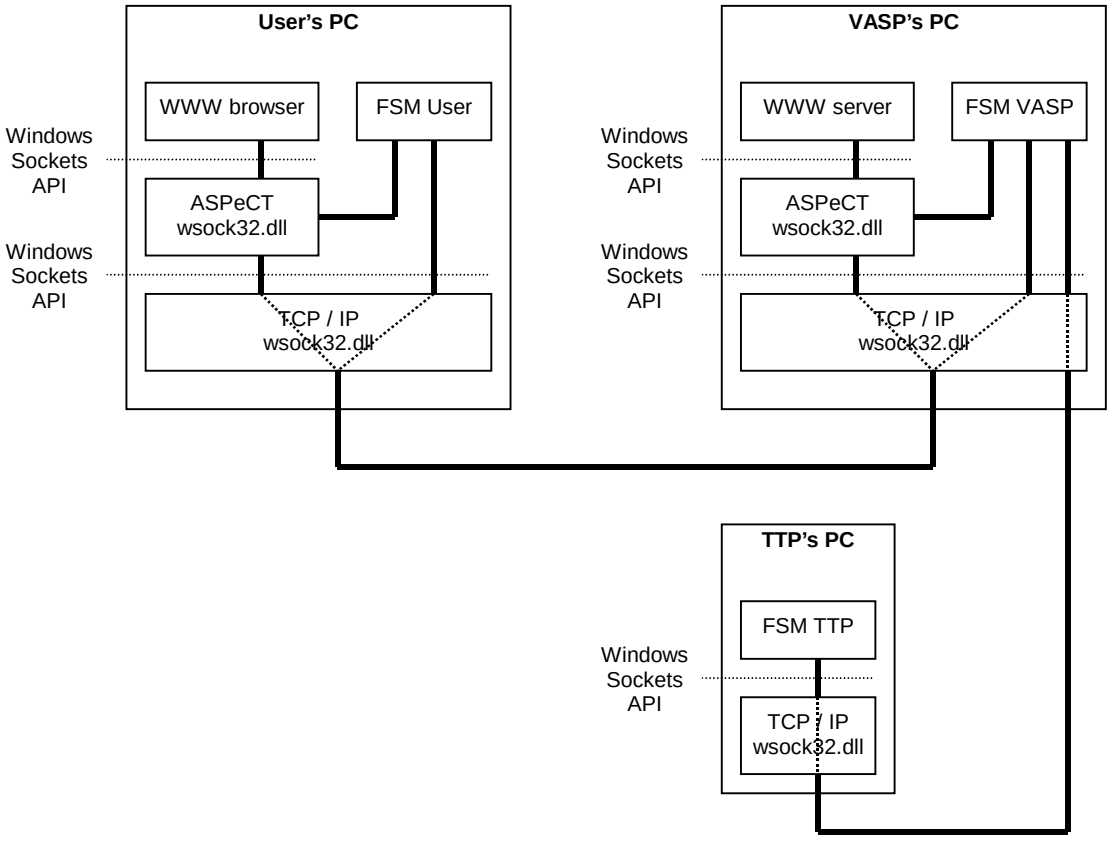


Figure 7.10 - Global structure with TTP

7.3.3.2.4 The tracer

To display all kinds of information to observers of the demonstration a tool was developed, the character-oriented tracer.

One tracer acts as a server for a group of processes, which can call the service of this tracer. Several tracers can run in parallel on one PC, each of them serving a separate group of clients. Messages sent by client processes are displayed in a chronological order, so the sequence of events in separate processes can be shown in one window even for events happening in real-time with very short delays.

In the typical demonstration scenario there is one tracer for the user's processes:

- The user's intermediate layer
- The user's FSM

There is one tracer for the VASP's processes:

- The VASP's intermediate layer
- The VASP's FSM

And there is one tracer for the TTP's processes:

- The TTP's FSM

7.3.3.2.5 Cryptographic library

The cryptographic library used by the joint demonstrator is ACRYL (Advanced CRYptographic Library) which is provided by Siemens AG.

7.3.3.2.6 Communication Interfaces

The communication between the FSMs is realised using TCP/IP via the Windows sockets API version 1.1 (user's FSM, VASP's FSM) and version 2.0 (TTP's FSM).

7.3.3.3 Hardware configuration

The second demonstration is based on three PCs, a laptop PC representing the UMTS user and desktop PCs representing the VAS provider and the TTP respectively, and a smart card. The user's smart card is attached via a card reader to the user's laptop PC. There are again two versions of the second demonstrator. As for the first demonstrator, the two versions differ in the way in which connectivity is provided between the user's laptop PC and the VAS provider's PC. In the first version, connectivity is provided by an Ethernet link. In the second version, connectivity is provided by GSM data connections. The first version is also the one used in the field trial. This version is shown in Figure 7.11. The network operator interacts only off-line and is not represented in the demonstration set-up.

When the demonstration is run over a GSM connection (version 2), the smart card attached to the PC is different from the SIM card needed to operate the GSM mobile phone. In the future, it is envisaged that - together with the advent of mobile communicators with integrated mobile terminal functionality - there will be multi-application smart cards which integrate SIM or UIM functionality with e. g. payment functions such as the ones needed for secure billing.

The smart card chip is the Siemens SLE44CR80S cryptocontroller which features a crypto co-processor for modular arithmetics which is needed to carry out public-key based cryptographic mechanisms efficiently.

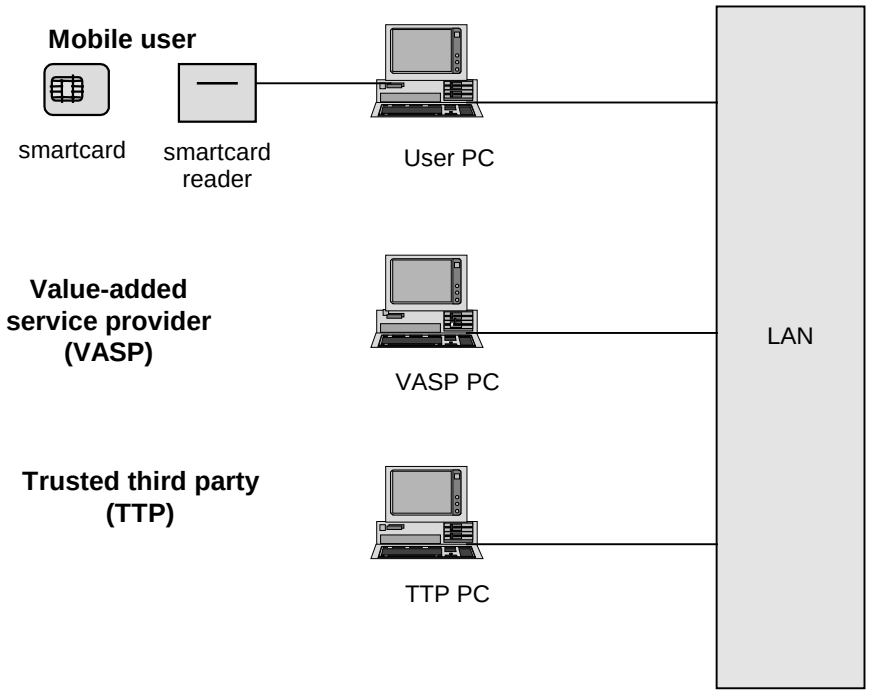


Figure 7.11 - Configuration of second demonstrator

7.3.4 Field trial

7.3.4.1 Introduction

In order to have the ASPeCT TTP and secure billing concepts which were implemented in the second demonstrator tested by real users, ASPeCT conducted a field trial. The field trial was named " Secure Billing for Information Services in Future Mobile Networks". The trial took place on the SwissCom premises in Basle, Switzerland, between 18 May 1998 and 22 May 1998.

The user group comprised a number of non-specialists representing "ordinary" future users of the system and a number of experts who, however, had not been involved in developing the demonstrator. The general

objective of the ‘TTP and secure billing field trial’ was to show that the system works when used by real users, and to measure and analyse the technical feasibility of the system in absolute terms and the acceptability of system performance and quality of service as perceived by users, and to obtain feedback from users regarding the functionality of the application. A secondary objective was to gain an impression of the awareness of users of the employed technology and its security implications, and of user acceptance of this technology in general. In this context it is important, however, to remember that the demonstrator used in the field trial is not a product. In particular, the user interface does not have the perfection of that of a product. The focus was more on demonstrating the fundamental concept and implementing the basic technology. Similarly, the field trial environment was not the real UMTS environment for which the trialled technology is intended; in particular the terminals, the network and the information offered were different.

The users were given documentation consisting of an introduction giving them the background of the trial and telling them what it was all about, a list of tasks they had to complete during the trial which guided them through the actual run of the trial, and two questionnaires, one on the users' background and one regarding the observations they made during the trial. The first questionnaire was to help the evaluators to weigh the responses from the users and group the responses according to previous experience with the relevant technology. The second questionnaire was to provide the results of the trial. There were two different versions of the list of tasks and the questionnaire, one for the normal users and one for the expert users. In addition, they were given a publication "Secure Billing for Information Services in Future Mobile Networks" and a one page summary "Secure Billing for Information Services using micropayments".

The questions in the second questionnaire were grouped under the following headlines:

- User friendliness
- Performance
- Security awareness
- Other

The complete documentation handed out to users in the field trial is contained in Annex A. A condensed version is contained in the following subsections.

7.3.4.2 Overview and background

This subsection gives an overview of the trial in much the same way in which the users were given an overview in the user documentation:

The technology tested in the field trials is at the confluence of three tributary technologies that have experienced dynamic growth in recent times, namely access to multimedia information via the World Wide Web, electronic payment systems and mobile radio.

Most people are familiar with the World Wide Web these days and have experience with browsers. In the field trial the “Mosaic” browser was used. The field trials did not give access to the world-wide Internet, just to a selected server that was also linked to the test network. The technology used is the same however.

At present, most of the information one can download via the World Wide Web is free of charge. In future, many information providers will want to earn money from their information. From the technical point of view, this can only happen if suitable payment systems are available that offer efficient and secure payment. Since the product is in electronic form it seems obvious to make payments in electronic form. It is often the case that an individual “item” of information, such as a newspaper article, is only of small value but that a large number of such “items” are downloaded so that their cost mounts up. The costs arising from a single transaction must also be very small. There have been lengthy discussions in recent years on which types of payment system would be suitable for such small payments. They are generally referred to as “micropayments”. In the field trial a micropayments system that goes by the name of “Tick Payment System” was used. This system takes its somewhat unusual name from the ticking of the charge meters used originally for billing telephone calls. Ticks are simply a kind of play money but they do have a fixed conversion rate into hard cash.

Obviously one of the most important factors for any payment system, whether electronic or not, is that it is secure against any form of misuse. This security is provided by appropriate cryptographic methods. Again, it is worth noting that the time, effort and resources invested in security should be in relation to the value of the objects to be protected. One aspect of security that is gaining in importance is the incontestability of the billing. This means that during the payment procedure data should be generated that can be used at a later date, not only by the parties involved but also by a neutral third party, as a basis for deciding whether a service has actually been used or whether the billing constitutes an unjustified request for payment. The method used in the field trial meets this requirement by making use of digital signatures.

In the field trial the users were sitting at PCs connected via a cable to a network giving access to the server on which the information was stored. This is far and away the most usual configuration at present for accessing the World Wide Web. Occasionally one will see laptop users logging on to a mobile network via a data modem, typically a GSM network. This configuration is only of limited use for downloading web pages however. Not only is the method unwieldy, it takes a long time to download information because the bit rate of current mobile systems is relatively low. In the not too distant future these restrictions will no longer apply. Hand-held or portable terminals will be available that combine the functionality of a laptop PC and adequate size screen with a mobile phone and data modem. UMTS will offer considerably greater bandwidth than current systems can offer and therefore drastically reduces the transmission time for information.

If users in a telecommunications network can transfer not only speech or data but also information content that are of special value to them, we talk about a value-added service. A contract with a service provider for a GSM-based mobile network will generally include a whole range of value-added services. Some of these services are listed here by way of illustration: breakdown service, travel service, travel information service, medical assistance away from home, secretarial service, and stock exchange information service. There are many more besides. In future there will be value-added services with special benefits in the mobile environment (particularly when linked to location services). A good example would be a European travel service for drivers. This service could inform drivers of the nearest hotel, restaurant or (open) garage, depending on the position determined by the location service, the date and the time of day. The route would be shown in the form of a map transmitted to the user’s terminal via the mobile system. The computer used in the field trial for storing the information ready for downloading is called the VASP (Value Added Service Provider) computer.

7.3.4.3 Trial configuration

The configuration used for the trial is that described for the first version of the second demonstrator - see subsection 7.3.3.3. The structure is depicted in the corresponding figure in that subsection. It is not repeated here.

It was decided to use wired terminals in the trial, because the alternative - using GSM data services - would have meant quite low data rates and would have proved very unsatisfactory for users browsing in a Web-like environment. This unsatisfactory aspect of the trial was expected to be a major determinant in the overall evaluation of the trial, at least for non-expert users. As the technology developed by ASPeCT has nothing to do with the provision of sufficient bandwidth, it was decided to go for an interconnection by LAN, so that the focus of the users would be on the ASPeCT developed features.

7.3.4.4 Trial scenario

Summarised below are the tasks users had to perform in the course of the field trial. These tasks were listed in detail in a separate document - see Annex A. The users were asked to keep to that list during the trials and to fill in a questionnaire at the end of the trials to provide feedback.

The users were handed a smart card by the field trial leader. This card acted as a security card and contained fundamental cryptographic keys and functions without which payments could not be made. In real life these cards would be received either from the UMTS service provider or from a trusted third party. The users were then asked to insert the smart card in the smart card reader.

The users found a PC (referred to as the “user PC”) running the Windows NT 4.0 operating system. The field trial leader had already started the PC up. The users now had to start the “Mosaic” browser by double clicking on its icon on the start-up screen.

The next screen showed a link to information provided for the purposes of the field trials. To make the information interesting and entertaining for ordinary users, tourist information (words and pictures) was included on the country of Jordan, extracts from Jordanian newspapers and hotel information. In addition - for those interested in security - FAQs (Frequently Asked Questions) on cryptography were accessible. The documents varied considerably in length and therefore in price. The payment method implemented for the field trials was such that one had to pay a certain number of ticks for a certain amount of bytes. The price of a document was therefore directly proportional to its volume. (The start page cost nothing.) This method is certainly not flexible enough for general use so it will have to be supplemented in a future version by a tariff structure based on the content and/or location (url) of the information.

On the following screen, a window with the name “Payment Status” was displayed. This window indicated the server the user was connected to, the number of bytes he had already received and the number of “ticks” due.

Expert users had to start the tracer by clicking once on the “Tracer User” icon along the bottom edge. The tracer enables them to track the communication protocol step by step - cf. subsection 7.3.3.2.4.

All users were then asked to click once on one of the links on the Mosaic start page. Each of the links related to the computer on which the relevant information was stored, the “VASP-PC”. Users could now see a dialogue box displaying the proposed tariff (in the example 1 tick for every 50 bytes). Users were then asked to agree to the proposed tariff. Only then, the program continued.

The web page downloaded from the VASP-PC then told users what other information was stored there. Users were asked to go from one link to the next in the usual way and view the information of interest to them.

During the session users could return to the home page (Mosaic start page) at any time.

Users were then asked to quit Mosaic. They saw the start screen again.

They now had to open the “Payment Manager” by double clicking on the relevant icon on the start screen. The Payment Manager shows which sessions the user had with which server. One can double click on any session to open a new window where details of the payments for that session can be viewed.

Finally users were asked to close the “Payment Manager” and remove the smart card.

Experts only were asked to perform a second run-through. This was for them to see whether a smart card was really needed to make the system work. Mosaic was launched again and the charge data were confirmed as usual. The expert users now had to try to continue the session without inserting their smart card in the reader. They learnt that they indeed had to insert their smart card.

Next a web page had to be downloaded as before. The smart card was then to be removed from the reader and the users was to try to download further information. He was asked to note the observations he made.

This completed the trial run-through for all users.

7.3.4.5 Graphical user interface

7.3.4.5.1 User side

On the client PC there are two ASPeCT applications with different graphical user interfaces:

- The secure billing application for viewing information on the current communication session (possibly including the display of detailed information in the character-orientated tracer). This information is stored in a file which can, for example, be used to check the correctness of bills that the VASP might send.

- A separate application for off-line viewing of information on previous communication sessions. It displays a listing showing all the VASPs the user has visited in the past, together with the accumulated payments for all the visits. The user then has the possibility of looking at a more detailed listing for a selected VASP showing the payment parameters for all sessions with that particular VASP. Furthermore, the user has the possibility to delete old information which is no longer of any interest to him.

7.3.4.5.2 VASP side

On the VASP's PC there is one ASPeCT application with a graphical user interface, namely the secure billing application. The GUI of the secure billing application on the VASP displays in its main window information about the current session with one particular user. This information is also stored in a file, such that detailed bills can be generated at a later date.

7.3.4.6 Trial users

7.3.4.6.1 Requirements on users' expertise:

Two different groups of users were sought, experts and non-experts. These two groups were loosely defined as follows:

A non-expert should be familiar with Windows-based PCs and with using a browser to download information from the World Wide Web. Everything else was explained in the user documentation. For an understanding of the technical background it would of course be useful, but not absolutely essential to have used a mobile phone, to know what a smart card is and perhaps have read or heard something about electronic payment systems.

An expert should also bring with him an understanding of the security aspects of the system on trial. He should be familiar with terms such as digital signature, authentication and certification. A knowledge of security problems relating to electronic payment systems would be an advantage. Experts were expected to familiarise themselves with at least the main features of the protocols used in the field trials. The enclosed publications went some way to providing this information.

7.3.4.6.2 User statistics

Of the 28 users originally selected for the trial, only 26 actually took part in the trial.

The 28 users where originally selected from four sources:

- SwissCom employees in Corporate technology department;
- members of the ASPA consortium (consisting of SwissCom and ASCOM employees operating the national host ATM platform in Basle);
- students from Oensingen Engineering School studying various engineering subjects;
- other.

The users were split into an expert group and a non expert group. Each group were given a different set of instructions for the trial and a different questionnaire to complete.

The expert group consisted of:

- 4 SwissCom technical employees;
- 1 ASPA technical employee.

The non-expert group consisted of:

- 1 SwissCom technical employee;
- 6 ASPA technical employees;
- 2 ASPA secretarial employees;

- 7 students on a telecoms programme;
- 2 students on an electronic data processing and telecoms programme;
- 1 student on an electronic data processing programme;
- 1 other student;
- 1 chemist.

Users were asked to complete a questionnaire on some general security related subjects. The responses to these questionnaires were used to build up a profile of each user group in order to help evaluate trial results. The results can be found in the section 7.4 on trial evaluation below.

7.3.4.7 Support

The support for the trial network (LAN, Windows NT PCs) was provided by ASPA, which has the responsibility of the operation of the 'Swiss National Host'.

The support for the application system (http server, http browser, secure billing software, TTP software, smart card hardware) was provided by Siemens AG. A hotline was provided during the trial.

7.3.4.8 Realisation vs. original plans

It was originally planned to base the TTP and secure billing trial on the experimental UMTS platform provided by the ACTS project EXODUS. The plans for the cooperation were laid down in a Memorandum of Understanding between the two projects, and were further specified in ASPeCT deliverable 19. Also, initial tests were carried out.

Unfortunately, the EXODUS UMTS platform could not be provided on time, so that ASPeCT had to go ahead with a fall-back solution based on communication over Local Area Networks.

What was lost through this deviation from the original plan?

It could not be tested in how much the performance of the employed technologies was compatible with a real UMTS network. Also, the mobile UMTS terminals developed by EXODUS were not available for the trial (cf. also subsection 7.3.4.3 above). This concerns mainly network induced delays and error rates. It is true that also the EXODUS testbed would only have been an approximation of a real UMTS network, but it certainly would have been closer to it than a LAN.

What could be retained?

The main objectives of the trial could still be realised, i.e. it could be shown that the system works when used by real users, delays induced by the ASPeCT implementation could be measured, the acceptability of that delay and of quality of service as perceived by users, and feedback from users regarding the functionality of the application could be obtained. The awareness and acceptance of users of the employed technology and its security implications could be tested as well.

7.4 Evaluation

7.4.1 Evaluation of demonstrator

7.4.1.1 First demo

The results of the evaluation of the first demonstrator were laid down in [D16].

7.4.1.2 Second demonstrator

The second demonstrator was evaluated in the context of the field trial. There was no separate evaluation document as there was for the first demonstrator. As the two demonstrations are quite similar from a user's perspective (although not from a system perspective) much of what was said for the evaluation of the first demonstrator is valid for the second demonstrator as well.

7.4.2 Evaluation of field trial

7.4.2.1 Technical feasibility

7.4.2.1.1 Objectives

The goal here was to analyse the performance of the trial configuration, and to confirm that the proposed approach provided a suitable basis for practical applications. Measurements were made of critical components such as cryptographic operations.

Among the objectives of the analysis was to find the average time of the cryptographic processes and of all the sessions (authentication/initialisation, charge ticks and re-initialisation) as well as the maximum and minimum delay. The results will help to evaluate whether the protocol can work in an efficient and reliable manner in a real system. They can also to identify which procedures contribute the main delays and to provide possible insights into potential improvements.

7.4.2.1.2 Methodology

A high-resolution timer was used to give high precision measurements. The system's high-resolution performance counter can act as a high-resolution timer when combined with the frequency of the counter. The approach was to record the value of the counter at the start and finish of the function to be measured (in the following figures, these observation points are identified as *measure[i]*). The difference between the two values, in counts, when divided by the frequency of the counter, in counts per second, gives the elapsed time, in seconds, for the function.

The *QueryPerformanceCounter* function returns the current value of the high-resolution performance counter. Calling this function at the beginning and end of the section of code that represents the function to be measured, we get the values of the counter at those two points. The *QueryPerformanceFrequency* function returns the frequency of the high-resolution performance counter in counts per second and dividing the counts with this value we get the time in seconds.

The *measure[]* entries for the authentication and initialisation protocol are shown in Figure 7.12.

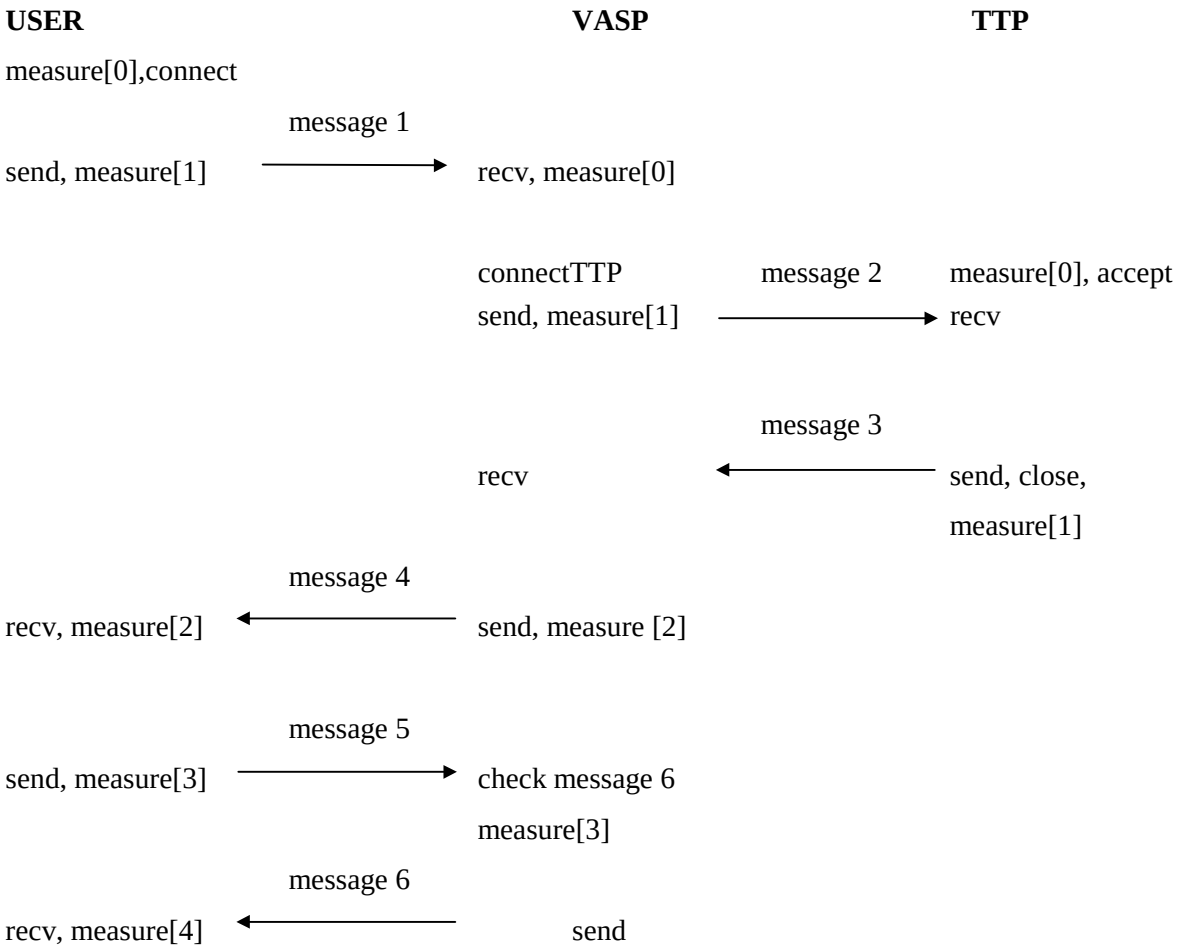


Figure 7.12 - Authentication and Initialisation of Payment Protocol

The way that the authentication protocol works is as follows (in each of these steps a counter value is stored in one of USER's, VASP's, or TTP's log file):

1. USER initiates the protocol and connects to VASP (measure[0] for USER).
2. USER sends the first authentication message to VASP (measure[1] for USER). The difference between measure[1] and measure[0] from the previous step measures the connection establishment and the construction of the first message, which involves the generation of a random number and computation of g^u.
3. VASP receives the first authentication message (measure[0] for VASP).
4. VASP connects to TTP (measure[0] for TTP).
5. VASP sends the second authentication message to TTP (measure[1] for VASP). The difference between measure[1] and measure[0] from step 3 gives the time required by VASP to connect to TTP and "forward" the first authentication message received from USER by slightly modifying it.
6. TTP computes the response and sends the third authentication message back to VASP (measure[1] for TTP). The difference between measure[1] and measure[0] from the previous step gives the total time that TTP needs to complete an authentication session. It involves connection establishment, generation of a time-stamp, certification chains, and a signature.
7. VASP receives the third authentication message from TTP and sends the fourth authentication message to USER (measure[2] for VASP).
8. USER receives the fourth authentication message (measure[2] for USER).

9. USER generates the fifth authentication message which includes a smart card signature and user dialog boxes. Then he/she sends the fifth authentication message to VASP (measure[3] for USER). The difference between measure[3] and measure[2] from the previous step gives the time required by USER to compute the fifth authentication message and by the human user to respond to a USER's dialog box.
10. VASP receives the fifth authentication message (measure[3] for VASP) sent by USER and checks it. Then, it sends the sixth authentication message to USER. The difference between this timer value and the one stored in step 3 gives the total time required by both VASP and TTP to complete the authentication.
11. USER receives the sixth authentication message (measure[4] for USER). The difference between measure[4] and measure[0] from step 1 gives the total time required by USER to complete one authentication session.

Whenever VASP requests a payment from USER, the Charge Ticks Protocol runs. This involves the following steps and entries of counter values:

1. VASP sends a payment request to USER (measure[0] for VASP).
2. USER receives the payment request (measure[0] for USER).
3. USER checks the request, computes the response and sends it back to VASP (measure[1] for USER). The difference between measure[1] and measure[0] from step 1 gives the time required by USER to perform the δ iterations of the one-way-function (where δ is the number of ticks whose payment is requested by VASP).
4. VASP receives the response and checks it (measure[1] for VASP). The difference between this counter value and the one taken in step 1 gives the total time of a charge ticks protocol session.

The *measure[]* entries for the charge ticks protocol are shown in Figure 7.13.

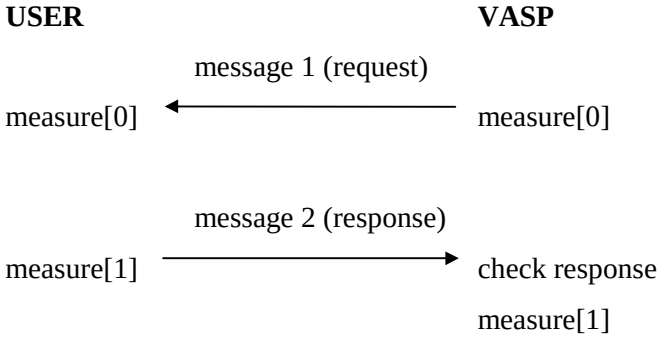


Figure 7.13 - Charge Ticks Protocol

When the total number of ticks requested by VASP exceeds the maximum number of ticks that USER can commit himself by one signature then the re-initialisation of payment protocol is run. This protocol involves the following steps:

1. VASP sends the first re-initialisation message (measure[0] for VASP).
2. USER receives the message (measure[0] for USER) and checks it.
3. USER constructs the response and sends it back to VASP (measure[1] for USER). The difference between measure[1] and measure[0] from the previous step gives the total time required by USER to check the request received from VASP, generate random α_0, compute $\alpha_T = F_{IV}^T(\alpha_0)$ and a signature.
4. VASP receives the response from USER (measure[1] for VASP). the difference between measure[1] and measure[0] from step 1 gives the total time of a re-initialisation of payments protocol session.

The *measure[]* entries for the re-initialisation of payment protocol are shown in Figure 7.14.

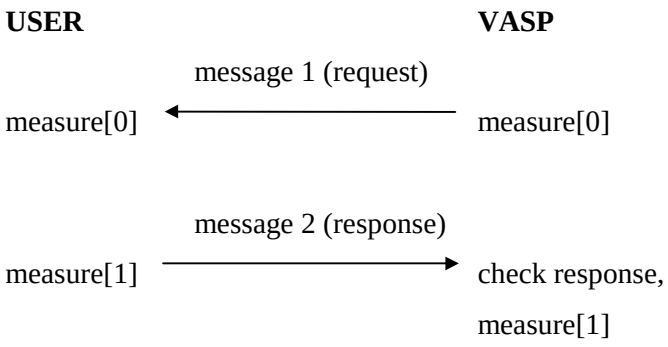


Figure 7.14 - Re-initialisation of Payment Protocol

7.4.2.1.3 Trial measurements (on-line)

A summary of the measurements taken for the three protocols follows. Table 7.1 shows the timings in milliseconds for all the sessions measured. All the times are in mesas.

	Authentication and Initialisation of Payment protocol			**Charge Ticks Protocol**		**Re-initialisation of Payment Protocol**	
	USER	VASP	TTP	USER	VASP	USER	VASP
Min:	2925.777	2887.838	103.536	0.543	9.213	992.364	1175.874
Max:	81130.783	81048.973	177.738	110.34	252.296	1151.766	1423.963
Avg:	17071.018	17022.493	129.183	8.678	62.088	1049.825	1247.986

Table 7.1 - Timings of the protocol sessions

The detailed timing measurements for the *authentication* and *initialisation of payment* protocol give the results shown in Table 7.2 (all the values shown are in milliseconds).

	Authentication and Initialisation of Payment protocol			Charge Ticks Protocol		Re-initialisation of Payment Protocol	
	USER	VASP	TTP	USER	VASP	USER	VASP
Min:							
Meas[1]-meas[0]	23.690	18.193	103.536	0.543	9.213	992.364	1175.874
Meas[2]-meas[1]	739.985	704.324					
Meas[3]-meas[2]	1938.351	2107.447					
Meas[4]-meas[3]	167.316						
Max:							
Meas[1]-meas[0]	31.389	90.722	177.738	110.34	252.296	1151.766	1423.963
Meas[2]-meas[1]	872.978	839.576					
Meas[3]-meas[2]	80133.047	80310.369					
Meas[4]-meas[3]	240.101						
Avg:							
Meas[1]-meas[0]	29.137	45.851	129.183	8.678	62.088	1049.825	1247.986
Meas[2]-meas[1]	786.062	726.439					
Meas[3]-meas[2]	16079.854	16250.203					
Meas[4]-meas[3]	175.966						

Table 7.2 - Detailed description of the timings of the three protocols

A number of extreme measurements were recorded in the trial, in the *Charge Ticks* and *Re-initialisation of Payments* protocols, were excluded from the above summary as they were caused by network problems and were not considered to provide typical protocol delays.

It should be noted that in the authentication and initialisation of payment protocol "Meas[3]-meas[2]" values do not give any indication about the satisfactory computation of the cryptographic algorithms, from a timings point of view, as there is an interaction with the human operator which of course affects both USER's and VASP's measurement.

7.4.2.1.4 Cryptographic operations measurements (off-line)

The individual delays caused by the cryptographic functions themselves, without the interference of user interaction, network delays or operating system management delays, were measured off-line on a Pentium Pro 200MHz[2] PC; each "Measurement" represents one step in the protocol):

1. Authentication and Initialisation of Payment protocol

Measurement 1: The elliptic curve exponentiation during the generation of 1st authentication message.

The delay introduced by this computation is approximately 35 msecs.

Measurement 2: There are no cryptographic functions executed during the generation of the 2nd authentication message. The delay introduced by the generation of the message is less than 5 msecs.

Measurement 3: The cryptographic function that causes most of the delay during the generation of the 3rd authentication message is the elliptic curve based signature generation, which takes about 35 msecs.

[2] this is slower than the processors used in the trial configuration PCs, explaining the trial measurement for the whole 1st authentication message (29.137 msecs) is less than the off-line exponentiation (35 msecs)

Measurement 4: When VASP receives the 3rd authentication message from TTP the following cryptographic procedures take place in VASP:

i. Two verifications on the certificates, which take approximately 65 msecs each.

ii. Verification of TTP's signature, which takes approximately 65 msecs.

iii. Computation of $(g^u)^v$ as part of the generation of the secret key *K*, which takes approximately 35 msecs.

In total the cryptographic functions executed by VASP before and during the generation of the 4th authentication message take approximately 230 msecs.

Measurement 5: When USER receives the 4th authentication message from VASP the following cryptographic procedures take place:

i. The two verifications on the certificates, which take approximately 65 msecs each.

ii. Verification of TTP's signature, which takes approximately 65 msecs.

iii. Computation of $(g^v)^u$ as part of the generation of the secret key *K* takes approximately 35 msecs.

iv. Generation of an elliptic curve based signature on USER's smart card, which takes approximately 900 msecs.

In total the cryptographic functions executed by USER before and during the generation of the 5th authentication message take approximately 1130 msecs.

Measurement 6: The verification of USER's signature in VASP takes approximately 65 msecs.

2. Charge Ticks Protocol

Measurement 1: During the generation of the 1st message (request) sent by VASP to USER there are no cryptographic operations taking place. The delay introduced by the generation of the 1st message is less than 5 msecs.

Measurement 2: The 512 iterations of the one-way function in USER take approximately 5 msecs.

Measurement 3: VASP checks the response sent by USER and therefore it has to reiterate the hash function, which takes approximately 5 msecs.

3. Re-initialisation of payment protocol

Measurement 1: There are no cryptographic operations taking place during the generation of the 1st message.

Measurement 2: Generation of a signature on USER‘s smart card takes approximately 900 msecs. The rest of the delays might have been caused by operations that involved heap memory and by the initialisation of the payment variables.

Measurement 3: Verification of USER‘s signature takes approximately 65 msecs.

The delays introduced by the cryptographic functions are summarised in Table 7.3.

Cryptographic function	Delay
Elliptic curve exponentiation	35 msecs
Verification of a certificate	65 msecs
Verification of an elliptic curve based signature	65 msecs
Computation of g^{uv}	35 msecs
Elliptic curve based signature generation on PC	35 msecs
Elliptic curve based signature generation on smart card	900 msecs
Iteration of the one-way function (512 times)	5 msecs

Table 7.3 - Cryptographic functions timings

7.4.2.1.5 Discussion

The performance of the protocols implemented for the trial seems to be at an acceptable level. Some exceptions of far too big values that were measured during the trial were caused by the slow response of the human operator during USER's dialog that takes place between measure[2] and measure[3]. If these values are excluded from the computations the remaining ones are in an acceptable range. The large delays introduced by the human user interaction is also implied by the standard deviation function, which in the case of a USER's authentication session has the value 21033.7.

In a GSM network the average set-up for a mobile originating call is about 4.3 secs. The average time for the authentication and initialisation of payment protocol, which includes the additional procedure of initialisation of payment, is 17.071 secs. However, as mentioned earlier, the main reason that causes this long time is the delay introduced by USER's interaction. Therefore, considering the minimum delay introduced by USER among all the sessions that took place, the total time for the specific session is just 2.93 secs which is less that the quoted GSM set-up time.

It should also be mentioned that one of the factors that introduced delays was the limited communication speed between PC and card reader and between card reader and smart card.

7.4.2.2 User acceptability

7.4.2.2.1 Overall impression

All users successfully registered with the TTP to obtain a UIM. None of the users had any major problems launching the applications on the user terminal.

During the initial phase of the trial expert users were first required to select the list of hotels from the catalogue on their terminal. The only major problems here were concerns from users about the fact that data was downloaded to the browser before the user was able to acknowledge a dialog box asking for a commitment to the VASP's charging rate. This is basically a matter of policy by the value added service provider, regarding how much service he is prepared to provide without guaranteed payment in exchange for a higher performance and user friendliness of the system.

Users were asked to observe the protocol messages displayed in the tracer. Most users just acknowledged that the protocol was being executed, but some users gave detailed and accurate description of the main events reported by the tracer. Understanding and awareness of the underlying protocols and mechanisms seemed to be high.

All users confirmed that the actual payment information displayed in the payment status window was generally the same as what they had expected. A few problems did arise because of rounding errors resulting in users being charged a few ticks more than they expected. The problem occurred because of the way individual prices listed in the catalogue are rounded down to the nearest tick. As this caused irritation a different method of rounding prices should be employed. The charges are based on the application of the

tariff on the total number of bytes downloaded in a session, rather than on individual documents which are downloaded.

One user reported that, according to the payment status window, ticks were released (i.e. they were charged again) if a previously downloaded page was viewed another time. It is acknowledged that it is desirable from a user's perspective that there is no additional charging if the same page is downloaded a second time. This desirable feature had already been identified in [D16], the evaluation of the first demonstrator. This feature could not be implemented in the trial, however, due to a lack of resources.

No major problems where reported during further browsing. Many users reported successful re-initialisation of the payment schemes. From the users perception this was generally regarded as a "reload", or the process of obtaining more credit.

7.4.2.2.2 Results of questionnaire

Trial users were asked to complete a questionnaire after they used the system. Two different questionnaires were used for the experts and non-experts. The results of the questionnaires for each of the two user groups are now discussed separately.

7.4.2.2.2.1 Expert users

Documentation: All expert users seemed very satisfied with user documentation (all gave 4 marks out of 4). However, one user indicated that it was difficult to assimilate all the material within the short allocated timeframe. Furthermore, one user said that it was not always clear how much detail was required in the questionnaire. Further information on the payment model would also have been of use to some users. Such information was available in the enclosed publication, but the timeframe was probably too short for users to take in this information.

Support: All users were satisfied with the on-site support during trial (most gave 3 or 4 marks out of 4).

User interface: The user interface was fairly acceptable to the trial users (an average of just under 3 marks out of 4). The importance of the payment status window was indicated by most users, with many suggesting that the system would benefit from an improved payment system window design. Most users also indicated that they would appreciate an on-screen history of transactions rather than just the ability to view previous transactions off-line. However it should be noted that the user interface would be significantly improved in a real product. The tracer was also judged to be user unfriendly, but such an interface would not be present in a real product and is only present in the trial scenario since it helps to give additional information to observers of the trial.

Stability: In general, the system was judged to be stable. Most of the expert users, however, had some problems with the applications locking up. This was not typically due to software failures, but was related to the fact that users were blacklisted by the service provider once they tried to gain access to services without paying with their smart card. Expert users tended to make such experiments, but were not aware of the blacklisting feature. In these cases the field trial leader had to manually remove these users from the blacklist using a special option in the VASP's secure billing software.

Performance: Overall performance was judged to be satisfactory. Most users did not perceive any performance degradation which they could attribute to the initialisation protocol, although many said that it is very difficult to give an objective comparison. Only one user specified an actual metric (< 1 second) for the expected delay due to this protocol. Users perceived a greater level of degradation relating to the charge protocol. Perhaps this is because users are prepared to wait for connection, but while connections are active they expect fast a response. The re-initialisation protocol was perceived to have a greater level of degradation than the initialisation protocol, even though the re-initialisation protocol should involve less computation and shorter messages. Again this can be explained by the fact that users are much less willing to tolerate delays during downloads. However, it should be pointed out that most users still considered the delay due to the re-initialisation protocol to be fairly acceptable.

Security awareness: Most users considered server authentication to be important, although some do not seem to see it as an essential requirement. There is more of a split for client authentication, with some users

considering it to be a strong requirement, while others considering it not to be important at all. The need for the user to be able to generate evidence as part of the payment scheme was generally acknowledged. There was a reasonably strong awareness of the need to spread payments, but it is unclear as to what level of granularity users expect. All users could see the need for security mechanisms, some could also appreciate the subtleties of the protocol. All users had some understanding of the role of the TTP, with most of them also claiming to have a full understanding. All users had some understanding of the role of the UIM, some of them also claimed to have a full understanding.

7.4.2.2.2.2 Non expert users

Documentation: All users seemed fairly satisfied with user documentation (all but one user gave 3 or 4 marks out of 4).

Support: All users were satisfied with the on-site support during trial (most gave 3 or 4 marks out of 4).

User interface: The user interface was fairly acceptable to the trial users (an average of just under 3 marks out of 4). However, some users indicated that the payment status windows were difficult to understand and should be improved. Having said that, it must be remembered that the field trial was based on a prototype solution and that the user interface would be significantly improved in a real product.

Stability: In general, the system was judged to be stable. However, some of the non expert users did have problems with the applications locking up. This was not typically due to software failures, but was related to the fact that users were blacklisted if they tried to gain access to services without paying with their smart card. Non expert users were less likely to do this than the expert users (see above).

Performance: Overall performance was judged to be satisfactory. Most users did not perceive any performance degradation which they could attribute to the initialisation protocol, although many said that it is very difficult to give an objective comparison. Users perceived a greater level of degradation relating to the charge protocol and the re-initialisation protocol. Again this can be explained by the fact that users are much less willing to tolerate delays during downloads.

Security awareness: Some non expert users seemed to appreciate the role of security mechanisms in providing an incontestable charging service in the trial scenario. Some users could also appreciate the role of the smart card in the trial by considering it to be an alternative payment device similar to a credit/debit card or an electronic purse. In general security awareness among non expert users was quite low. As a result non expert users tended to place trust in the provision of adequate security features by operators and providers rather than obtain a detailed understanding of the role of the various security mechanisms themselves.

7.4.2.3 Discussion on trial results

The system was judged to be stable by users. Overall performance was judged to be satisfactory. Most users did not perceive any performance degradation which they could attribute to the initialisation protocol, although many said that it is very difficult to give an objective comparison. The user interface was fairly acceptable to the trial users (but bear in mind that the focus of the ASPeCT work was not to provide a product level Graphical User Interface). The importance of the payment status window was indicated by most users, with many suggesting that the system would benefit from a improved payment system window design. Most users also indicated that they would appreciate an on-screen history of transactions rather than just the ability to view previous transactions off-line.

Some users found the blacklisting and subsequent barring of users who had failed to pay once (e.g. by not inserting the smart card) confusing. Therefore, more information on the conditions under which the system is working would have been needed. In a real-life scenario, the right balance between the interest of the service provider to protect himself and the interest of the user in a continuous provision of the service would need to be struck.

All users seemed very satisfied with the user documentation and with the on-site support during trial.

The questions relating to the security awareness of users provided mixed results. While a fairly high number of users is well aware of the needs for security, the results also showed that more work is to be done to

educate the general public on security issues which they need to understand before entering into on-line payment services.

7.5 Conclusions

The ASPeCT secure billing work produced two major results:

1. An implementation of a micropayment system to pay for the provision of value added services in a mobile system
2. A public-key based protocol suite for authentication, key agreement and initialisation of payment in a mobile environment

A micropayment system is used in ASPeCT to pay for the provision of valued added services which provide information to the user based on WorldWideWeb technology. The novelty is not the payment protocol itself, but the way in which it is integrated with the authentication protocol proposed for the mobile system UMTS and the payment scenario for basic and value added services in UMTS.

In particular, the problem had to be solved how to integrate protocol layers realising the ASPeCT authentication and payment protocols with the standard http over TCP/IP stack. This problem was solved by introducing an ASPeCT security layer in between the http layer and the TCP/IP layer in such a way that the ASPeCT security layer provides the same Winsock interface to the http layer above which it uses from the TCP/IP layer below.

The implemented system has two phases: an initialisation phase in which the user of a value added UMTS service commits to initial values of the payment scheme by a digital signature, and an actual payment phase in which payments are made to the provider of a value added UMTS service. Initialisation is performed as part of a protocol developed by ASPeCT. The use of the digital signature guarantees the security of the bill by providing a non-repudiation service. By the nature of the selected payment system, the guarantee provided by the digital signature extends over all the payments made subsequently in the same authentication session.

The protocol developed by ASPeCT for initialisation of payment also provides for authentication and key agreement between user and network; it was particularly designed to fit the performance constraints of mobile networks. Its design exploits the advances in two fields: Crypto-controller smart cards (which have a co-processor which efficiently supports public-key cryptographic mechanisms) and elliptic-curve cryptosystems (which permit the use of smaller cryptographic parameters).

The main achievements of ASPeCT relating to the first major result are:

- the definition of an electronic payment scenario and charging model for basic and value added service suitable for a telecoms environment;
- the integration of a micropayment system with the http over TCP/IP protocol stack;
- a guarantee for the security and integrity of the bill by coupling the payment with the ASPeCT authentication protocol;
- the implementation of the system in a demonstrator to show its feasibility.

The main achievements of ASPeCT relating to the second major result are:

- development of a protocol suite fitting the constraints of a mobile environment;
- testing the strength of the protocol by theoretical evaluation of possible attacks on the protocol;
- analysing solutions with a similar scope available from the literature and comparing them with the ASPeCT protocol;
- implementing the ASPeCT protocol in a demonstrator thereby showing the feasibility in terms of performance constraints.

8 Detection and management of fraud in UMTS networks

8.1 Introduction

8.1.1 Scope

The scope of the ASPeCT fraud detection and management work was to investigate the various fraud scenarios present in today's Analogue and GSM market and identify the areas where AI technologies could be implemented in a state of the art fraud detection tool which is able to cater for tomorrow's UMTS systems.

The project chose to use three separate AI techniques in four stand-alone tools which, when combined, form a powerful fraud detection engine. This chapter details the background to fraud detection, the design of the four tools and how these tools were integrated into BRUTUS – a web based GUI front-end for user management of the tools. Finally the results from the trials of BRUTUS are given and some comment is made on the way forward for further research.

8.1.2 Purpose

It is estimated that the mobile communication industry loses several million ECU's per year due to fraud. Therefore, prevention and early detection of fraudulent activity is an important goal for network operators. It is clear that the additional security measures taken in GSM and in the future UMTS system make these networks less vulnerable to fraud. Nevertheless, certain types of commercial fraud are very hard to preclude by technical means. These types of fraud will be discussed in section 8.2. The use of sophisticated fraud detection techniques can assist in early detection of such frauds, and will also reduce the effectiveness of technical frauds.

However, it is important to point out at this early stage that it is impossible to totally eliminate fraud. The fraudster will always find a way to beat the system and a fraud detection tool has to be cost effective. Figure 8.1 below, shows the relationship between the cost of fraud and countermeasure devices verses the ability of the fraud detection tool. As efforts are increased to countermeasure fraud the cost of these countermeasures become prohibitively expensive to the point where the cost of the countermeasure is far greater than the cost to the company of the fraud taking place. The thicker curve shows the two graphs summed to find the optimum, most cost-effective level.

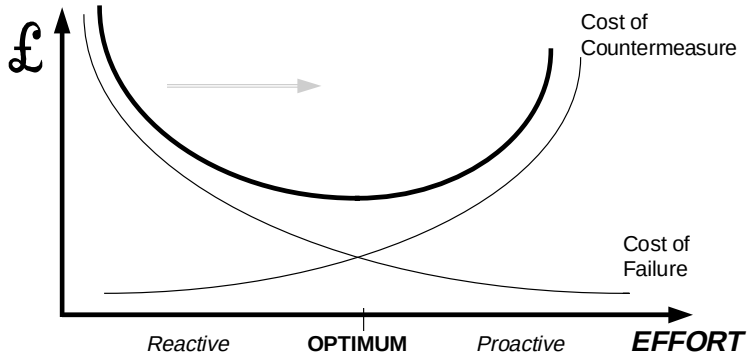


Figure 8.1 - Cost Analysis of Fraud Tools

The use of modern AI techniques is not new in fraud detection. As will be discussed elsewhere in this report, substantial AI expertise has been brought to bear in the area of credit card transaction fraud detection. The problem of fraud detection for credit cards is in many respects more straightforward than for mobile telecoms. Just as an example there is only one type of fraud and only one type of transaction, in contrast with the large number of mobile phone fraud scenarios we will describe below, and the different types of calls that can be monitored in fraud detection. Hence, a straightforward application of the solution strategy developed for credit card fraud is neither feasible nor would it be effective. This work had the task

of studying the problem of fraud detection in mobile telecoms, of identifying the relevant fraud scenarios, determining the resulting fraud indicators and of designing and demonstrating novel techniques that are effective state-of-the-art solutions to this problem.

8.2 Fraud scenarios

This section describes a number of known fraud scenarios. Many new frauds have appeared since the ASPeCT project began. This has been predominantly due to the introduction of pre-payment schemes.

8.2.1 Non pre-payment fraud scenarios

8.2.1.1 Subscription fraud

This is by far the most common fraud encountered on the GSM network. A person uses false identification to obtain a service. Subscription fraud can be further subdivided into two categories. The first is for *personal usage* by the fraudster, or someone he passes the phone onto. The second is for profiteering whereby the fraudster might claim to be a small business to obtain a number of handsets for *Direct Call Selling* purposes. The fraudster now sells on the airtime to people wishing to make long distance calls for long durations with no intention of paying the bill.

8.2.1.2 PABX fraud

Many companies provide a dial-on service through their PABX system in order that employees can use it for international sales purposes from wherever they may be located. This service is open to abuse if a fraudster obtains a password for such a system. Fraudsters often try to guess these PABX passwords by making many short duration back-to-back calls. These calls tend to occur out of office hours and are to fixed landline numbers.

8.2.1.3 Freephone fraud

This occurs when a person uses a calling card service to dial onwards. In some countries, this is illegal in itself. More often though, it is the fact that these services tend to attract fraudulent mobiles. The fingerprint of such activity tends to be back-to-back calls and long duration calls.

8.2.1.4 Call-back fraud

In some countries it is illegal to use a calling card dial back service.

8.2.1.5 Premium rate fraud

This involves the abuse of the premium rate services. It can occur in different ways. For example a person could set up a premium rate sex line with a national operator. The operator is obliged to pay the owner of the line a proportion of the revenue generated. The fraudster then uses a fraudulent mobile to dial this number for long periods. He may also get other people to do the same. The fraudster then pockets the revenue without paying for the calls.

A further way in which premium rate services can be abused is by setting up a fraudulent mobile to divert calls to a popular premium rate line. The caller then only pays normal rates whilst the fraudulent mobile picks up the tab at the premium rate. Characteristics are long back-to-back calls.

8.2.1.6 Fax-back and malicious call-back fraud

Some companies provide a fax-back service to distribute information to interested parties. This service is open to abuse if requests are directed at premium rate lines. Distressing messages can be left on an answer-phone requiring a number to be dialled for further info. The caller is kept on the line using every trick in the book. Some calling card companies provide a call-back facilities instead of a free-phone number, these too can be redirected to a premium rate line.

A very recent scenario is the use of an Internet virus that changes the telephone number in your modem settings to dial an ISP somewhere in the Caribbean, needless to say at a premium rate. Unless you noticed the tones that the modem was dialling, you could easily loose a fortune.

8.2.1.7 Technical internal fraud

This is a very complex problem with many diverse characteristics. One scenario might be that an employee of the network operator uses information that he is privileged to in order to assist a further fraud to occur. Cloned SIMs coupled with authentication data can be used to impersonate a genuine subscriber. If this occurs, the genuine subscriber is often the first to notice his bill has increased. Due to the duality of this fraud it is often referred to as *superimposition fraud*. We would anticipate this fraud to be on the increase due to recent successes in GSM cloning.

Technical fraud does not need to be this complex however. It can be as simple as a service provider handing out subscriptions to individuals who would not normally pass vetting procedures. Dealers often receive large financial incentives to obtain subscribers. Dealerships have been set up in the past to sign up fictitious people to a network just to obtain the financial incentives. It is only a lot later that the NO realises that these accounts have been 'rather inactive'.

Within this category comes a new type of fraud we discovered only recently. In addition to incentives, service providers can be penalised when subscribers disconnect. To be precise, this happens if the number of disconnection's exceeds a threshold in a given month. We recently observed that service providers were barring subscribers who were terminating their contract, instead of disconnecting them, until a month where the disconnection would be acceptable, thus avoiding any penalties.

8.2.1.8 Mobile to mobile fraud

This is very similar in nature to PABX fraud. A fraudulent mobile, with international forwarding enabled, is used to provide a dial-on service to other fraudulent mobiles that do not have this service.

8.2.1.9 Roaming fraud

This is when a fraudster makes use of the delays in the transfer of Toll Tickets through roaming on a foreign network or through using a foreign SIM on an UK network. (Technically the latter should be called 'visitor fraud'.) Weekend usage presented a very reliable delay in the past. Such delays are thankfully being reduced by modern billing systems.

8.2.1.10 Tumbling fraud

Not achievable by your average fraudster. A roamer on a foreign network utilises the delays in Authentication to make calls. When the authentication challenge is scored incorrectly by the home network, the call is dropped. By this time the fraudster has changed his ID. This clearly requires information known only to the NO and hence can be coupled with technical internal fraud.

8.2.1.11 Hijacking

Circumventing the networks security using a powerful radio to seize a communications channel that has been established by a valid user during the process of placing a call.

8.2.1.12 Handset theft

Obvious, but may become a growing problem. The more readily accessible mobile phones are to infrequent users, the longer the phone might be available for fraudulent use before it is noticed missing.

8.2.2 New prepay fraud scenarios

In this section the ways in which fraudsters are adapting to prepayment are discussed, which was introduced to reduce exposure to fraud and bad debt. Due to the infancy of this technology there are a number of obvious and not so obvious loopholes that enable fraudsters to make money. This mainly revolves around the voucher scheme whereby a person walks into a shop and purchases airtime via a physical voucher. The voucher contains a code number that when entered into the phone enables calls to be made.

8.2.2.1 Cheque fraud

A very basic fraud whereby large quantities of airtime are purchased in shops using stolen cheques. The cheques then bounce, by which time the fraudster has sold on the vouchers at discounted prices.

8.2.2.2 Credit card fraud

By using credit cards as a guaranteed form of payment, vouchers can again be purchased from shops or top-up time can be bought over the phone just by issuing the credit card details or even over the Internet.

8.2.2.3 Voucher theft

The physical vouchers bought from a shop can be snatched or stolen.

8.2.2.4 Voucher ID duplication

This occurs when the quality of the prepay voucher is substandard and the voucher's ID can be read without breaking the seal. As a result, the airtime might have already been spent by the time that the customer purchases it.

8.2.2.5 Faulty vouchers

If a batch of vouchers are found to be faulty and they are not correctly disposed of or accounted for then there is a clear opening for fraudulent abuse of position. This is getting back into the realms of technical internal fraud.

8.2.2.6 Network access fraud

This occurs if unauthorised access to the network is gained in order to remove the prepayment marker that is attributed to an IMSI. This enables the fraudster to make calls as if he had a genuine credit subscriber account. Arguably, as there is no billing information for this customer he may remain undetected for longer.

8.2.2.7 Network attack

This operates in a very similar manner to PABX fraud in that a fraudster repeatedly tries to guess a pre-paid code by entering DTMF tones from the keypad. This might be coupled with a faulty voucher scenario where only partial completion of a set of numbers is required.

8.2.2.8 Long duration calls

This involves the exploitation of a bug in many hot billing systems in that calls that are in progress cannot be taken down. Thus with a single unit you can call granny in the New York Bronx for hours at a time.

8.2.2.9 Handset theft

As prepay phones are not greatly subsidised, they are quite attractive to thieves and easy to resell. Because there is no SIM locking involved with prepay mobiles, a service can be obtained from a number of networks, even foreign ones.

8.2.2.10 SMS abuse

There is a distinct possibility that it may be possible to still utilise an SMS service once a voucher has expired, even when roaming.

8.2.2.11 Roaming fraud

It is already possible to roam using a prepaid service that naturally opens up the home network to roaming prepayment fraud.

8.3 Requirements for fraud detection

8.3.1 Introduction

In this section, an overview of the basic requirements for fraud detection is presented.

It is divided up into four sub-sections. Section 1 is this introduction. Section 2 describes the fraud detection environment, and discusses the relationships between the elements. Section 3 then discusses the functional requirements of a Fraud Detection Tool (FDT), whilst Section 4 discusses the system requirements.

8.3.2 The fraud detection environment

There are three main roles that need to be considered in a discussion about the Fraud Detection Environment. The first is the User, the entity that makes calls on the network. The User has a contractual relationship with the second entity, the Service Provider (SP), who charges the User for the calls that are made. The third entity is the Network Operator (NO) who sells blocks of airtime to the SP for selling on to the User. There is no direct contract between the NO and the User.

The purpose of an FDT is to detect fraudulent behaviour of Users from their usage data before the cost of such activity becomes too great. To achieve this, it is clear that the tool should be placed where it can receive this usage data as quickly as possible. This means that the FDT should optimally be closely connected to the Network.

Within the Network, there are two possible sources of usage data that can be used for Fraud Detection. The first source uses the Toll Tickets (or Call Detail Records (CDRs)) which are the records of the calls produced for billing purposes and are generally constructed once the call has finished. The second source is the Signalling Data that is produced in the network.

The advantage of using Signalling Data is that it makes more usage information available. For example, the location of the User at every Location Update would be available, as well as the opportunity to monitor the set-up of calls when they happen rather than only once the call is finished. However, the advantage of this approach is also the disadvantage. The quantity of data produced is of the order of one or two orders of magnitude greater than that produced for billing.

The advantage of using the Toll Tickets (TTs) is that the information that is produced for billing purposes also contains the important usage behaviour that is useful for Fraud Detection. However, billing data needs to be gathered securely, and not necessarily quickly. Hence the data is often one day old before it is processed. To allow for such things as hot billing, where bills can be created very rapidly, and also to minimise this fraud risk window, a mediation device may be included in the network that polls the switches for their TTs on a regular basis. The data thus collected can be considered to be pseudo-real time, as the difference between real time delivery and polled retrieval can be as small as is practically possible.

Once the TTs are processed, actions need to be taken in response to any alarms that are raised by the FDT. Such responses could be to question the Users to see if they were valid or fraudsters. However, another level of complexity exists, as there is no direct contract between the User and the NO. Hence, the alarms will tend to be distributed to the relevant SPs, and they will take appropriate action.

Within this environment there are functional requirements and system requirements that are required to be met to create a useful FDT.

8.3.3 Functional requirements

8.3.3.1 TT data feed

The FDT data feed of TTs must be complete and must correctly reflect the actual usage. Any artificial filtering or ordering of subscriber call data must be avoided, as it would certainly lead to false conclusions.

The FDT will also require associated network data for the correct operation of the tool, for example, base station information, billing information etc.

8.3.3.2 Alarm processes

The NO must have in place the mechanisms for prompt actions following the relevant fraud alarms (e.g. notification of corresponding SP, barring etc).

8.3.3.3 Fraud boundaries

The ability must be provided to discern fraud from heavy or unusual but legitimate network usage. Therefore fraud detection must find the right balance between false alarms and correct detection of fraud instances. This is a more or less dynamic trade-off, which can change during different periods or for different services. Despite the importance of fraud detection, NOs and SPs are justifiably not willing to risk

bothering many of their 'good' subscribers, based on the misclassification margin. The fraud detection tool user should be able to tune this trade-off and control the fraud detection rate against the false alarm rate.

8.3.3.4 Outcome visibility

The output of fraud detection must be meaningful to human users. Assistance should be provided in the form of facts upon which business decisions can be made, e.g. barring, disconnection, legal measures etc. Therefore, together with the fraud alarms, the reasons for the raising of the alarm also need to be available.

8.3.3.5 NO–SP co-operation

A level of co-operation regarding the strategy for fraud prevention must be established between the NO and the SP, since each party possesses complementary information and both parties suffer the consequences of fraud. The NO has the means and the data necessary to identify suspicious subscriber behaviour. On the other hand, the SP owns the subscription, and handles personal data and billing records. Normally the SP is unwilling to reveal the credit history of its subscribers to any third party including the NO, even if the latter provides indications of potential fraudulent activity. However, its feedback is required by the NO for FDT result validation. It can normally be expected that a definite decision on whether an alarm observed by the NO's FDT can be recorded as fraud and acted upon, should be taken in co-operation with the respective SP.

8.3.3.6 Fraud prediction

Taking a step further, fraud detection effectiveness increases if the actual fraud intention can be detected, before fraud is committed. In such a scenario, the Network Operator is warned about abnormal subscriber behaviour that may contain fraudulent characteristics and can prepare for immediate action when more concrete proof becomes available. A more sophisticated FDT would be needed in this case, providing more than simple rule-based detection.

8.3.4 System requirements

8.3.4.1 Performance

To minimise fraud losses, it is important that the FDT operates as close to real-time as possible. Using a mediation device to provide such a feed has another consequence, that of varying processing requirements. The usage of the Network varies throughout the day, and there will be a 'busy hour' in the day, which is usually taken to be where one hour contains 15% of the day's calls. Hence the system will have to be able to process TTs at a rate governed by this value.

8.3.4.2 Customisation

The FDT must be easily tuneable for customisation of fraud detection sensitivity and for alarm filtering.

8.3.4.3 Scalability-flexibility

The FDT must be scalable and adaptable in the fast growing mobile network environment. That would mean that any thresholds set during the initialisation period should not lower because of the network expansion. The FDT must also be flexible towards new and changing fraud methods and characteristics (i.e. behaviour changes) that are expected to manifest themselves in the future.

8.4 Available technologies

There are currently over 30 fraud management solutions available in marketplace. Very few have performed a rigorous analysis of the artificial intelligence technologies that can be applied to reducing fraud. The ASPeCT project has certainly accomplished this.

What the ASPeCT tool currently does not possess, which marketable products do, is a flexible interface, strong database technologies, user friendliness and case management. In the latter stages of the project, a number of these issues were tackled. We do not think that it would take very long to enhance the ASPeCT tool to a stage where it could be marketed.

In this section we address some of the features of current market products which are highly desirable and necessary, in addition to the technology of the fraud engine. We omit the individual naming of companies for confidentiality reasons.

8.4.1 Geographical information

In order to set velocity traps, geographical information is required. At the very minimum we would need the distances between different cell sites. Very few of the tools on the market have complete geographical databases.

A further advantage of this comes when we consider case management. When investigating potential crime syndicates of organised mobile telephone fraudsters, the crime scenario is very often complex. For the fraud analyst, it becomes virtually impossible to see the complete picture. Rather than being able to eliminate whole syndicates, only certain links of the chain removed. These links are all too frequently replaced very quickly. With the ability to track the origins and destinations of Fraudsters calls, we might hope to tackle the syndicate as a whole. Currently operators interested in working at this level rely on further interpretation software.

8.4.2 Database management systems

The efficient performance of fraud management system is crucial. Without sophisticated database technologies we cannot hope to meet the through put for future network operators. Current products on the market have given a lot of thought to this issue. One company in particular, from the UK, has achieved a through put off 60 million toll tickets per day. This is accomplished using a standard desktop server.

In addition to the general performance of the system, in order to facilitate strong case management features, we need to be able to query the database. Results need to be obtained quickly and should not hamper normal processing. Some products, that do not have particular strong case management features, have opted for flat filing systems. This can significantly increase through put, but at the expense of case management.

8.4.3 Unsupervised profiling

A good forward thinking fraud management product will be data driven. With the nature of fraud changing all the time, we cannot anticipate that one set of heuristics will suffice. The impact of UMTS on current fraud management technology could be considerable. The ability to let the data determine what currently are the important features will enable the production of efficient behaviour profiles, crucial to long term fraud management and the ever increasing demand on the system.

As technology improves and more and more data products appear on the network, so the whole nature of profiling will change. Only one other product that we are aware of has addressed this issue and utilises a clustering technique to characterise the important features of the data and thus reduce redundancy. Other companies are now addressing the issue, however some play down its significance.

8.4.4 Flexibility

The ASPeCT fraud management tool was designed to prove that the AI concepts developed do indeed perform the task required of them. A market product needs to be flexible to meet the needs of the individual operator. Below are some flexible criteria, which are provided in some existing market products.

- The provision of a **rule definition language** to allow triggers to be built from functions of any number of variables within the system.
- **Flexible API** to facilitate making custom modifications such as interfacing with the network monitoring system.
- Provision made for **future technology to be integrated**.
- A **communications infrastructure** to both pass information in and out of the fraud management system to related areas of the business.

- **Help facilities and decision support** that are regularly updated either in-house or by the product vendor.
- **Helpful and informative GUI**.
- **Flexible report writing facilities**.

In summary, what we have witnessed in the market place to date are products which focus more on peripheral information, sound data management and generally a rules based approach. Many products are being marketed that claim to utilise Neural Networks, that when it boils down to it, contain little functionality. The system is still almost wholly reliant on the rules base and NNs are used as a marketing gimmick. Fortunately this is now changing as the product base has become established and the market place is aware of the potential of real AI techniques.

8.5 ASPeCT approach

8.5.1 Introduction

This section details the design and implementation of the ASPeCT fraud detection tool. The system is a suite of four tools and a front-end GUI known as BRUTUS. The four tools used will be briefly described below. The following sections will give further details of each tool.

From the other approaches described above, it is clear that a Rule-Based tool is a necessity to determine if fraud is occurring. It is a white box approach and hence the end-user can be given a reason why the tool has flagged an alarm for a particular user. This would be essential for the legalities of denying a user service on the network.

However, a Rule-Based tool on its own is not a complete enough tool to deal with the complexities and intricacies of detecting mobile telecoms fraud. Some form of "fuzzy" soft-computing technique is required to handle scenarios that cannot be precisely specified by rules (a situation that is prevalent in mobile Telecom fraud. This is less common in other fraud detection areas, such as credit card fraud). Therefore a supervised Neural Network is also implemented to be able to deal with strange or abnormal scenarios.

The fraudster is constantly looking for new ways to beat the system, so a method is also required to detect new or novel types of fraud. Here an Unsupervised Neural Network is used to look at how a user's behaviour changes over time. It needs no prior knowledge of fraud unlike the previous two tools. There are two Unsupervised Neural Networks used in BRUTUS. An A-Number analysis, which detects changes in the users' behaviour on the phone, and an international B-Number analysis, which looks at specific changes in behaviour of a user making international calls.

These four AI tools are integrated together to form the complete ASPeCT Fraud Detection Tool - BRUTUS. Each tool can detect separate suspicious looking or unusual behaviour and raise an alarm appropriately. A GUI has also been implemented for easy user management. For the prototype system it is run via a web browser. The GUI can keep track of suspicious users and allows the operator to look at a specific user's calls to give the final decision whether they are a fraudster or not.

8.5.1.1 User profiling

The fraud detection tool uses a completely data-driven approach. All the information gathered by the tool only comes from the individual user-specific Toll Tickets (Note that no geographical information is used). A method must be found to profile each user and extract relevant information from the Toll Ticket to try to detect any fraudulent use. The descriptions of each of the four tools below also include a section on how user profiling is implemented within each tool.

8.5.1.1.1 Absolute or differential analysis

Toll Tickets are data records containing details pertaining to every mobile phone call attempt that is made. Toll Tickets are transmitted to the network operator by the cells or switches that the mobile phone was communicating with at the time due to proximity.

In addition to providing necessary billing information, Toll Tickets contain a wealth of information that can be used to catch a fraudster. Section 8.2 introduced a number of fraud scenarios that have been encountered in analogue networks and detailed the characteristics of Toll Ticket parameters observed whilst the fraud is being committed. Existing fraud detection systems tend to interrogate sequences of Toll Tickets comparing a function of the various fields with fixed criteria known as *triggers*. A trigger, if activated, raises an alert status that cumulatively would lead to an investigation by the network operator. Such fixed trigger systems perform what is known as an *absolute* analysis of the Toll Tickets and are good at detecting the extremes of fraudulent activity.

Another approach to the problem is to perform a *differential* analysis. Here, behavioural patterns of the mobile phone are monitored by comparing its most recent activities with a history of its usage. Criteria can then be derived to use as triggers that are activated when usage patterns of the mobile phone change significantly over a short period of time. A change in the behaviour pattern of a mobile phone is a common characteristic in nearly all fraud scenarios excluding those committed on initial subscription where there is no behavioural pattern established.

There are many advantages to performing a differential analysis through profiling the behaviour of a user. Firstly, certain behavioural patterns may be considered anomalous for one type of user, and hence potentially indicative of fraud, that are considered acceptable for another. With a differential analysis flexible criteria can be developed that detect any change in usage based on a detailed history profile of user behaviour. This takes fraud detection down to the personal level comparing like with like enabling detection of less obvious frauds that may only be noticed at the personal usage level. An absolute usage system would not detect fraud at this level. In addition, however, because a typical user is not a fraudster, the majority of criteria that would have triggered an alarm in an absolute usage system will be seen as a large change in behaviour in a differential usage system. In this way a differential analysis can be seen as incorporating the absolute approach.

8.5.1.1.2 The differential approach

Most fraud indicators do not become apparent from an individual Toll Ticket. With the possible exception of a velocity trap, confidence can only be gained in detecting a real fraud through investigating a fairly long sequence of Toll Tickets. This is particularly the case when considering more subtle changes in a user's behaviour by performing a differential analysis.

A differential usage system requires information concerning the users history of behaviour plus a more recent sample of the mobile phones activities. An initial approach might be to extract and encode information from Toll Tickets and to store it in record format. This would require two windows or spans over the sequence of transactions for each user. The shorter sequence is called the Current User Profile (CUP) and the longer sequence, the User Profile History (UPH).

Both profiles could be treated and maintained as finite length queues. When a new Toll Ticket arrives for a given user, the oldest entry from the UPH would be discarded and the oldest entry from the CUP would move to the back of the UPH queue. The new record encoded from the incoming Toll Ticket would then join the back of the CUP queue. Clearly it is not optimal to search and retrieve historical information concerning a user's activities prior to each calculation, on receipt of a new Toll Ticket. A more suitable approach is to compute a single cumulative CUP and UPH, for each user, from incoming Toll Tickets that can be stored as individual records, in a database. To maintain the concept of having two different spans over the Toll Tickets without retaining a database record for each Toll Ticket, both profiles need to be decayed before the influence of a new Toll Ticket can be taken into consideration. A profile for each user can then be represented as a probability distribution by normalising the data in the profile.

8.5.1.1.3 Relevant toll ticket data

There are two important requirements for user profiling. At first, efficiency is of the foremost concern for storing the user data and for performing updates. Secondly, user profiles have to realise a precise description of user behaviour to facilitate reliable fraud detection. All the information that the fraud detection tool will use is only derived from the toll tickets provided by the network operator. However using all the fields in

each Toll Ticket would slow the system down and so six fields that are thought to be most useful are extracted from each Toll Ticket. These are;

- Charged_IMSI (identifies the user)
- First_Cell_Id (location characteristic for mobile originating calls)
- Start_Time (time the call was first connected)
- Chargeable_Duration (base for all cost estimations)
- B_Type_of_Number (for distinguishing between national / international calls)
- Non_Charged_Party (the number dialled)

8.5.2 Rule based

Most of today's fraud detection tools are either rule-based or at least comprise a rule-based detection component. A rule-based approach allows detecting the definite frauds with a low rate of false alarms. Moreover, the rule-based tool can easily provide reasons for an alarm being raised.

In addition, a (traditional) rule-based component may support (modern) neural network technologies in several ways. Using Neural Networks makes it easier to filter out exceptions. Exceptions can be suspicious looking cases of definite non-fraud as well as unsuspicious looking cases of non-fraud.

The subsequent sections describe the rule-based approach in more detail. The Protocol Data Analysis Tool (PDAT) has become the central part of the rule based fraud detection tool. Much effort was put into adapting PDAT, whose original purpose was audit trail analysis for UNIX systems, to the new problem of fraud detection. The main tasks were the introduction of user profiles stored in a database and the realisation of a new protocol that allows PDAT to understand both, user profiles as well as toll ticket formats. Once established, PDAT provides a comprehensive infrastructure based on a graphical user interface (GUI) for editing alarm criteria during runtime. The ability of showing alarms is not used within the integrated prototype, since it comes with the more convenient monitoring GUI, which shows aggregated and sorted alarm information for all fraud detection components.

8.5.2.1 User profiling

An essential concept of the rule-based fraud detection tool is *user profiling*. A user profile is a set of features, which may - sometimes in combination with other features - indicate fraudulent behaviour. A user profile is maintained for each subscriber of the mobile network. A continuous flow of toll tickets (TTs) is processed by extracting and aggregating information for each user's specific profile. The integrated prototype is confined to using the six most fraud relevant toll ticket (TT) components, which are:

Charged_IMSI, Charging_Start_Date, Charging_Start_Time, Chargeable_Duration, B_Type_of_Number and Non_Charged_Party.

In most cases telecommunication fraud manifests itself with significant changes in a user's behaviour over a certain time period. Therefore, different user profiles are maintained for different periods of time, a current user profile (CUP) describing the user's short-term behaviour, and a user profile history (UPH) which stands for the long-term behaviour. A comparison between features of the CUP and the UPH gives a differential analysis of user behaviour.

Moreover, user profiling helps the rule-based approach to overcome its most criticised drawback, the inflexibility of one set of rules applied to all users. User profiles allow a far more flexible, user specific treatment. The independently stored profiles allow each subscriber to be observed separately. For even more flexibility, user specific thresholds could be defined within the profile.

A very important quality of the UPH is that it slowly but permanently adapts to the user behaviour in the following way; smooth and long-lasting changes in behaviour - usual for normal behaviour - are by and by adopted by the UPH while significant short-term changes - indicative of fraud - are still detected. In the case

of erratic user behaviour the user profile also adapts to this erratic behaviour and avoids wrong alarms unless the user leaves his normal range of behaviour.

8.5.2.1.1 Current User Profile (CUP)

The number of features maintained in user profiles has been restricted to the most significant ones. Using a high number of statistical variables does not necessarily improve the quality of a fraud tool: If very few events contribute to a certain feature, the statistical reliability of this feature can become bad. The fields of a CUP in the rule-based component are as follows:

- **cup_start** (starting date and time of the CUP)
- **cup_nr_nat** (total number of national calls during time interval)
- **cup_nat_dur** (total duration of national calls during time interval**)**
- **cup_nr_int** (total number of international calls during time interval)
- **cup_int_dur** (total duration of international calls during time interval**)**

In principle, a CUP can be built using two different techniques, a *sliding window* and a *hopping window* technique. The sliding window extends over a fixed time period (e.g. 24 hours). At any time the CUP's values reflect aggregated information out of the recent 24 hours. The advantage of this fixed length time interval is a good statistical significance and reliability of the CUP'S features. However, this has to be paid for by a more complicated and costly implementation. For all users, all toll tickets must be stored for the window's time period. On each incoming TT we must update the set of TTs belonging to the sliding window and must recompute the CUP's values. The hopping window has a fixed starting time and a fixed ending time. It is varying in length from 0 to its maximum length (ending time minus starting time). This technique does not require all toll tickets to be stored for a certain time. The disadvantage of this technique is that the CUP's values lack statistical significance and reliability as long as the window is quite small. To overcome this we can also use a sequence of CUPs (CUP_i, i = 1,...,m) to store detailed user information up to the medium-term past.

The CUP realised in the rule based FDT uses the hopping window technique with 24 hour intervals. This time interval may of course be set to a different value. In order to spread the workload for computing a User Profile History Update (UPH, see below) uniformly, the 24 hour interval may start at different times of the day for different users. For one user, the starting time is always the same. The first prototype realises the sequence (CUP_0, CUP_1) only. CUP_1 is the last but one user profile and is guaranteed to be finished, while CUP_0, the profile of the current day, is generally under construction. Thus the rule-based component works with a hopping window of a minimum length of 24 hours and a maximum length of 48 hours. .

On each incoming toll ticket, the fields of CUP_0 are updated. It is also checked for each incoming toll ticket whether a CUP has completed its life span. If the time difference between the current time of an incoming toll ticket and 'cup_start' is greater or equal the chosen time interval (here 24 hours), but less than twice that interval (48 hours) the following steps are performed:

- update UPH with the sequence (CUP_0, CUP_1) (as described below)
- $CUP_1 := CUP_0$**;**
- compute CUP_0.cup_start ($date_0 = date_1+1$)
- set all other fields of CUP_0 to values of incoming toll ticket;

If the time difference between the current time of an incoming toll ticket and 'cup_start' is between n*interval and (n+1)* interval for n>1 then the above procedure is carried out n times, thereby introducing "intermediate" CUPs with values set to 0 (except for cup_start) for the intermediate intervals with no activity.

8.5.2.1.2 User Profile History (UPH)

The UPH, which represents the long-term behaviour of a subscriber, consists of the following fields:

- **uph_start** (absolute time in seconds, when user record was created)
- **uph_op_mode** (operation mode flag, either ″training″ or″fading″)
- **uph_nr_nat** (long-term average number of national calls during time interval)
- **uph_nr2_nat** (long-term average squared number of national calls during time interval**)**
- **uph_nat_dur** (long-term average duration of national calls during time interval**)**
- **uph_nat_dur2** (long-term average squared duration of national calls during time interval**)**
- **uph_nr_int** (long-term average number of international calls during time interval)
- **uph_nr2_int** (long-term average squared number of international calls during time interval**)**
- **uph_int_dur** (long-term average duration of international calls during time interval**)**
- **uph_int_dur2** (long-term average squared duration of international calls during time interval**)**
- **fraud_probability**

The UPH will be updated whenever a CUP has completed its live span. This is done by decaying the current UPH-values and adding "fresh" data taken from the existing CUPs. Exponential fading is used to compute the components of the UPH:

For the components UPH.x with x = uph_nr_nat, uph_nat_dur, uph_nr_int or uph_int_dur we set

UPH_{new} .x = (1-f_m) UPH_{old} .x + $\Sigma_{i=0,..,m-1} f_i \, CUP_i$ .y ,

with m=2, and fading-factors f_i, $0 \le f_i < 1$, $\Sigma_{i=0,..,m} f_i = 1$ and with y = cup_nr_nat, cup_nat_dur, cup_nr_int or cup_int_dur.

For the components UPH.x with x = uph_nr2_nat, uph_nat_dur2, uph_nr2_int or uph_int_dur2 we set

UPH_{new} .x = (1-f_m) UPH_{old} .x + $\Sigma_{i=0,..,m-1} f_i \, CUP_i^2$ .y ,

with m=2, and fading-factors f_i, $0 \le f_i < 1$, $\Sigma_{i=0,..,m} f_i = 1$ and with and with y = cup_nr_nat, cup_nat_dur, cup_nr_int or cup_int_dur.

Since the long-term behaviour is not dependent on a single toll ticket, the UPH is not updated on each incoming toll ticket. The update is still triggered by a toll ticket, but not before the CUP's time interval (24 hours) is finished. Depending on the arrival of the triggering toll ticket, this may result in a long time between two UPH updates, however, without causing a delay in the recognition of fraud.

8.5.2.2 Rule-based fraud analysis

8.5.2.2.1 Architecture of the rule-based tool

The final architecture of the rule-based tool within the integrated prototype is shown in Figure 8.2 - Architecture of the rule-based part within the integrated prototype. The main change due to the integration is the introduction of a common stream between all fraud engines. For reasons described in D18 the rule-based tool is placed best at the end of the sequence of all tools. The tool is now able to operate on toll tickets that have been attributed by the other tools and may be named ATT's (attributed TT's). This way, additional information provided by the other tools such as results of the B-number analysis can and will be used.

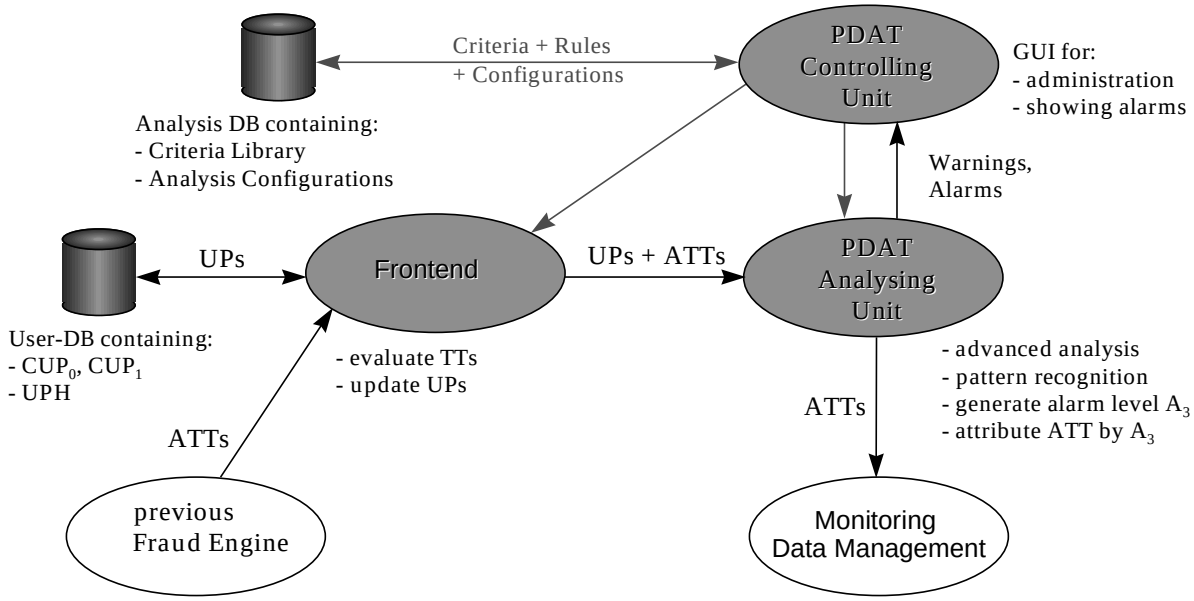


Figure 8.2 - Architecture of the rule-based part within the integrated prototype

As described in D08, the Protocol Data Analysis Tool (PDAT) provides a comprehensive infrastructure based on a graphical user interface (GUI) for showing alarms and for editing alarm criteria during runtime.

Within the integrated prototype the part of the GUI showing alarms will be of less importance, since a more user friendly ranking of suspicious users is shown in the monitoring GUI now. A direct comparison with the other tools' results and an adjustable combination of these results will be shown as well.

The administration part of the PDAT GUI, however, is still the essential part for editing rules or changing complete rule configurations.

8.5.2.2.2 Differential analysis

The differential analysis is invoked on several occasions:

1.) during each UPH-update

2.) if the absolute analysis found something strange, which caused a message at least.

3.) if any rule dictates an explicit differential analysis

Part of the differential analysis is comparing the fields of the CUP and the UPH in pairs and check the difference against several thresholds. These thresholds will be expressed using the mean *M* and standard deviation σ of the entries of the CUPs, i.e. of the user's day-to-day behaviour.

They can be estimated by the entries of UPH as follows:

set M = **uph_nr_nat,** set σ^2 = **uph_nr2_nat - (uph_nr_nat)^2 .**

(similarly for the other entries of UPH).

The GDBM database, which has proven to be a very fast database and the restriction to six TT fields guarantee for fulfilling the network operator's performance needs. If future requirements ask for even more performance, this can be achieved by realising a swapping technique. Note also, that the fraud detection problem can by its nature easily be parallelised.

8.5.2.2.3 Absolute analysis

The purpose of an absolute analysis is to generate alarms at a too high level of calling behaviour. A fraud tool that only uses differential analysis may easily be hoaxed by smart fraudsters using trail and error. They will quickly find out to what extend changes over a certain time period remain undetected and can steadily increase their traffic without limitation. This is clearly prevented by an absolute analysis.

8.5.2.2.4 Determining the fraud probability

The rule-based component contains roughly three categories of rules, which are rules implementing

- absolute analysis
- differential analysis
- a combination absolute/differential with lower thresholds

For evaluation purposes a common alarm level A across all rules is introduced. Consider i rules of the form: "if value V_i > threshold T_i, then raise an alarm".

Then, the common alarm level A can be defined as: $A_{com} = \max (V_i / T_i)$, for all rules i. Now, the alarms will raise by one meta-rule "if value A_{com} > threshold T_{com}, then raise an alarm". By varying T_{com} we can adjust the tool's basic setting (e.g. get as much fraudsters as possible or avoid as much false alarms as possible). However, this value is a real number and not a fraud probability. For the integrated prototype we are mapping the alarm level A_{com} to a fraud probability using the tangens hyperbolicus: $P = \tanh (c A_{com})$. A constant c is used to standardise the alarm level between the other fraud engines so that they are comparable and can be used for ordering and common evaluations

The applied rules only focus on the user behaviour determined by the most important Toll Ticket features as identified in prior documents (duration and number of national/international calls mainly). Since there are no cases of cloning in GSM known to the authors, overlapping call checks or velocity checks are not applied. The rules had been slightly adjusted in relative strength among each other with the full batch of 75000 new subscribers. We did no adjustment based on the fraudulent data to avoid overfitting.

8.5.2.3 The administration GUI

8.5.2.3.1 PDAL language concepts

Important goals were flexibility and broad applicability, including the analysis of general protocol data, which is achieved by the special language PDAL (Protocol Data Analysis Language). PDAL allows the programming of analysis criteria as well as a GUI-aided configuration of the analysis at runtime. The language's concept is related to AWK, but is enhanced with especially the following:

- A **protocol format** describes the possible content of the several data records. The supported data types are int, float, string and time-stamp. The current protocol format describing toll tickets and user ticket allows PDAL to refer to the values as base for the following features.
- An **analysis criterion** formulates a condition on single data records or on a pattern of several data records. There are three types of analysis criteria (in growing complexity): filter, criterion, and behaviour.
- A **rule** assigns a specific action to an analysis criterion. If the criterion is fulfilled, this action will be performed. Typical actions are warnings or alarms shown at the GUI or a certain system call.
- An **analysis configuration** consists of a list of rules.
- Except the protocol format of course, all of these concepts can be changed during runtime. The threshold of a filter, or a tree of filters a criterion consists of, or rules may be added to or deleted from a analysis configuration.
- **Dynamic tables** are a means for recognising patterns of information across several data records. They allow storing intermediate results in associative arrays, which can be indexed by every data type (string, int, float, ...).

8.5.2.3.2 GUI features

PDAT is designed in a manager/agent-architecture, with the Controlling Unit (CU) as manager and the Analysing Unit (AU) playing the agent's role. The essential part of the Controlling Unit (CU) is a GUI, that supports all the administration tasks, which are configuration of analysis criteria and rules, controlling the

AU and displaying all warnings and alarms received from the AU. Figure 8.3 shows the GUI of PDAT during operation.

The PDAT Desktop is the root window of the GUI, which shows a menu list at the top, an alarm display, and additional status information and statistics at the right side. The menu list provides the following features:

PDAT	Control of the CU
Host	Control of the selected AU. In general we can deal with several AUs, possible operations are: connect, disconnect, transfer (of a change configuration) and kill.
Report	Detailed description of alarms concerning one or all criteria
Configuration	Edit an analysis configuration (set of rules): enable/disable certain rules, load/save set of rule
Critbase	Edit the criterion base (analysis criteria): edit certain criterion, load/save set of criteria

The status information shows which configuration is currently active ('up': user profiles) and to which AU represented by its host we are connected. Statistical information consists of how many user records have been processed and how many events have been transferred to the CU. All alarms and warnings are listed in the main display part. Additionally, the PDAT icon represents the severity of the current event by its colour. The icon looks grey at the beginning, changes to yellow for a warning or to red for an alarm and goes back to grey after clicking the icon for acknowledgement.

The 'Configuration Rules' window gives an impression of how rules are dealt with. Firstly, a new rule can be introduced by using the GUI as an editor. By clicking the 'Check' button a PDAL syntax check is applied to the new rule. Secondly, we can simply change a criterion (e.g. change a threshold) as well as the concerning rule (e.g. change the issued action). Finally, rules can be added to or deleted from the current set of applied criteria.

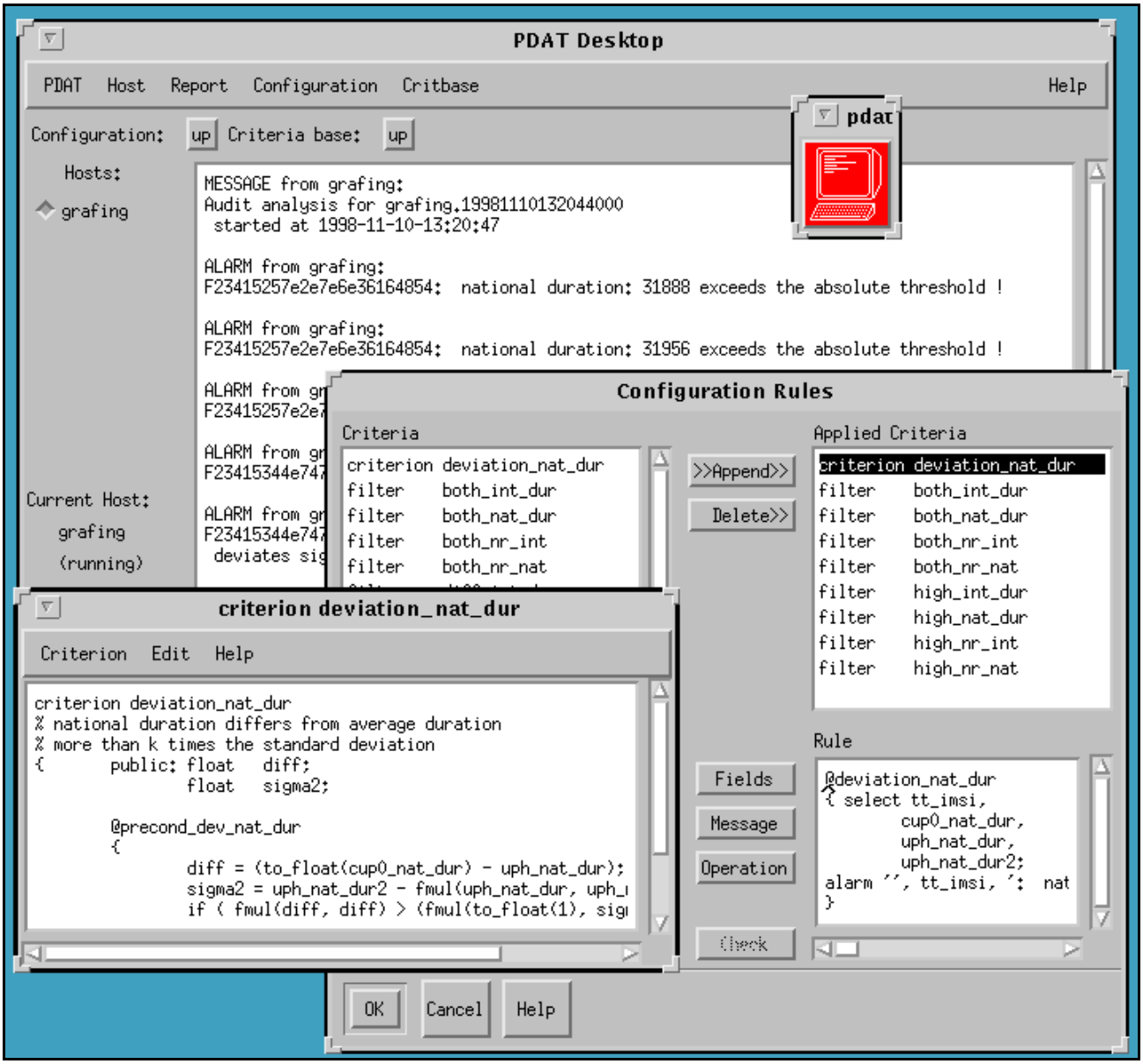


Figure 8.3 - Administration GUI of the Rule-based Fraud Detection Component

8.5.3 Supervised neural network

The general architecture for the neural network approach to fraud detection is described in previous deliverables and we refer the reader to these documents for a first presentation of the fraud detection tool and of the real-time environment. We describe here the specifics of the implementation of the fraud detection tool. That is, how this tool extracts the User Profile Record, the Current User Profile, and the User Profile History from sequences of toll tickets; how it extracts the relevant features from these profiles; and how the neural network processes these features to produce its decision.

8.5.3.1 Profiling

8.5.3.1.1 User Profile Record (UPR)

The six fields that the mediation device simulator provides to the system are the TT_IMSI, TT_CHARGING_START_DATE, TT_CHARGING_START_TIME, TT_NON_CHARGED_PARTY, TT_B_TYPE_OF_NUMBER. This information is stored in the User Profile Record. However, we obtain the Current User Profile and User Profile History by using a filtering technique on the User Profile Record, we therefore need to translate the TT_CHARGING_START_DATE and TT_CHARGING_START_TIME

to numerical values. The mediation device simulator converts thus the TT_CHARGING_START_DATE to the number of days from some reference date using a calendar and the TT_CHARGING_START_TIME to the number of seconds from midnight. This way, the absolute time of beginning of the call is equal to (in seconds from the reference date on midnight) TT_CHARGING_START_DATE * 86400 + TT_CHARGING_START_TIME.

8.5.3.1.2 Current User Profile (CUP)

The Current User Profile contains the information about the short-term behaviour of the user. We have decided to focus on the following measures of the behaviour of the user: the duration of calls and the interval between calls. High call duration, and low call interval (high call frequency) are indicators of, for example, call selling; while low call duration and low call interval are indicators of a PABX attack. We further split the call duration and the call interval between national and international calls. Furthermore, we do not only compute the mean duration of calls and the mean interval between calls, but also the variance of the call duration and of the call interval.

If we work with the absolute time of a call, we have the problem that calls tend to have very high intervals during the night, and low intervals during peak hours. To compensate for this, we determine the daily distribution of the calls and re-parameterise time so that the activity is equally distributed. The result is that the "time" difference between 10 p.m. and 6 a.m., may be equal to the difference between 10 a.m. and 10:30 a.m., because the amount of activity during the two periods is equal. The absolute time of the last national call *tn*, of the last international call *ti*, and of the last other call, have to be part of the profile to compute time differences. The short-term averages are computed using a first order filter. At each toll ticket *t*, we can compute the average $<x(t)>\alpha$ of a quantity *x(t)* over all previous toll tickets as follows:

$$< x(t) > \alpha = \begin{cases} (1-\alpha) < x(t-1) > \alpha + \alpha . x(t) & \text{if } x(t) \text{ is defined} \\ < x(t-1) > \alpha & \text{if } x(t) \text{ is not defined} \end{cases}$$

It is important to note that the quantity *x(t)* might not be defined at every toll ticket; for example, only one of duration of national call *dn*, duration of international call *di*, and duration of other call do will be defined at a time. The filter is, in fact, a first-order low-pass filter, and it gives thus an estimate of the mean *E(x)* of the signal *x(t)* (if the signal is a sequence of independently identically distributed random variables). Further, the filter can track changes in the mean. So, the short-term average of the duration of national calls will be $<dn>_a$ in our notation. To compute a short-term standard deviation of a quantity *x(t),* we can still use similar filters but on $x(t)^2$. We derive this from the definition of the variance as follows ($\hat{\mu}, \hat{\sigma}$ being estimates of the mean μ and standard deviation σ):

$$\mu = E(x)$$

$$\sigma^2 = E((x-\mu)^2) \overset{i.i.d.}{=} E(x^2) - (E(x))^2$$

$$\hat{\mu} = < x >_\alpha$$

$$\hat{\sigma} = \sqrt{\max(< x^2 >_\alpha - (< x >_\alpha)^2, 0)}$$

This finally gives the following 10 fields for the Current User Profile:

- Absolute time of last national call t_n
- Absolute time of last international call t_i
- Short-term average of the duration of national calls $<d_n>_\alpha$
- Short-term average of the duration of international calls $<d_i>_\alpha$
- Short-term average of the squared duration of national calls $<(d_n)^2>_\alpha$
- Short-term average of the squared duration of international calls $<(d_i)^2>_\alpha$

- Short-term average of the call interval between national calls $<i_n>_\alpha$
- Short-term average of the call interval between international calls $<i_i>_\alpha$
- Short-term average of the squared call interval between national calls $<(i_n)^2>_\alpha$
- Short-term average of the squared call interval between international calls $<(i_i)^2>_\alpha$

8.5.3.1.3 User Profile History (UPH)

We derive the User Profile History in a similar fashion by filtering the Current User Profile with a first-order filter with parameter β. This means that the User Profile History is, in fact, a second-order filter of the signals. It thus estimates average quantities, but on a longer time scale than the Current User Profile. Processing the call duration and call interval through a first-order filter to obtain the Current User Profile, and again through another first-order filter to obtain the User Profile History minimises the memory requirements for the updates of the filter, and therefore minimises the load on the database of user profiles. Furthermore, the difference between the User Profile History and the Current User Profile can be interpreted as a second-order band-pass filter on call duration and call interval. This means that it is affected neither by very short-term variations (let us say, between one call and the next) neither by very long-term variations (therefore allowing us to track long-term changes in the behaviour of the user). The difference between the User Profile History and the Current User Profile allows us to detect deviations from the normal behaviour of a user. The date of first call is also kept in the profile to determine at which point differential analysis becomes applicable (since, at the beginning, the User Profile History does not contain any relevant information). These considerations results in the following User Profile History (using the same notation as in the previous paragraph).

- Date of first call
- Long-term average duration of national calls $<<d_n>_\alpha>_\beta$
- Long-term average duration of international calls $<<d_i>_\alpha>_\beta$
- Long-term average squared duration of national calls $<<(d_n)^2>_\alpha>_\beta$
- Long-term average squared duration of international calls $<<(d_i)^2>_\alpha>_\beta$
- Long-term average call interval between national calls $<<i_n>_\alpha>_\beta$
- Long-term average call interval between international calls $<<i_i>_\alpha>_\beta$
- Long-term average squared call interval between national calls $<<(i_n)^2>_\alpha>_\beta$
- Long-term average squared call interval between international calls $<<(i_i)^2>_\alpha>_\beta$

8.5.3.2 Feature extraction

The features used by the classifier will not be the content of the Current User Profile and User Profile History directly, but estimates of means and standard deviations. The means are obtained directly at the output of the filters, but the standard deviations must be computed as $\hat{\sigma} = \sqrt{\max(< x^2 > -(< x >)^2, 0)}$, where the bracket denotes a first-order or a second-order filter. This results in the following vector of features, which is the input to the classifier.

- Number of days since first activity
- Short-term mean of the duration of national calls
- Short-term mean of the duration of international calls
- Short-term standard deviation of the duration of national calls
- Short-term standard deviation of the duration of international calls

- Short-term mean of the call interval between national calls
- Short-term mean of the call interval between international calls
- Short-term standard deviation of the call interval between national calls
- Short-term standard deviation of the call interval between international calls
- Long-term mean of the duration of national calls
- Long-term mean of the duration of international calls
- Long-term standard deviation of the duration of national calls
- Long-term standard deviation of the duration of international calls
- Long-term mean of the call interval between national calls
- Long-term mean of the call interval between international calls
- Long-term standard deviation of the call interval between national calls
- Long-term standard deviation of the call interval between international calls

8.5.3.3 Storing user profiles

User profiles must be swapped between disk and main memory each time a new toll ticket arrives at the fraud detection tool. The performance requirements are severe, as peak performance must exceed 30 toll tickets per second. We obtain such performance by using a simple, but optimised database tool called GDBM. The database is accessed by a key, which is the IMSI of the user and provides a content, which is the concatenation of the Current User Profile and User Profile History.

8.5.3.4 Supervised learning

After designing the front-end, we must design the classifier. The front-end processes the toll tickets to produce sequences of user profiles; and then extracts the features needed by the classifier from these profiles. The classifier then maps a vector of features to an alarm value between 0 and 1 using a multilayer perceptron.

8.5.3.4.1 Multilayer perceptron

The neural network used in the fraud detection engine is a multilayer perceptron. It is defined as follows. The network is composed of elementary units called neurons. Each neuron produces at its output a simple non-linear transformation of its inputs depending on the value of the weights of the network:

$$y = \sigma(\sum_{i=1}^{n} w_i x_i + w_0), \quad \text{where } \sigma(z) = \tanh(z) \text{ or } \sigma(z) = \frac{1}{1+e^{-z}}.$$

The neurons are then arranged in a two-hidden-layer network with D inputs, H_1 hidden neurons in the first layer, H_2 hidden neurons in the second layer, and C outputs. The outputs z_m of the network can then be defined as

$$h_{1_k} = \sigma(\sum_{l=1}^{D} w_{kl} x_l + w_{k0})$$

$$h_{2_l} = \sigma(\sum_{k=1}^{H_1} v_{ik} h_{1_k} + v_{i0})$$

$$z_m = \sigma(\sum_{l=1}^{H_2} u_{lm} h_{2_l} + u_{l0})$$

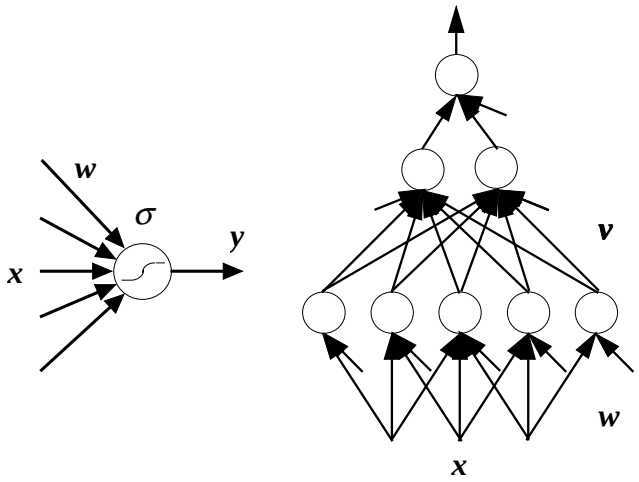


Figure 8.4 - Sigmoidal neuron and multilayer perceptron architecture

The main property of multilayer perceptrons is that they can approximate any function of the input to an arbitrary degree of accuracy, provided that enough hidden neurons are available. They can achieve this approximation with a relatively small number of parameters.

8.5.3.4.2 Labelling

For supervised learning, we organise the data available for design in a data set of labelled pairs $D = \{(X_1,Y_1),...,(X_K,Y_K)\}$, where Y_k is the fraud label ($Y_k = 0$ for normal behaviour, $Y_k = 1$ for fraud) associated to the k-th pattern with features X_k extracted from the user profile. The training data consists for the first part of the calls made by 300 users from the two-month download from Vodafone; these users are deemed normal. It also consists for a second part of the calls made by 300 fraudulent users. For all 600 users, all the available toll tickets are processed through the front-end of the system to produce sequences of user profiles. We label the sequences of user profiles for the normal users as non-fraudulent. For the fraudsters, we studied the evolution of the profiles over time to determine the beginning of the fraudulent behaviour; and we labelled the profiles as fraudulent during the fraudulent behaviour and as non-fraudulent otherwise.

8.5.3.4.3 Training

The first step is to choose the architecture of the neural network, that is the number of layers, and the number of neurons in each layer. Once we have chosen the architecture, the output of the network is a function of its input X_k and of the parameters w (the weights) of the neural network. There is a discrepancy between the output of the classifier $z(X_k,w)$ and the desired output Y_k. The learning of training of the classifier consists in adapting the weights so as to minimise this discrepancy. The measure of discrepancy is quadratic.

find w that minimizes $\mathrm{E} = \sum_{k=1}^{\mathrm{K}} \left\| \mathrm{Y_k} - z(X_k, w) \right\|^2$.

We achieve this minimisation using a gradient-descent method, namely the Levenberg-Marquardt algorithm. This powerful method is based on algebraic procedures, which permits, given a value of the weights, determination of the modification of the weights that would lead to an optimal reduction of the error. After we have modified the weights, we have a smaller error and a new value of the weights. A new correction to the weights is evaluated, so as to reduce the error as rapidly as possible. We repeat this procedure until the error ceases to decrease.

8.5.3.4.4 Cross-validation

We split the data set into three subsets: the training set, the validation set, and the test set. In order to maximise the performance on previously unseen data we use the following procedure called cross-validation. The weights are adapted by minimising the error on the training set, but we observe the error on

the validation set during this process; and we stop the minimisation when the error on the validation set reaches a minimum. We then estimate the expected performance on new data by computing the error on the test set.

8.5.3.4.5 Determination of the optimal architecture

We determine the optimal weights using the error minimisation procedure, but we have to repeat the procedure to search for a global optimum of the optimisation procedure, since gradient-descent methods are only guaranteed to converge to a local optimum. Furthermore, we have to repeat this procedures for different architectures of the neural network to determine the optimal one. Once we have found the optimal neural network, we simply have to use it on top of the front-end and it will produce an alarm value between 0 and 1 each time a toll ticket is presented to the fraud detection tool.

Within the trial, the supervised tool will also use the alarm level of the unsupervised tool as a form of a priori weighting in its training phase and later on in its alarm generation.

8.5.4 Unsupervised learning tool

When considering the task of detecting fraudulent activity in mobile telecommunications networks, the challenge is to find a suitable representation of Toll Ticket data to summarise mobile phone usage and form user behaviour profiles. The format of these profiles should suit the needs of the specific fraud engine. This section considers a profiling technique derived from Unsupervised Learning in Neural Networks. User profiles will be generated automatically by the classification of GSM Toll Tickets into one of a set of Toll Ticket prototypes. As Toll Tickets are classified, statistical information pertaining to the number of times a Toll Ticket prototype is excited, by the presentation of an incoming Toll Ticket, will be computed and stored in the form of a record, the length of which will be equal to the number of prototypes.

By considering two different time spans over the Toll Tickets, two profile records for each user can be generated. The profile representing the shorter Toll Ticket span can be considered as representing the user's most recent activity. This is called the Current User Profile (CUP). The longer span will create the User Profile History (UPH). Training can be seen as the presentation of clean profiles to the system to define the boundaries of acceptable behaviour, based on a differential analysis. The task of the system, after training, is to raise alarms when it is presented with profiles where the difference between the CUP and the UPH is outside the realms of normal usage. An alert status will be raised if the profiles are significantly different.

This system requires only clean (non-fraudulent) data for training. This is advantageous because the fraudster has not yet caught up with GSM technology and few fraud scenarios are seen in practice. In addition, this system has the potential to detect new types of fraud as and when they occur. The system is able to do this because it is not being trained to recognise specific fraud scenarios, but rather altered or unusual usage.

Determining the set of prototypes which are used as the features for forming the user profile is analogous to choosing the cones and rods in the eye, which determine what information can be registered and hence what types of object can be detected in the environment. The aim has been not only to find a set of features that works with the current data, but to develop design principles for altering the `focus' of attention of the system to allow novel fraud to be more closely monitored as our awareness of its salient features develops. This type of meta-learning has been referred to as an adaptive critic, meaning that there are two levels of learning taking place, the basic level of comparing different profiles and the meta-level of adapting the features used to form the profiles themselves. This adaptation must occur off-line in response to information acquired about fraudulent activity over a longer timescale.

8.5.4.1 Prototyping

Prototyping is a method of forming an optimal discrete representation of a naturally continuous random variable. The processing of continuous random variables by discrete systems generally reduces empirical information. Neural Networks are capable of forming optimal discrete representations of continuous random variables through their ability to converge, by lateral interaction, to stable uniformly distributed states.

The maximum-entropy principal states that; *the mapping of a continuous random variable X into a set of K discrete prototypes Q reduces the empirical information by the least amount if a uniform distribution* { $\mathbf{P}(q_i) = \frac{1}{k}, i = 1...K$ }, *corresponding to the absolute maximum (* $S_Q = \log K$*) of information entropy, is assigned to Q.*

In [REF], Grabec provides a way to extend this principal to multiple dimensions. When considering the set of all possible Toll Tickets, a dimension to represent every parameter that needs to be included in the analysis is required. Each parameter in a Toll Ticket can assume a range of values and is thus itself a random variable. Grabec's technique allows a number of prototypes to be created that dynamically and uniformly span the set of samples taken from the space of possible Toll Tickets. Using this technique enables each incoming Toll Ticket to be classified to the prototype that most closely resembles all its characteristics.

A user profile can then be constructed as a vector of counters representing the number of times each prototype has been excited by the presentation of an incoming Toll Ticket, for that user. In order to maintain the notion of a time-span over the toll tickets, a decay factor is applied to the vector of counters prior to incorporating information from any incoming Toll Ticket. By using two different decay factors, profiles can be maintained representing the two different time-spans over the Toll Tickets, namely the CUP and the UPH.

In order to generate a set of condensed Toll Ticket prototypes, a non-fraudulent data set of Toll Tickets is required. An iterative procedure is set up to dynamically distribute prototypes over this Toll Ticket sub-space through sampling the incoming stream of condensed Toll Tickets.

This can be demonstrated by defining the procedure for a one-dimensional space, i.e. considering a condensed Toll Ticket consisting of only one parameter instead of six.

First, the iterative procedure to calculate the change in the current value of the K prototypes Q is defined;

$$\Delta q_l^{(i+1)} = B_l - \sum_{k \neq l} C_{lk} \Delta q_k^{(i)} \qquad ; l = 1...K \tag{8.1}$$

starting with $\Delta q_l^{(0)} = B_l$. We define matrix **C** as;

$$C_{lk} = \left[1 - \frac{(q_l - q_k)^2}{2\sigma^2}\right] \exp\left[\frac{-(q_l - q_k)^2}{4\sigma^2}\right] \tag{8.2}$$

Where $\sigma \approx \frac{S}{K}$ approximates the standard deviation, and *S* is the expected range of X. In practice, the results are not particularly sensitive to the choice of σ and so a broad estimation of the range *S* of X will suffice.

Vector **B** is defined as

$$B_l = \frac{K}{N+1}\left\{(X_{N+1} - q_l)\exp\left[\frac{-(X_{N+1} - q_l)^2}{4\sigma^2}\right] - \frac{1}{K}\sum_{k=1}^{K}(q_k - q_l)\exp\left[\frac{-(q_k - q_l)^2}{4\sigma^2}\right]\right\} \tag{8.3}$$

Starting with $q_0(k) = X(k)$; $k = 1,...K$. We calculate changes in Q on sampling a new condensed Toll Ticket from X. Using (8.1) we iterate, calculating a vector of changes ΔQ based on the new Toll Ticket. Experimental results have shown that four iterations of (8.1) will be sufficient to stabilise the vector. The current vector of prototypes Q can then be updated by ΔQ.

The above procedure extends to the multidimensional case where condensed Toll Tickets consist of more than one parameter. The vector of prototypes becomes a vector of 'vector prototypes' - in other words a matrix. The process is repeated, by sampling condensed Toll Tickets until the matrix $\Delta Q < \varepsilon$, an arbitrary stability criterion. When this condition holds over a defined number of incoming condensed Toll Tickets, the matrix of prototypes can be considered to uniformly span the condensed Toll Ticket space.

For the Unsupervised Neural Network tool the number of prototypes *K* was set at 110. 50 prototypes for national and international calls respectively, and 10 for other services such as VoiceMail and SMS messaging. Four parameters are considered from each Toll Ticket, namely TT-CHARGED-IMSI, TT-CHARGING-START-TIME, TT-CHARGEABLE-DURATION and TT-B-TYPE-OF-NUMBER. These are calculated and can be plotted on a graph as start time versus call duration. Figure 8.5 and Figure 8.7 show the generated prototypes for national and international calls, while Figure 8.6 and Figure 8.8 show 50,000 data points of Toll Ticket data to show how the prototypes span the real data set.

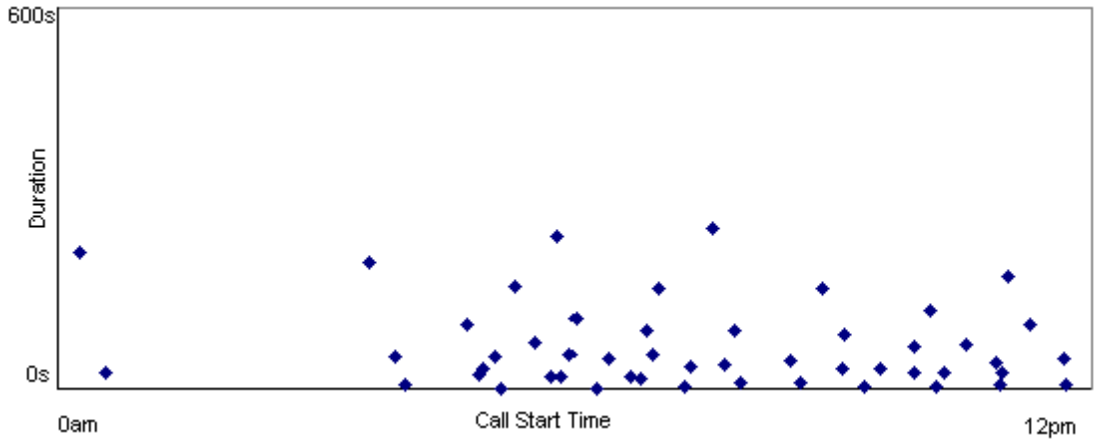


Figure 8.5 - 50 Prototypes for national calls

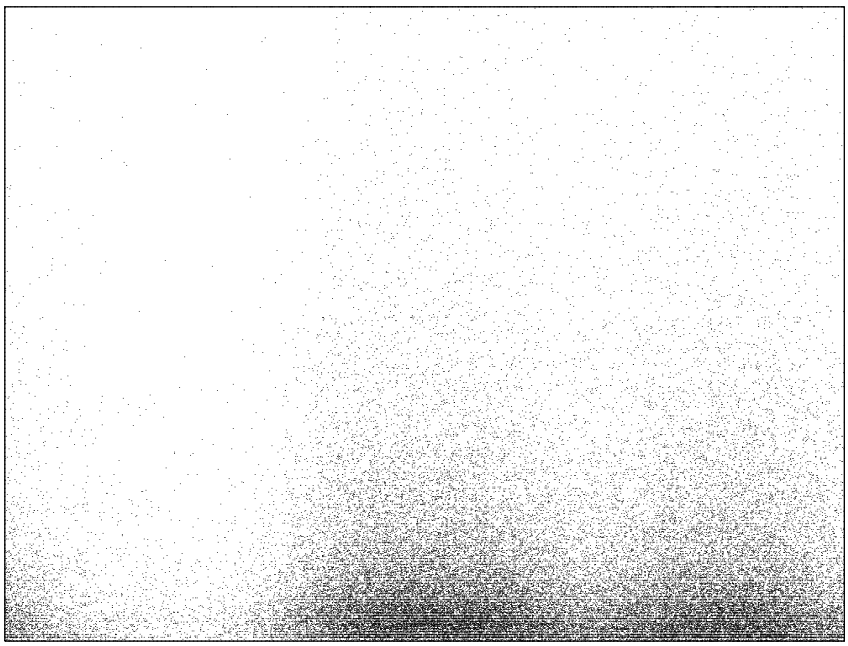


Figure 8.6 - 50,000 National calls to Neural Network prototyper

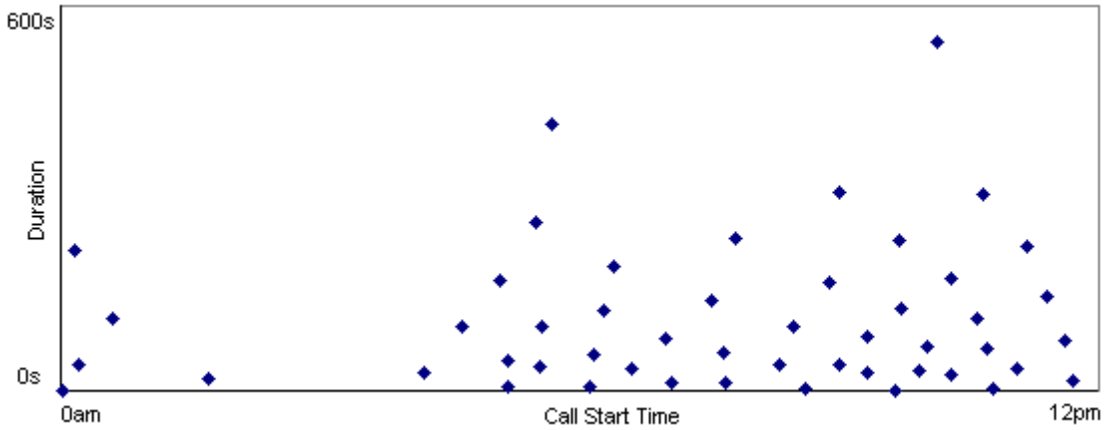


Figure 8.7 - 50 Prototypes for international calls

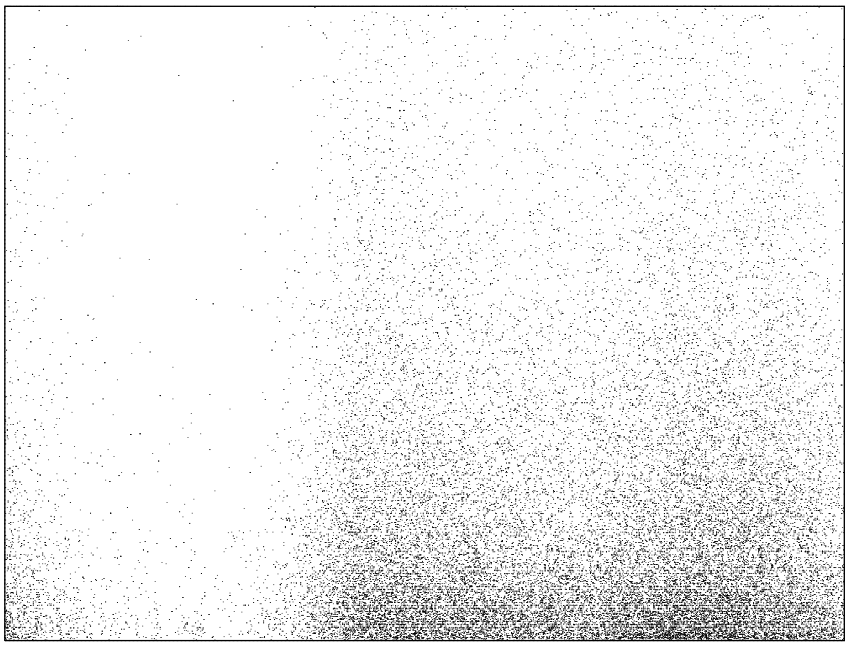


Figure 8.8 - 50,000 International calls to Neural Network prototyper

8.5.4.2 Constructing a statistical user profile for each user

Once a matrix of condensed Toll Ticket prototypes has been constructed (this could be considered the learning phase of the network), the user profiles can be built. As condensed Toll Tickets arrive for a user, each one is classified by one of the prototypes. A vector of counters keeps track of the number of times each prototype has been excited by an incoming condensed Toll Ticket, for that user. This vector of counters becomes the user's profile and is stored in a database.

8.5.4.3 Maintaining current and history profiles as probability distributions

As mentioned above, two different time-spans (the CUP and UPH) over the incoming Toll Tickets for each user have to be stored in able to perform a differential analysis on them. These two profiles are stored as probability distributions using two different decay factors α and β to maintain the concept of two time-spans over the Toll Tickets.

When a Toll Ticket is presented to the system to update a user's CUP, each element of the CUP is multiplied by decay factor α. The entry in the profile corresponding to the prototype i, that was excited by the presentation of the incoming condensed Toll Ticket, is then incremented by an amount $(1-\alpha)$.

Updating the CUP in this manner will maintain the profile as a probability distribution. After updating the CUP, both profiles are presented to the fraud engine as discussed in the next section. Following presentation to the fraud engine, the UPH is updated using

$$H_i = \beta H_i + (1-\beta)C_i$$

Where H_i and C_i represent the *i*th element of the UPH and CUP respectively.

The exact value of the two decay factors α and β are likely to be critical for a successful fraud detection tool. Within the scope of the project they were set to 0.9 and 0.98 respectively.

By applying a multiplicative decay factor, each counter in the profile, corresponding to a prototype, once excited will never actually decay to zero. This is important because if a particular behavioural event occurs very infrequently, such as an overseas call for some users, this could be seen as a behavioural anomaly if the profile entry corresponding to it was zero.

8.5.4.4 The fraud engine

The fraud engine operates in two modes, a training mode and a detection mode. In the training phase, clean user profiles, in the form of probability distributions, will be presented to the fraud engine. It is the task of the fraud engine to determine the difference between these distributions.

To perform this task, a measure known as the Hellinger distance shown below in equation (8.4) is used.

$$d = \sum_{l=0}^{K} \left(\sqrt{C_i} - \sqrt{H_i}\right)^2 \qquad (8.4)$$

Where **C** and **H** are the CUP and UPH respectively and *K* is the number of entries in the profile record. The Hellinger distance will always be a value between zero and two where zero is for equal distributions and two represents orthogonality. The Hellinger distance can be seen as a measure of how erratic the behaviour is.

As clean profiles are passed to the fraud engine, in training mode, the maximum value of d that was found ($d_{threshold}$) is determined. This value represents the greatest difference found between a CUP and UPH and shows the most erratic behaviour seen in the clean training set of user profiles.

In detection mode the fraud engine again calculates d according to equation (8.4). If the resultant value of d is greater than the threshold value computed in training then an alert status is raised proportional to $\left|d - d_{threshold}\right|$.

Figure 8.9 shows a CUP and UPH for a user who is considered to have acceptable behaviour. I.e. the Neural Network has not raised an alarm. Figure 8.10 shows a CUP and UPH for a user who has raised an alarm. Here the alarm has been raised due to a sudden drop in the activity of the VoiceMail service. This scenario could occur if the handset has been stolen. The thief is very unlikely to use supplementary services such as VoiceMail!

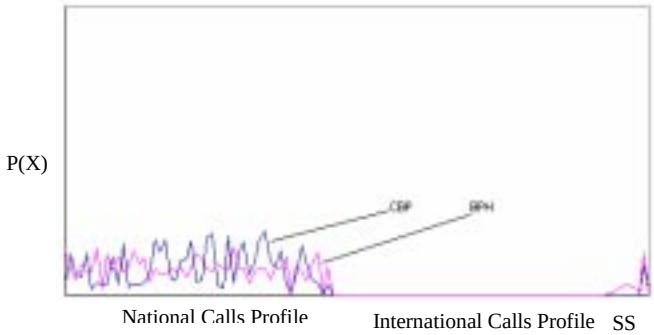


Figure 8.9 - A CUP and an UPH of a subscriber exhibiting acceptable behaviour

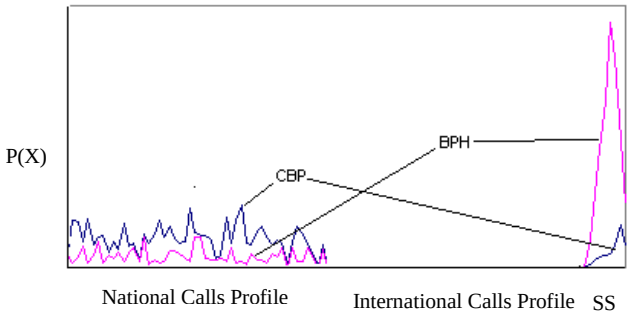


Figure 8.10 - A CUP and an UPH of a subscriber who raised an alarm.

8.5.5 B – Number analysis tool

8.5.5.1 Introduction

One of the most common and expensive types of fraud, is Subscription Fraud. This will normally involve a subscriber using a false identity to purchase as many phones as he can in order to sell them, or the airtime, to people wishing to make cheap international calls. Most fraudulent call destinations are to the Indian or African sub-continent.

This section discusses the development of a B-number analysis tool that monitors the destinations of calls on a per subscriber basis. The destinations of calls (the B-Number) are weighted differently so that well known destinations for fraudulent calls can be given special attention. The tool uses an Unsupervised Neural Network technique as described in the above section.

Note that, international calls amount to approximately 10% of the total call distribution and so, ten times the quantity of data is required to test this tool to the same extent as the A-number analysis (Unsupervised Neural Network tool). It would also be desirable to perform a national B-number analysis. This should enable us to target drug rings and prostitution rackets that tend to use fraudulent mobiles for their criminal activity. It would also be possible to take this a stage further and consider geographical information and weight calls that are destined for cell sites with a reputation for attracting fraudulent activity. This has not been implemented in the ASPeCT work.

8.5.5.2 Adding tags to toll ticket fields

The B-number analysis tool receives Toll Tickets in the cut down form of six fields that have been developed for ease of viewing and manipulation. The six fields passed to the B-number analysis are the IMSI, Charging Start Date, Charging Start Time, Chargeable Duration, B-number and B-type of the call. The first task of the B-number analysis tool is to reformat these fields adding the tags that were agreed to identify each field.

The tag identifiers are TMSI for IMSI, TCSD for charge start date, TCST for charge start time, TCDR for chargeable duration, TBNB for the B-number and TBTP for the B-type of number. To the end of this is appended the current alarm value for the subscriber in the BALM field. The B signifies to subsequent processes, receiving the modified Toll Tickets, that the field was generated by the B-number analysis tool.

8.5.5.3 Dividing the world into fraud risk categories

All the countries of the world were grouped together that belonged to the same geographical area or, countries which have strong economic bonds with each other. It is assumed that people tend to have more contacts with people of neighbouring countries or, for business reasons, with their economic partners. This leads to the implementation of 10 different classes corresponding roughly to the following regions: North

America, Africa, South America, Australia, Asia, Russia, East Block, European Community, Middle East and Central Asia.

During runtime, for each Toll Ticket related to an international call, the country code is extracted (the country code is kept unsanitised). Note, to make a correct classification, we sometimes have to distinguish between dialling codes such as 00-353, which belongs to Ireland and should be assigned to the European Community class, and 00-355 which belongs to Albania and should be assigned to the East-Block class.

8.5.5.4 B-number profiling

As a Toll Ticket arrives for a user, the tool first determines if the call made concerns an international destination, and applies the international analysis if necessary. Each time an international call is made the Toll Ticket is assigned to one of the classes it belongs to and a vector of counters keeps track of the number of times each class has been excited by an incoming Toll Ticket. By considering two different time-spans over the toll tickets, we generate two profile records for each user. The profile representing the shorter Toll Ticket span represents the user's most recent activity and is called the CUP while the longer span represents the user's history of usage and is called the UPH. The two profiles are maintained as probability distributions using two different decay factors α and β, both between 0 and 1. When a new Toll Ticket arrives, the user's CUP is updated. Each element of the CUP is multiplied by the factor α and the class to which the incoming Toll Ticket belongs is incremented by a factor $1-\alpha$.

The update rules for the CUP are thus:

$$CUP_{i\,new} = CUP_{i_{old}} * \alpha \quad \text{for } i \neq k$$

$$CUP_{k_{new}} = CUP_{k_{old}} * \alpha + (1-\alpha) \quad \text{for } i = k$$

with *k* being the number of the class to which the Toll Ticket belongs and *i* referring to the index of each class. This profiling method is identical to the profiling described in the previous section for the general Unsupervised Neural Network tool (which performs an A-Number analysis).

By assigning a 1 to the class which the Toll Ticket belongs to, the first time an international call is made, and using this updating technique, the profile is maintained as a probability distribution function. After updating the CUP both profiles are presented to the fraud engine that will determine the alarm level. It is necessary to allow both profiles to develop adequately for each user before considering the alarm level as evidence that anomalous behaviour is occurring. Following presentation to the fraud engine, the UPH is updated by incorporating information from the CUP to it and by using a decay factor of β.

The update rule used for the UPH is:

$$UPH_{i_{new}} = UPH_{i_{old}} * \beta + (1-\beta) * CUP_i,$$

where *i* refers to the index of each class. So as not to increase unnecessarily the overhead on the system, the same α and β factors as the one used in the A-number analysis.

8.5.5.5 The fraud engine

The fraud engine takes the B-number profile record consisting of the CUP and UPH as input and calculates a *modified* Hellinger distance over all the entries of the profile record. Each entry in the Hellinger distance is weighted by a factor depending on how frequently this class is called over all the users. More importance can be attached to changes occurring to classes that a genuine subscriber rarely calls while minimising the influence of changes to classes that are very often called. The weights are determined by first calculating the number of calls to each class over all the Toll Tickets. This gives us the histogram as shown below where the scale used is logarithmic :

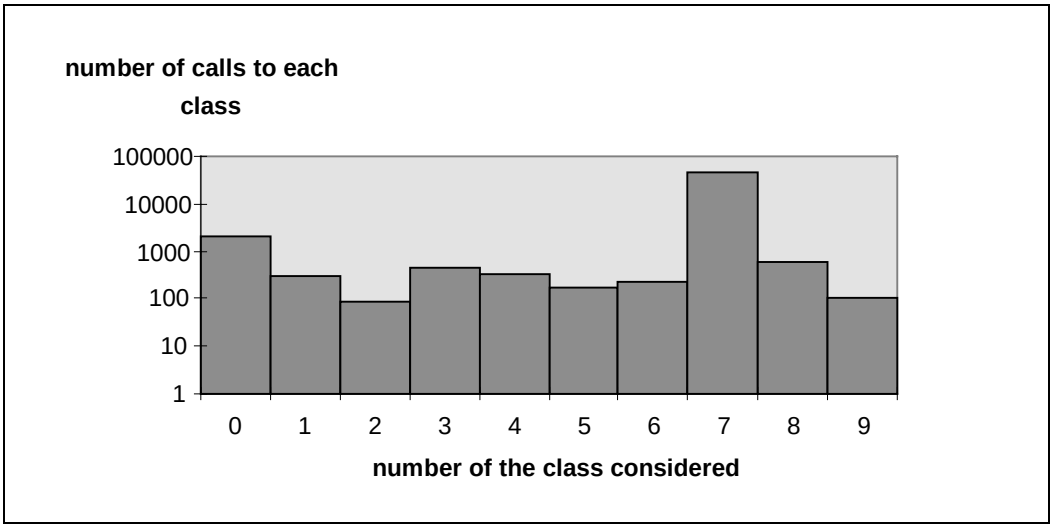


Figure 8.11 - Number of calls made to each class over the sequence of international Toll Tickets.

The figure shows that the majority of calls are made to the European Community class, which corresponds to class number seven, while the other classes seem to have a more or less equal probability of being called. It was decided therefore to assign a weight of 0.5 to the European Community class, and to assign each other class a weight given by:

$w_i = 1 - \frac{n_i}{\sum_{j \neq k} n_j}$, where n_i is the number of calls to class i, and k is the index of the European Community class. The factor 0.5 for the European Community was chosen arbitrarily. It has to be high enough to allow changes in this class to raise an alarm and low enough so as not to raise an alarm too often.

8.5.6 Brutus

In this section we present a high level description of the fraud detection trial configuration. The integrated tools will process Toll Tickets in a sequential manner as shown in the figure below. Toll Tickets flow through the architecture accumulating information pertaining to the analysis as it takes place. Subsequent modules have the ability to utilise this information in support of their own decisions. We refer the reader to the previous sections describing the technical approach to each of the modules.

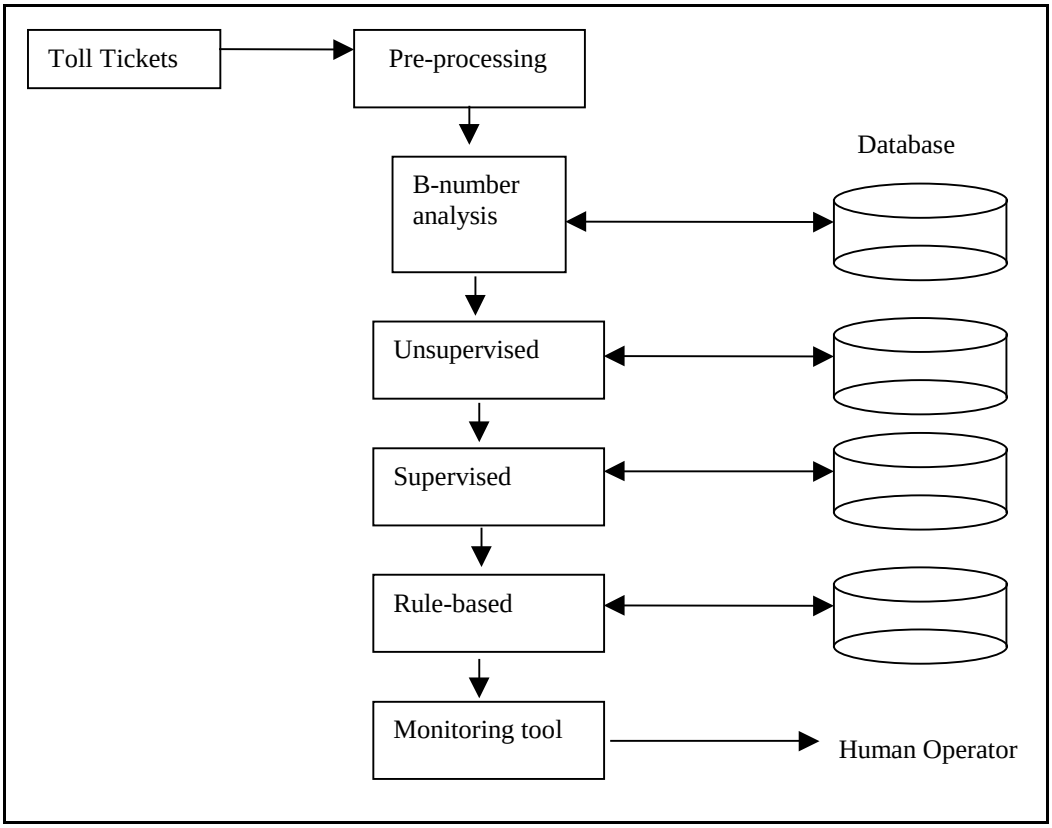


Figure 8.12 - Trial Architecture

All tools are using the same database implementation - which is GDBM, a simple and fast database on UNIX. This database fulfils the needs of user profiling, where the data records are always accessed via the IMSI as the only key. Within the first prototype the monitoring tool simply used a set of files for storing monitored users and the calls made by them.

Prior to the integration each tool was embedded into a common 'real' environment comprising a mediation device simulator feeding the tools with toll tickets and a monitoring tool to collect suspicious IMSIs and the corresponding call data. This environment will be kept for the project trial.

The trial fraud detection system will come with one common GUI to display the alarms. This GUI has mainly been finalised as part of the monitoring tool. In addition, some basic administration features are welcome enhancements. One of these features is a control-bar for each contributing tool to adjust their respective alarm thresholds. This way the tools are adjustable against each other during runtime and we can easily specify to what extent each tool contributes to the common result. A higher-level monitoring tool manages the exchange and representation of the information and for example also sets certain tuning parameters in the fraud detection tools for customising the fraud detection sensitivity.

The ordering of the fraud detection tools is motivated by the following considerations:

- The B-number analysis is added because it is believed that adding information on the destination of the calls will be able to boost the already reported performance of the individual tools. This module comes first because it adds information to the data-stream that could easily be used by all three tools.
- The Unsupervised NN is very good for novelty detection. It has good negative predictive value, which means that it can eliminate those users very easily for which certainly nothing is happening. Therefore it could be used as a first filter to all incoming calls. Another reason for putting this module high up in the chain is that its profiling could possibly also be used as input to the Supervised NN.
- The Supervised NN can efficiently pinpoint users whose behaviour is similar to previously observed and recorded fraudulent behaviour. Its training routines can be tuned to bias the performance towards a high

positive predictive value, i.e. when it puts a fraudulent label on a user, the subsequent modules and/or human operator can be confident that there really is something happening.

- The Rule Based system does very well in explaining why alarms have been raised. It could for example be extended with extra rules to inspect why previous modules had raised an alarm. In this fashion, new fraud scenarios can possibly be identified. It can also be used to define hyper rules based on alarms raised by other tools and not only on its own profiling/information.

Initially, raw data from the network is pre-processed, discarding irrelevant components, and retaining useful data fields encoded in a suitable format. Secondly, the already present information on the user (i.e. the user's profile) is retrieved from the database. From the profile and the incoming data, relevant observables are derived. With these observables, we perform the following actions:

- The profile is updated and stored in the database for later re-use.
- An audit trail is maintained.
- The artificial intelligence component performs the fraud detection and generates a report on the alarm status of that user.

This report is then handled by the intelligent monitoring tool, which serves as a (graphical) interface to the human operator. Tasks performed by the monitoring tool are:

- Filter the types of alarm that the operator wishes to handle.
- Generate operator customised data and alarm level presentation for visual inspection.
- Set tuning parameters in the detection tools.

Raw data consists of TTs entering the prototype. The pre-processing block selects data fields and puts them into a format easily manageable by software. Each 'detection module' in the subsequent chain then extracts and uses the data of its choice from the general data-stream. Then the module adds its findings/results to the data stream while leaving the original data unchanged. The next module can then select from this augmented data stream the relevant information that it wishes to use in its own fraud detection. This means that subsequent fraud detection modules can use the profiling and/or the results of the preceding module. In a first implementation, where the work will be distributed over different computer platforms, each detection module will keep its own database.

Individual modules forward information that they receive from any other module and add tagged information of their own should it be required. The data is fully human-understandable at all times. It is structured as a sequence of tag/value elements.

- Tags are four-printable-symbol strings.
- Values are arbitrarily long printable-symbol strings.
- The size of a Tag label should be fixed.
- Blank spaces are used to separate tags and values.

The input behaviour of each tool is the following. When it reads a string, the tool scans it for the tag of the fields it wants to use and extracts the corresponding values, overlooking the tag/value pairs it does not use for its own processing. The first six elements (twelve fields) correspond to the six fields in the toll ticket used in the first demonstrator. The output behaviour is the following. The tool copies the string it received at its input directly to its output, and adds tag-value pairs for all the information it wants to output (for example, for use by the monitoring tool). Writing to standard output should only happen at one place in the code, so that the structure of the output can be updated easily.

Example:

The output of the Toll Ticket simulator might look as follows:

TMSI F23415124546303b2d224c63 TCSD 19960716 TCST 220038 TCDR 000693 TBNB

FFFFFFFFFFFFFF30198672b641014 TBTP 01 TSDN 0016 TSTS 079238 TBZC 03

TMSI F23415124546303b2d224c63 TCSD 19960716 TCST 221856 TCDR 000031 TBNB

FFFFFFFFFFFFF017433333d571a4f TBTP 00 TSDN 0016 TSTS 080336 TBZC 00

TMSI F23415124546303b2d224c63 TCSD 19960716 TCST 222015 TCDR 000367 TBNB

FFFFFFFFFFFFF017433333d571a4f TBTP 00 TSDN 0016 TSTS 080415 TBZC 00

TMSI F23415140807624d312f4b4c TCSD 19960716 TCST 224913 TCDR 000003 TBNB

FFFFFFFFFFFFFFFFFFF4646250c79 TBTP 00 TSDN 0016 TSTS 082153 TBZC 00

This output is sent to the fraud detection tool. But the tool might not need all these fields. Let us say that instead of using the B-type, it prefers to use the zone code TBZC (Toll Ticket B-number Zone Code) that gives the region of the world the call was made to. Also, it does not use the TSDN (Toll Ticket Starting Date Normalised) and TSTS (Toll Ticket Starting Time in Seconds) fields. It further processes the information, possibly producing an alarm. At the output, it copies what it received at the input plus all the information it finds relevant (here, the information about the alarms). The output could look as follows:

TMSI F23415124546303b2d224c63 TCSD 19960716 TCST 220038 TCDR 000693 TBNB

FFFFFFFFFFFFFF30198672b641014 TBTP 01 TSDN 0016 TSTS 079238 TBZC 03 SALR

ALARM SALV 0.87

TMSI F23415124546303b2d224c63 TCSD 19960716 TCST 221856 TCDR 000031 TBNB

FFFFFFFFFFFFF017433333d571a4f TBTP 00 TSDN 0016 TSTS 080336 TBZC 00 SALR

NOALR SALV 0.37

TMSI F23415124546303b2d224c63 TCSD 19960716 TCST 222015 TCDR 000367 TBNB

FFFFFFFFFFFFF017433333d571a4f TBTP 00 TSDN 0016 TSTS 080415 TBZC 00 SALR

NOALR SALV 0.33

TMSI F23415140807624d312f4b4c TCSD 19960716 TCST 224913 TCDR 000003 TBNB

FFFFFFFFFFFFFFFFFFF4646250c79 TBTP 00 TSDN 0016 TSTS 082153 TBZC 00 SALR

NOALR SALV 0.12

TMSI F23415137f381c3f526f710a TCSD 19960717 TCST 074916 TCDR 000001 TBNB

FFFFFFFFFFFFFFFFFFF4646250c79 TBTP 00 TSDN 0017 TSTS 028156 TBZC 00 SALR

ALARM SALV 0.98

TMSI F2341513363e667947231933 TCSD 19960717 TCST 080242 TCDR 000023 TBNB

FFFFFFFFFFFFFF301423d5c494b49 TBTP 01 TSDN 0017 TSTS 028962 TBZC 03 SALR

NOALR SALV 0.07

The added information consists of the flag SALR (Supervised NN ALaRm) and the alarm level SALV (Supervised NN Alarm LeVel).

8.5.6.1 Monitoring and Graphical User Interface

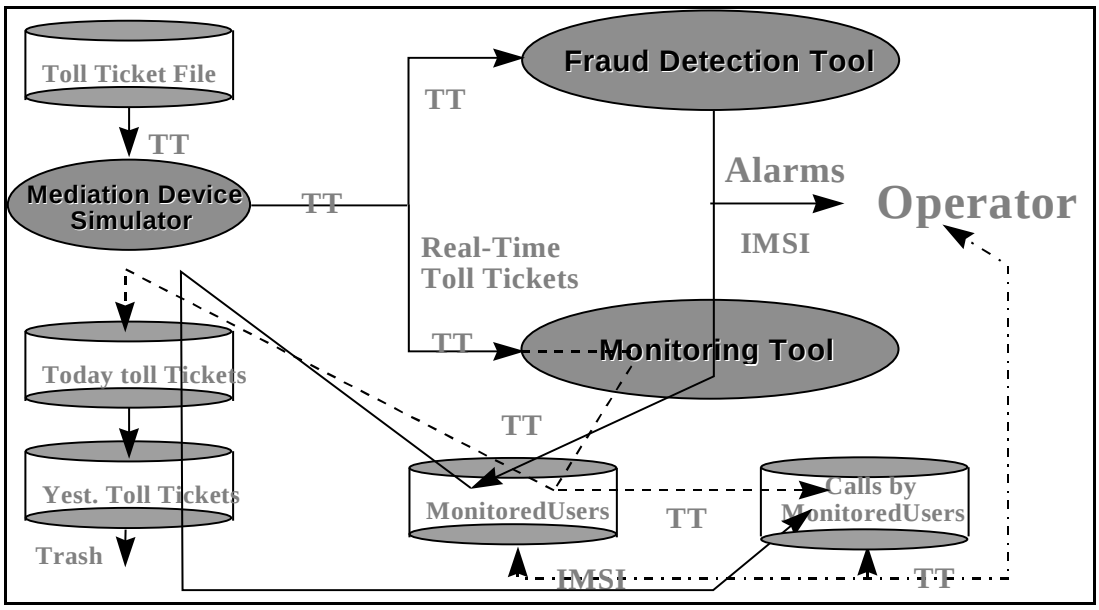


Figure 8.13 - Monitoring within the First Demonstrator

Figure 8.13 shows the monitoring as it was in the first demonstrator. The framework comprises a number of C, C++ and Perl software simulations to emulate a real time processing environment. Firstly, the billing mediation device is simulated to send Toll Tickets out to the fraud detection tool with a pre-set mean interval between each ticket and variance based on a Poisson distribution. These Toll Tickets are also passed to a monitoring tool, which checks for alarms being raised by the fraud detection tool. The monitoring tool then stores toll tickets for any subscribers exhibiting suspicious behaviour. The monitoring tool also keeps files containing the current day's and previous day's Toll Tickets in files sorted by IMSI. Once a subscriber has become suspicious his TTs are retrieved from these files and stored with the calls by monitored users.

For the integrated fraud system the monitoring framework has been improved in three ways.

Improvement 1: *Use fraud probabilities.* Instead of the information "alarm yes/no" we are using alarm levels A, $0 \leq A \leq 1$ denoting fraud probabilities. Each of the fraud detection engines will compute an alarm level and will add this information to the common stream of information passed through all fraud engines towards the monitoring tool. We can view the TTs as being attributed with a list of alarm levels. These attributed TTs (ATTs) will be the basic information within the monitoring framework. The alarm levels have to be standardised across all fraud engines so that they are comparable and can be used for ordering and common evaluations. A framework of this monitoring is depicted in Figure 8.14.

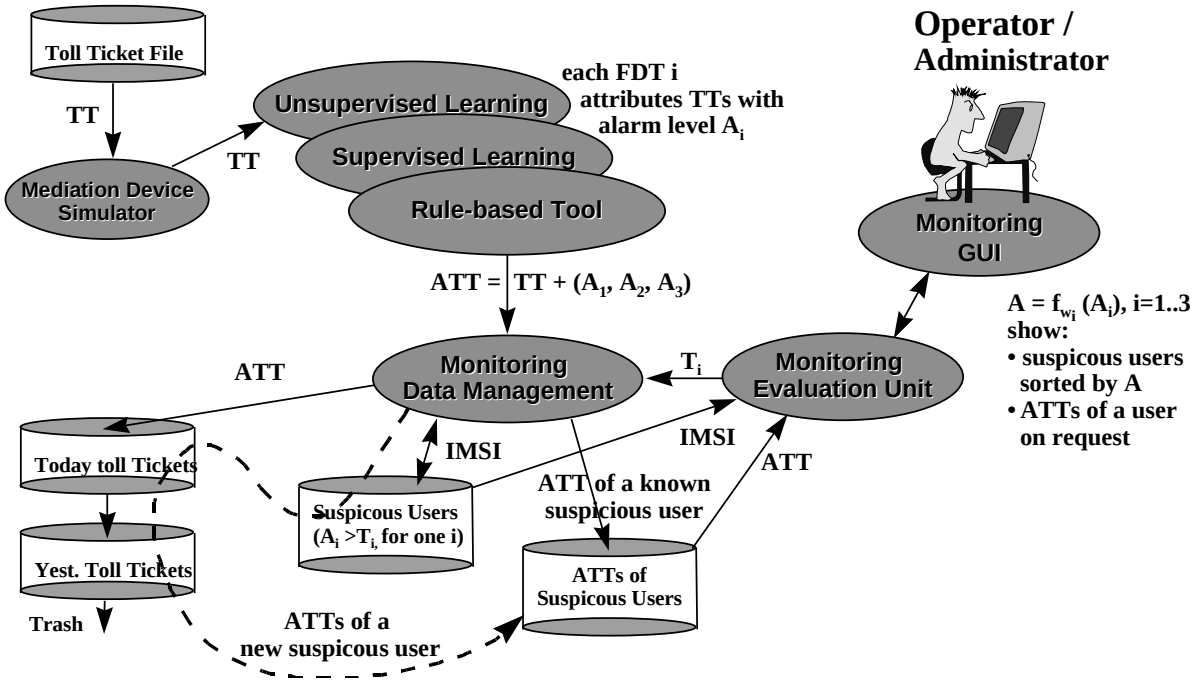


Figure 8.14 - Monitoring within the Integrated Fraud System

Improvement 2: *Evaluate The Monitored Information.* In the first demonstrator the monitoring task was restricted to data collection only. The result was a list of suspicious users and a list of all TTs of suspicious users. This part will be a basic part of the integrated prototype as well and is named "Monitoring Data Management". The main difference from the first demonstrator's monitoring is that we are using much lower thresholds, i.e. the subset of suspicious users for closer monitoring will be much larger than the set of monitored users in the first demonstrator. For each fraud engine a basic threshold T_i will determine whether a user will be classified as suspicious. A user will be classified suspicious if his alarm level exceeds one of the thresholds T_i. In general, a suspicious user is not a person raising an alarm. Suspicious in this case means candidate for further investigations. A user who is classified not suspicious at a certain time may however be classified suspicious later on.

When a user is classified suspicious for the first time, he will be stored into the suspicious user's file. All ATTs relating to him will be extracted from the buffers containing all TTs of the last two days. These buffers may also be expanded to a longer-term history. Subsequently, the extracted ATTs are stored in a separate file for further investigation. For users already known to be suspicious all ATTs are directly fed into the ATT file.

Of course, there must also be a mechanism to classify users as not suspicious again. This will be regularly done by inspecting the ATT-file of suspicious users. Such users who fall below certain thresholds will then be deleted from the suspicious users file.

Based on the suspicious users file and the ATT-file of suspicious users, an evaluation will be performed by the "Monitoring Evaluation Unit". The evaluation unit will work on the suspicious users and their ATTs only. The main evaluation task is preparing information about suspicious users as requested by the monitoring GUI. Thus, the evaluation unit provides a GUI-server. It is intended to implement the evaluation unit in Java. In view of the time limits in ASPeCT it will be checked whether it is possible to realise this server as a Web-Server accessible via HTTP.

Improvement 3: *Show suspicious users and behaviour via the GUI.* The monitoring GUI comprises functionality for supervision and for configuration. The most important feature will be showing suspicious users sorted by their alarm levels i.e. by the probability of being fraudsters. The related graphical element will be a table showing the fraudsters' IMSIs, the alarm levels A_i for each tool and a common alarm level A_{com}. By selecting a user entry in the GUI the set of ATTs relating to the user can be recalled.

8.5.6.2 Implementation of the GUI

The monitoring tool has been interfaced with a graphical interface that allows the operator to control the alarms produced by the integrated system. The GUI is programmed using the Javascript scripting language. This allows us to control the monitoring tool through a simple Internet browser, such as Netscape. A screenshot of the GUI is presented below.

The GUI incorporates the following functionalities. It gives a list of all the suspicious users. This list can be updated when new alarms are generated (Brutus the dog wakes up when a new alarm is produced and goes back to sleep when no alarms are produced). By selecting a user in the list of suspicious users, we can generate a list of all the calls made by this suspicious user. We can then provide a graphical display of the duration of these calls and their destination (national, international, or supplementary services). We can also edit a record for this user that will contain all the information that the operator deems necessary for the analysis of the case. It is possible for the operator to email this information to another service that for example deals with the disconnection of the suspicious users. When alarms are wrongly generated, it is possible to remove a suspicious user from the list of suspicious users temporarily or filter this user out permanently. Finally, the operator can change the weighting parameters of the different tools or the alarm threshold of the integrated tool at convenience.

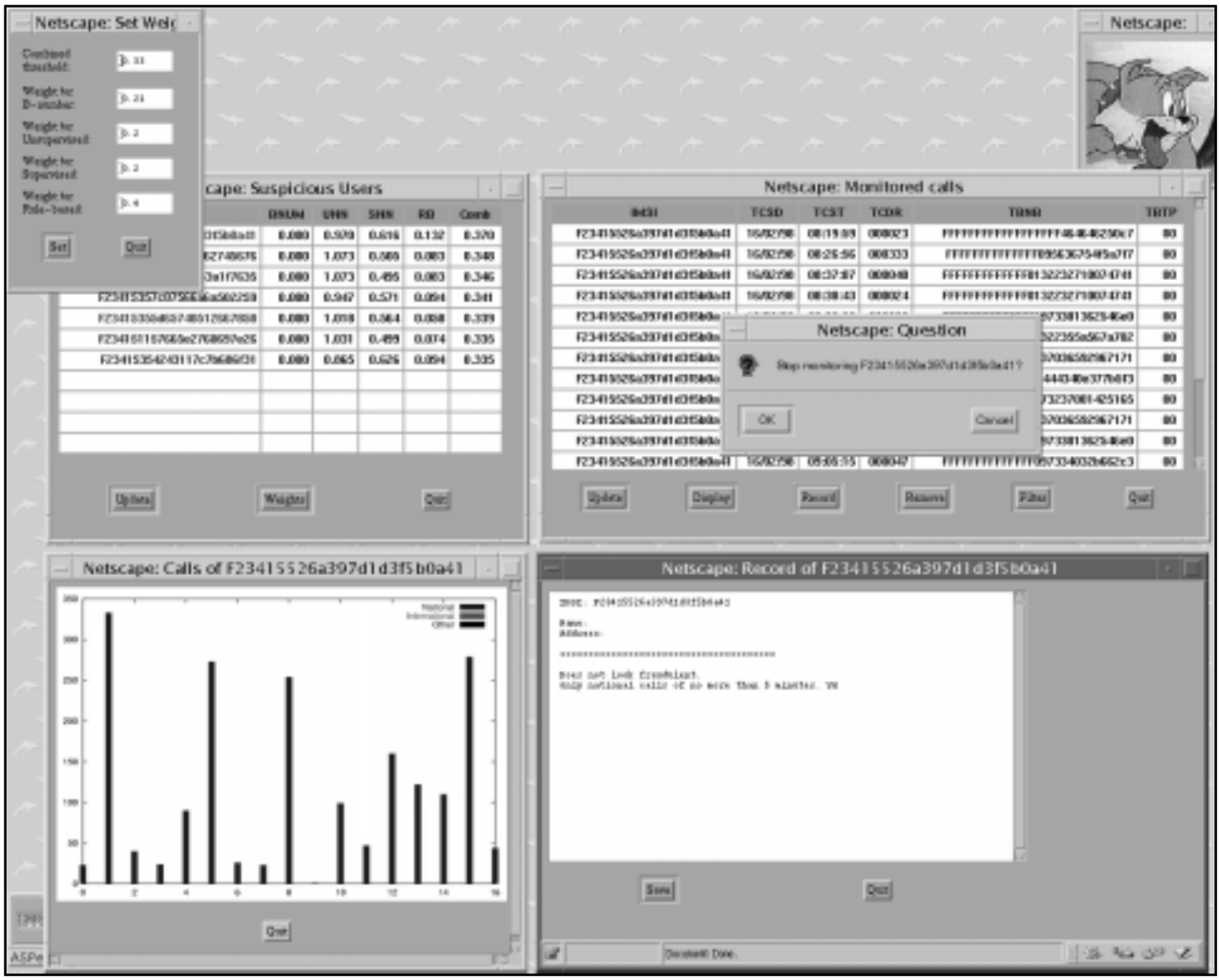


Figure 8.15 - Screenshot of the graphical user interface of the integrated fraud detection system.

8.6 Description of trials

8.6.1 Data sets

For a realistic trial genuine network data needs to be used. Suitable UMTS network data is not available because - apart from the fact that the number of users in UMTS trials is quite small - only simulated fraud could occur there. Therefore, the fraud detection concepts can not be validated in a UMTS system. Only in operational commercial networks, serving a large user population, is there a substantial probability that fraud attempts will occur. Thus, the Network Operators need to supply real usage data, representative of the operation of their network, not subjected to any ordering or filtering, to be used for the evaluation process.

Additionally, taking into account the data protection laws, the user data must be converted in a way that prevents any user identification, before processing by the fraud detection tools. This is the sanitisation procedure, by which all fields with references to users' identities in the subscriber data are encrypted. The fields are enciphered without additional context, therefore allowing the comparison of these fields between different toll tickets. Thus, it can still be checked whether two fields refer to the same item (user or destination).

8.6.2 Vodafone data

The data used in the trial is GSM toll tickets in an ASPeCT specific sub-set of the archived Eurobill format. This particular format contains 25 fields, among which the ASPeCT tools can isolate the particularly important ones for fraud detection. Examples of such fields are A- and B-numbers (call originator and call recipient subscriber numbers), the call starting time, the duration of the call.

Vodafone Ltd provided a set of data containing approximately 4 months worth of toll tickets for approximately 20,000 users. The users were chosen as a series of CHARGED-IMSI groupings to capture an expected wide range of behaviours within the data. The Toll Tickets were collected and stored on a daily basis. Once collected, this data was concatenated into approximately 100 files for distribution, training and processing. From this data it would also be possible to select only a small subset of the users to further reduce the size of the data set.

All subscriber information contained in the respective toll ticket fields was encrypted whilst preserving the 25-field format, as well as all information that allows distinction between individual users. Thus, the confidentiality of personal data was protected while the case individuality is retained. Any suspicious users that are identified can be investigated within Vodafone by reversing the sanitisation process. This analysis is included as part of the evaluation of the tools.

8.6.3 Combination of the different tools

The common alarm level $A_{com} = f(w_1, w_2, w_3, A_1, A_2, A_3)$ will be used for the ordering. The function f is used for combining the thresholds computed by all tools to a common alarm level. The weights w_i will allow users to manually adjust the influence of each tool on the common result. One of the main tasks of the trial was to determine a well-suited function on combining the results.

We opted for a well-known approach from statistical theory: logistic regression modelling. The combination function has then the form $f = 1/(1+\exp(-\Sigma\, w_i A_i))$. The advantages of this combination function are the following:

a) The determination of the parameters w_i is straightforward. The necessary optimisation can be based fully on the individual results of the different tools (B-number analysis, unsupervised neural network, supervised neural network, and rule-based system) on a training-set similar to the one used in the development of the supervised neural network

b) The number of parameters that have to be estimated is low and the optimisation procedure is guaranteed to converge to an optimal solution.

c) The resulting parameters w_i are also statistically meaningful in that contributions with large parameter values contribute exponentially more to the probability of fraud than contributions with low parameter

values. This interpretation means that our integrated tool is building an estimate of the probability of fraud on the basis of the behaviour of a user.

This logistic regression modelling provides a start estimate of the relative weighting of the individual tools. Adjusting the weights during the daily operation of the fraud detection engine will be a task of an administrator. The same holds for changing the thresholds for a minimum suspicion. An adjustable global threshold T_{com} will allow the raising of an alarm, if A_{com} exceeds T_{com}. This is meant for the critical cases where the tool should alert an operator and in a later stage proactively propose countermeasures. In our software implementation of the fraud detection tool, the operator uses a simple web browser interface to adapt these weights.

8.7 Results of trials

We evaluate the performance of the integrated tool along the following lines. First, we focus on numbers and percentages and measure the classification performance of the combined tool through the previously introduced method of the Receiver-Operator-Characteristic Curve. Secondly, we investigate the usability of the tool's alarms from the perspective of the Network Operator. We generate a list of suspicious users from the toll ticket data provided by Vodafone and have it investigated by Vodafone's staff.

8.7.1 Evaluation of technical effectiveness

To assess the performance of the combined fraud detection tool used in the trial, we use the same performance index that was used to measure the performance of the individual tools in the previous reports. Also there we focused on finding a performance index that reflects the daily practice of fraud management. The striking feature of fraud detection is the importance of the trade-off between detection of fraudulent users and the production of false alarms. Indeed, we could develop a very conservative system that would generate alarms at the lowest levels of suspicion. But Network Operators and Service Providers are, from a commercial point of view, extremely cautious about unduly bothering good subscribers. Moreover, even levels of false alarms that would be considered excellent from a statistical point of view (let us say, one percent of misclassification), would be completely unacceptable in our case. For example, one percent false alarms for one million subscribers means ten thousand false alarms. Conversely, we could guarantee that we do not generate any false alarms simply by not implementing any fraud detection system. Yet, the burden of loss of revenues caused by fraud makes this solution unattractive.

Therefore, the problem of the fraud detection tool will be to find the right balance between false alarms and correct detection. This optimal trade-off might be different for different operators, different services, or different periods. We thus developed the fraud detection tools to produce a single measure of suspicious behaviour each time it receives a new toll ticket. The decision itself comes from choosing an appropriate threshold and deciding to classify a user as suspicious if its activity rises above that threshold at any time of its profile history. A low threshold will guarantee high detection, but will generate many false alarms. High threshold will guarantee few false alarms, but will detect few fraudulent users.

The Receiver-Operating-Characteristic plots the percentage of correct detection of fraudulent users versus the percentage of false alarms for new users (as illustrated in the coming sections). The index of performance that we need to maximise is the surface under the curve. This is a very practical index of performance, more appropriate to our investigations than standard statistical measures. Such a trade-off curve will give the user of the fraud detection tools control over the fraud detection rate and false alarm rate.

8.7.2 Technical description of the data

The available examples of frauds were collected from the TACS network of Vodafone. This data contains a total of 317 fraudulent users who generated 131.594 toll tickets. The example of live network data consists of a three-month download (from 16-02-1998 to 16-05-1998) from 20.212 users for a total of about three million toll tickets. The number of fraudulent users in the live data is expected to be low.

From the live data, we selected 562 random users for which we retained the toll tickets from the first 40 days of the Vodafone data. We limited the time range to 40 days so that the average period of activity of the

normal users is the same as the average lifetime of a fraudulent example. We selected 500 users from the live data purely at random and added another 62 random users from those users who had placed international call.

The evaluation data is split into a training set and a test set. The training set is used to determine the parameters of the integrated tool. We use the test set to evaluate the different fraud detection tools and also to evaluate the integrated tool. In the available data, we chose users that were most representative of either suspicious or non-suspicious behaviour.

The training set then consists of 71 examples of fraudulent users and 153 users from the live data. Those 71 fraudsters made 25,523 national calls and 2,933 international calls. We labelled these calls manually and selected 16,831 national calls and 2,728 international calls that were again most representative of fraud. The 153 regular users made 22,003 national calls and 1,255 international calls. We labelled these calls manually and selected 19,257 national calls and 1,038 international calls that were again most representative of normal behaviour.

The test set then consists of 105 examples of fraudulent users and 321 users from the live data. Those 105 fraudsters made 39,550 national calls and 6,088 international calls. We labelled these calls manually and selected 23,764 national calls and 5,507 international calls that were again most representative of fraud. The 321 regular users made 28,807 national calls and 1,055 international calls. We labelled these calls manually and selected 24,807 national calls and 826 international calls that were again most representative of normal behaviour.

The proportion of the number of international calls to national calls in the evaluation data (both training set and test set) is therefore higher than the real proportion in the live data. This bias increases the discriminating power (sensitivity) of the fraud detection tool by forcing it to deal with the more confusing examples (users who call internationally are more difficult to separate from the fraudulent users as these often call internationally). Our evaluation data is therefore more difficult than the live data from the network but we can expect our tool to have a performance higher than the performance achievable with a balanced data set.

8.7.3 Performance of the individual tools and integration

We evaluated the performance of the different modules and of the integrated tool again with the Receiver-Operating Curve. This ROC curve gives the percentage of fraudulent toll tickets correctly detected versus the percentage of toll tickets incorrectly generating alarms. This procedure is different from studying the percentage of fraudulent *users* who are detected. Our measure is more targeted towards the actual operation of the system, where the operator needs to classify each toll ticket to make a decision about a user.

The data is split into training set and test set. The training set is used to determine the parameters of the combination function (logistic model) of the different sub-modules. The test set is used to evaluate the performance of the integrated tools. The optimisation procedure is based on the optimisation of the squared error followed by the maximisation of the area under the ROC curve for the region of relevant performance (less than one-percent false alarms). For the different sub-modules, both the training set and the test set have never been seen before and must therefore be regarded as purely prospective tests.

8.7.3.1 Comparison of alarm histograms and ROC curves

In this section we will report on the individual performance of the tools. We show both the histograms of the alarms on fraudulent and normal toll tickets and the ROC curves. These are given from top to bottom for the training and the test set.

8.7.3.1.1 B-number

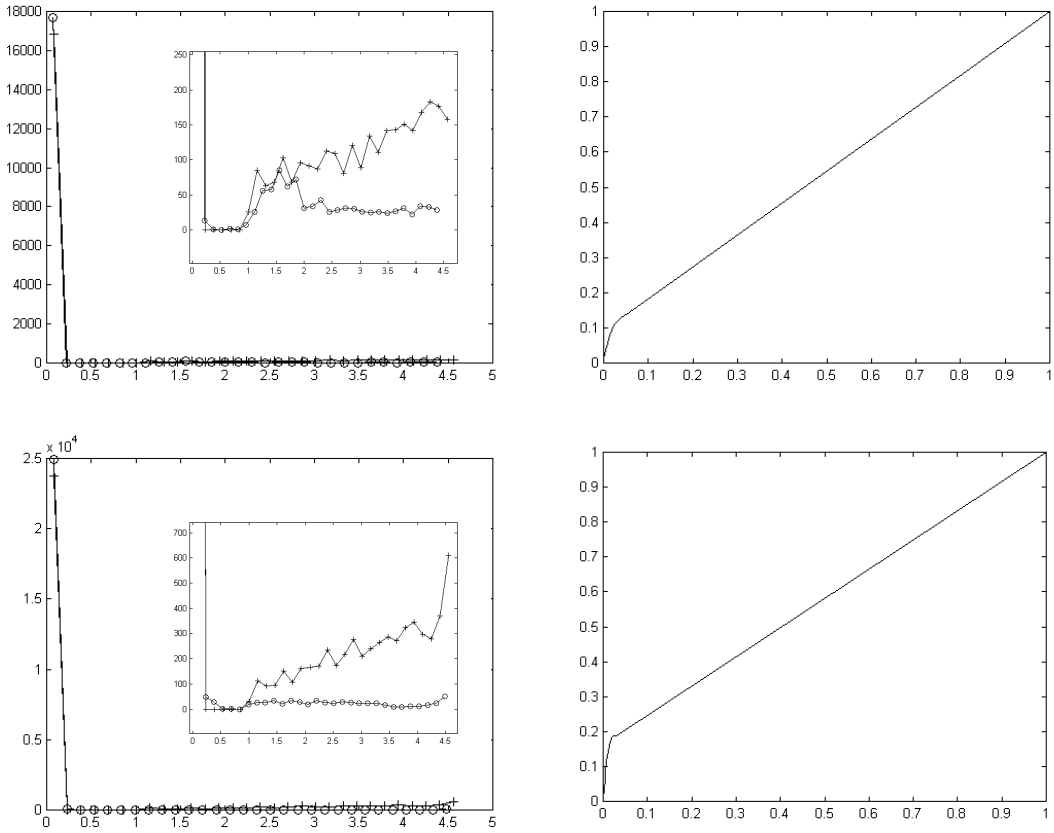


Figure 8.16 - The results of the B-number analysis tool:
histogram (left) and ROC curve (right), for training (top) and test sets (bottom)

8.7.3.1.2 Unsupervised neural network

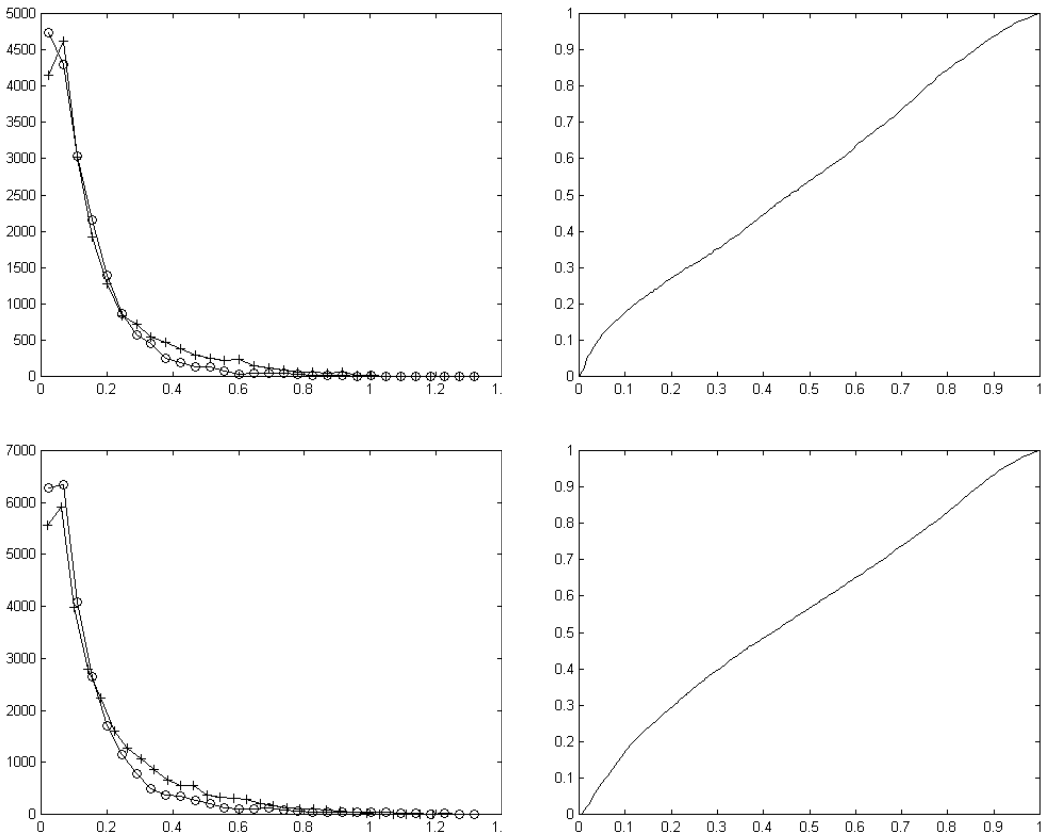


Figure 8.17 - The results of the unsupervised neural network tool: histogram (left) and ROC curve (right), for training (top) and test sets (bottom)

8.7.3.1.3 Supervised neural network

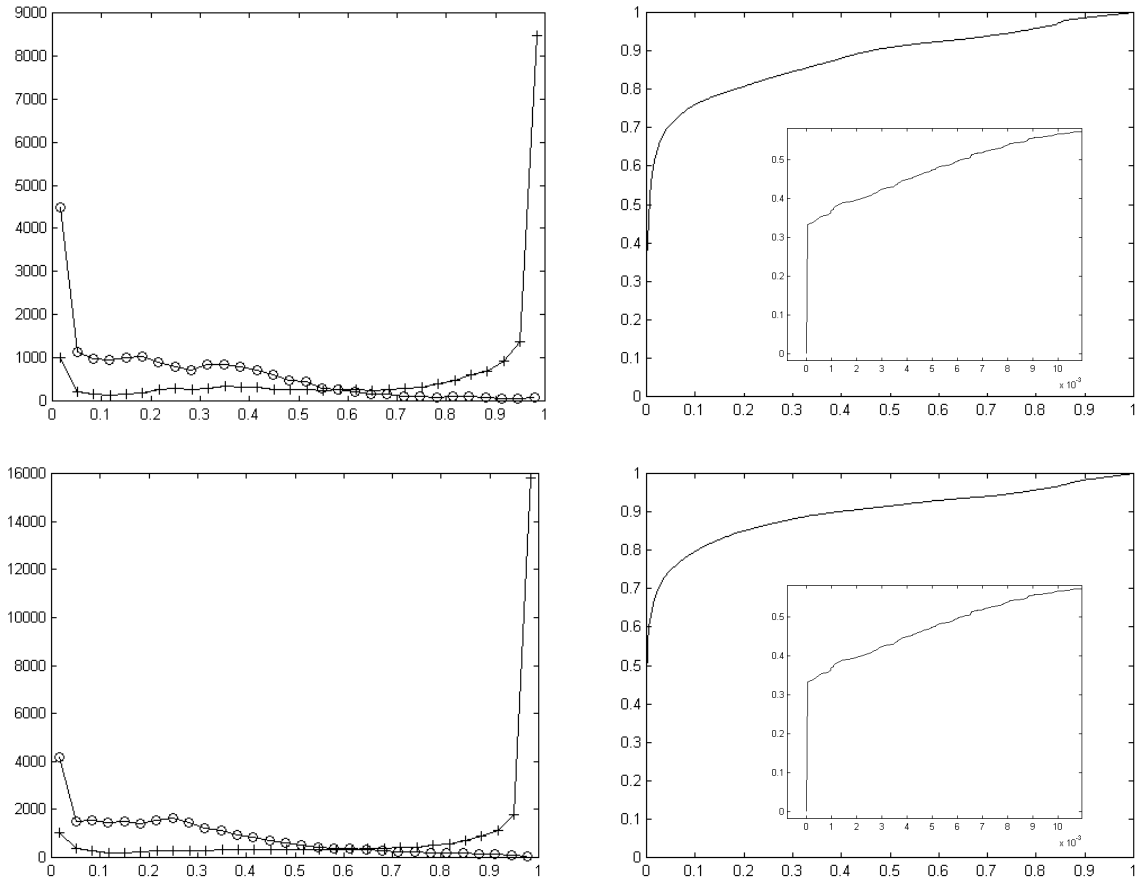


Figure 8.18 - The results of the supervised neural network tool:
histogram (left) and ROC curve (right), for training (top) and test sets (bottom)

8.7.3.1.4 Rule based

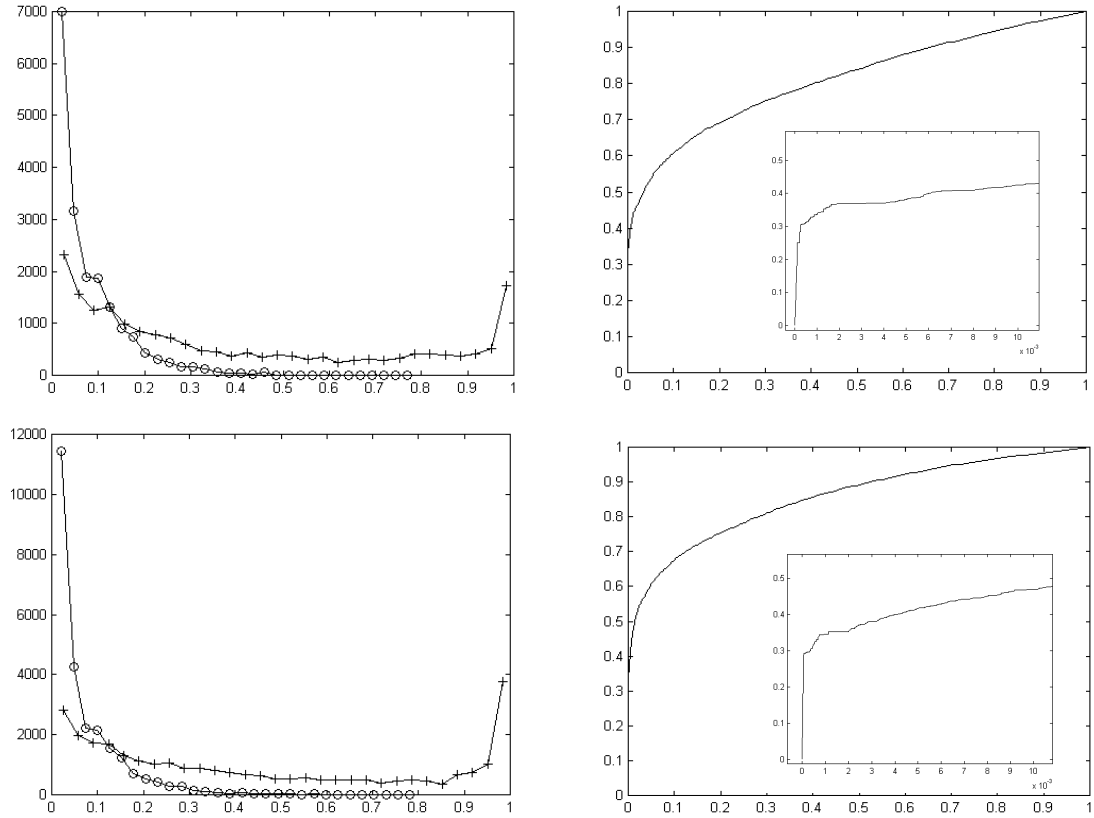


Figure 8.19 - The results of the rule based tool:
histogram (left) and ROC curve (right), for training (top) and test sets (bottom)

8.7.3.2 Integration of tools

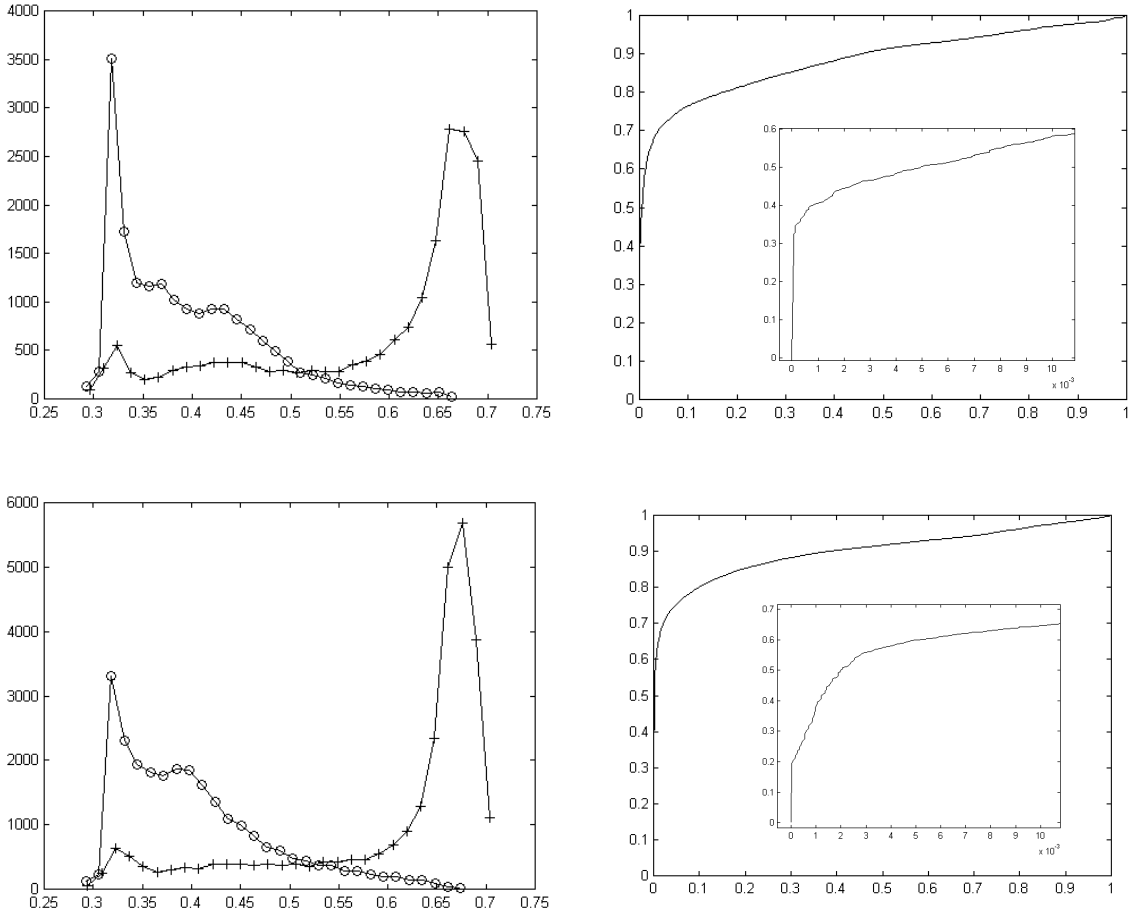


Figure 8.20 - The results of the integrated tool:
histogram (left) and ROC curve (right), for training (top) and test sets (bottom)

8.7.3.3 Comparison via area under the ROC curve

The following table gives the percentage area under the ROC curves for the different sub-modules and for the integrated tools.

Area under the ROC-curve	Training set	Test set
B-number	54.54%	57.97%
Unsupervised	54.56%	55.75%
Supervised	87.17%	88.92%
Rule-based	81.24%	85.63%
Integration	88.45%	90.08%

Table 8.1 - Performance of the different modules and the integrated tools on the two subsets of the data.

The integration of the tools gives a 2% increase of the ROC performance against the best sub-module (supervised neural network), which is statistically significant ($p<0.0001$). But the main improvement is that the behaviour of the integrated tool in the region of low false positive rates has improved significantly. At 0.02% false alarms, the supervised neural network detects 40% of the fraudulent users (both on the training and test sets) while the integrated tool detects 45% of the fraudulent users on the training set and 50% on the test set.

8.7.4 Evaluation of user acceptability

The user in this particular trial is the Network Operator Vodafone. Preliminary analysis of Panafon data indicated that the behaviour of its users was vastly different from the behaviour of Vodafone users. The fraud detection system therefore needs to be retuned to be applicable to Panafon customers. For this reason, Panafon data was withheld from the trial.

The user acceptability of the fraud tool has been evaluated in a number of ways. The integrated fraud detection tool generated a list of 27 suspicious users over the first forty-day period of the Vodafone data. The analysts of Vodafone have looked at these suspicious users. We present the results of their investigations in the following table.

IMSI	Description	Possible Fraud
F23415357057581f4f6d3412	Heavy and varied use, but fine.	No
F234153500145a0337693e02	Sudden change in usage due to change in SIM and hence IMSI.	No
F23415355a58410f34737339	Nothing to indicate unacceptable behaviour	No
F23415356b1c191b4b7e683f	Nothing to indicate unacceptable behaviour	No
F2341535610a3262525b5048	This is a hire phone - hence strange behaviour.	No
F23415521e0a3071576f1000	Mad usage behaviour. Actually appears to be a hire phone, as bills always paid.	No
F2341552554361530479 1452	Change in behaviour due to international access being unbarred.	No
F234154858596f69281f6104	Looks like subscription fraud - Subscriber still barred.	Yes
F23415355e46124314620f01	High business usage, but OK.	No
F2341548282659013 82e7f56	Subscriber eventually had account closed. Looks like probable fraud.	Yes
F23415354154577a1c627c16	Nothing to indicate unacceptable behaviour	No
F23415526754463d06522b66	This looks like subscription fraud.	Yes
F23415482313674f79050836	Nothing to indicate unacceptable behaviour	No
F23415115622761f1c537c2f	SIM change with heavy usage.	No
F2341535474d362f54500536	Hire phone	No
F23415357a7b327f316d0261	Nothing to indicate unacceptable behaviour	No
F234154808457e5512513d16	No information available	N/K
F23415487e54520476094e36	Account closed.	No
F2341548496524781b515d73	No information available	N/K
F234154820334007 6a532f3d	Subscriber eventually had account closed. Looks like probable fraud.	Yes
F234154873083331 0e050e6a	Hire phone	No
F23415355a2c3877234a377f	SIM change with heavy usage.	No
F23415520b1c0d3a3e04371a	SIM change with heavy usage.	No
F23415357a40352c654b255e	Change in behaviour due to international access being unbarred.	No
F2341552665929064f1c5623	SIM change with heavy usage.	No
F23415356c73256e6b061c3d	Change in behaviour due to international access being unbarred.	No
F23415351e061d5a18791d59	Change in behaviour due to international access being unbarred.	No

Table 8.2 - List of 27 suspicious users with their classification as probably fraudulent or not.

We can see that of the 27 suspicious users, 4 were probable cases of fraud. (Vodafone's analysts cannot determine with absolute certainty whether or not a user is fraudulent or not as this information is available only to the Service Providers. They can however determine whether a user was barred from the network and then infer from an abnormal calling pattern that the user was fraudulent).

The following 4 IMSIs were associated to probable fraud: F234154858596f69281f6104, F2341548282659013 82e7f56, F23415526754463d06522b66**,** F234154820334007 6a532f3d. We are going to analyse the behaviour of these users in more details. We present first a table giving the date of the first call in our data set, date of the first alarm, and date where the user was barred from the network (i.e., the date where the fraud was detected and the account terminated).

Fraudster	Date of first call	Date of first alarm	Date of barring
F234154858596f69281f6104	**Feb 18, 1998**	**Feb 24, 1998**	**Mar 14, 1998**
F2341548282659013 82e7f56	**Feb 16, 1998**	**Feb 19, 1998**	**Mar 6, 1998**
F23415526754463d06522b66	**Feb 17, 1998**	**Mar 12, 1998**	**Sep 10, 1998**
F2341548203340076a532f3d	**Feb 16, 1998**	**Feb 19, 1998**	**Oct 6, 1998**

Table 8.3 - Fraudulent users with the date of first call in the data, date of first alarm, and date of barring

For the first two fraudsters we see that the first alarm is produced not too far from the barring date, while for the last two fraudsters the barring occurs only months after the first alarm. The interpretation of this fact is somewhat difficult. We would expect that the barring would follow the billing cycle relatively closely and that fraud should not be detected with a delay of more than a couple billing cycles. We should then regard the alarms produced by the fraud detection engine as false alarms because the fraudsters should have still been paying his bills at the time of the alarm. However, given the expected low proportion of fraud on the GSM network and the shortness of our list of suspicious users, the chance of a false alarm occurring with a "future" fraudster is low. More investigations are needed to assess the exact value of these alarms but they cannot be completed within the scope of the trials.

We now present the behaviour of the supervised neural network for these fraudsters as the implementation of this tool allows graphical displays of individual users. A first remark is that the horizontal axis of all the graphs represents the number of days starting from a fixed date (1 January 1994).

The first fraudster has both national and international calls while the last three only have national calls. The alarms in the first fraudster are principally caused by a rapid increase in the duration of international calls (about day 1515). The alarm is maintained and confirmed when the bulk of fraud begins (day 530 with a large number of calls). This is probably subscription fraud (the most common fraud on GSM networks). The second fraudster starts calling a lot right from the beginning and an alarm is raised almost immediately. This is probably also subscription fraud. The third fraudster raises an alarm around day 1530 where a sudden gap in the average duration of national calls is detected. The fourth fraudster has a large number of national calls and is detected immediately. This is probably also subscription fraud.

The system therefore produces 23 false alarms for 4 detections of fraud. This ratio of false positives to true positive is good. The Vodafone analysts however pointed out that a large number of false alarms had the following three causes: hire phones, SIM changes, international unbarring. Hire phones naturally tend to have an aberrant behaviour as they are used by business people making many international calls over a short period. Changes of SIM in the phone indicate that the owner of the phone has changed so that we are likely to observe a sudden change in user behaviour. International unbarring of subscribers happens when subscribers are allowed to call internationally only after a given period after they connected to the network. This procedure is used to prevent international subscription fraud. However, the users will have zero international activity during the barring period while some legitimate users may start calling a lot internationally as soon as the barring period ends. This produces a sudden change of behaviour that raises an alarm. The Vodafone analysts judge by coupling the fraud detection tool with a database of hire phones, SIM changes, and international unbarring that would repress false alarms, we would obtain a commercially useful fraud detection tool.

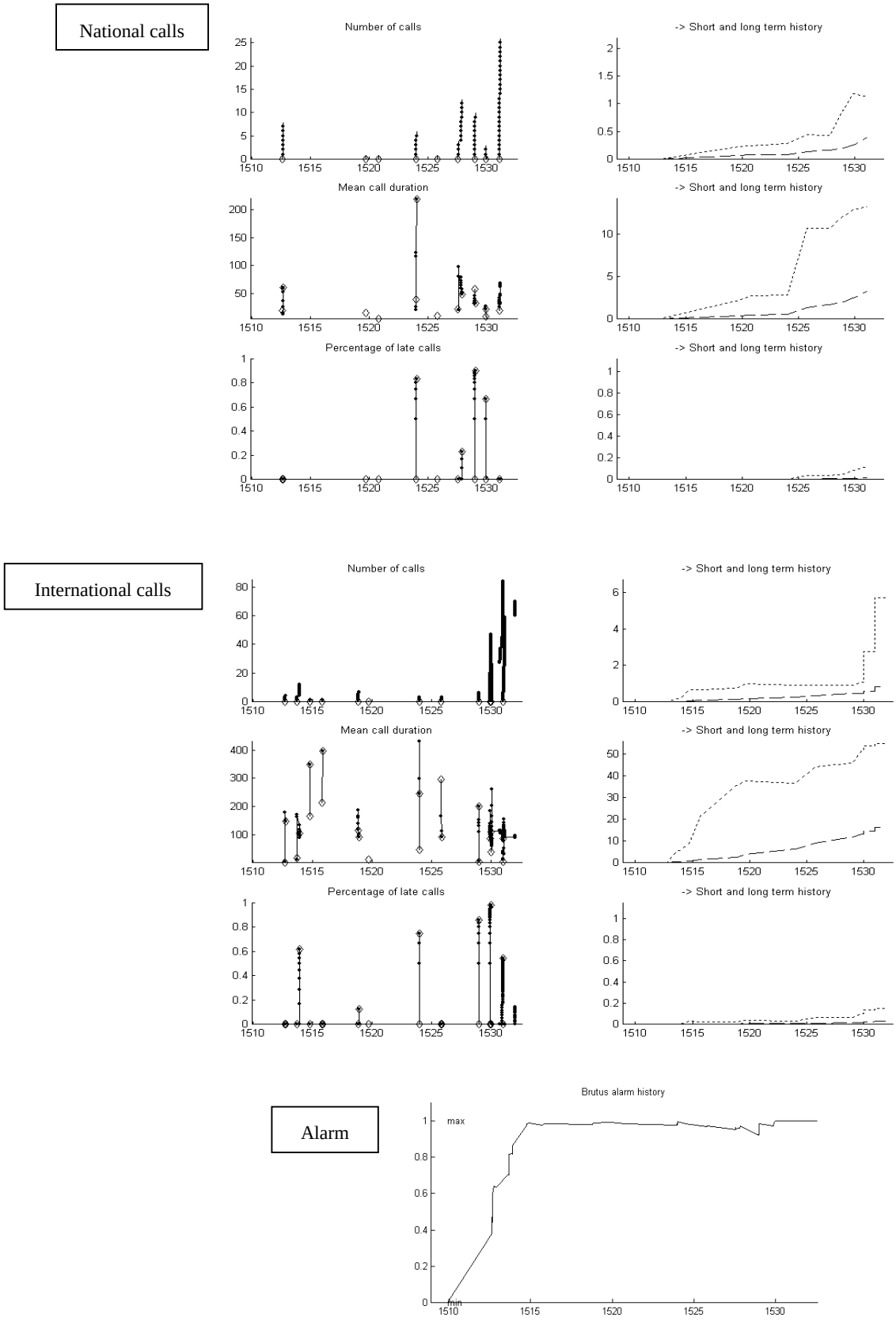


Figure 8.21 - Fraudster F234154858596f69281f6104.
Display of (from top to bottom) national calls, international calls, and alarms. For national and international calls, we see (from left to right) daily quantities and their short-term plus long-term averages. The quantities displayed are the number of calls, mean duration of a call and percentage of late calls.

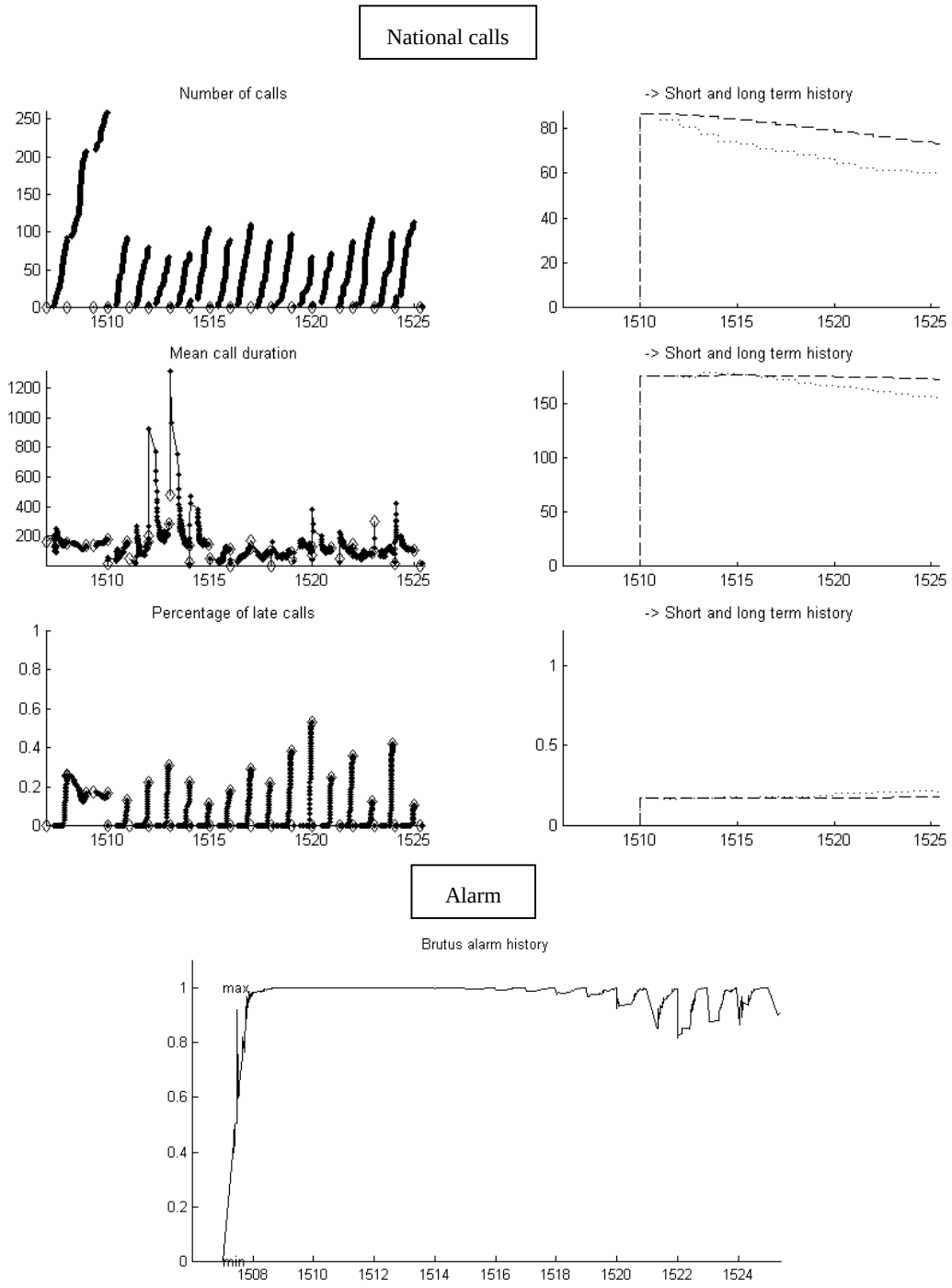


Figure 8.22 - Fraudster F234154828265901382e7f56.
Display of (from top to bottom) national calls and alarms. For national calls, we see (from left to right) daily quantities and their short-term plus long-term averages. The quantities displayed are the number of calls, mean duration and percentage of late calls.

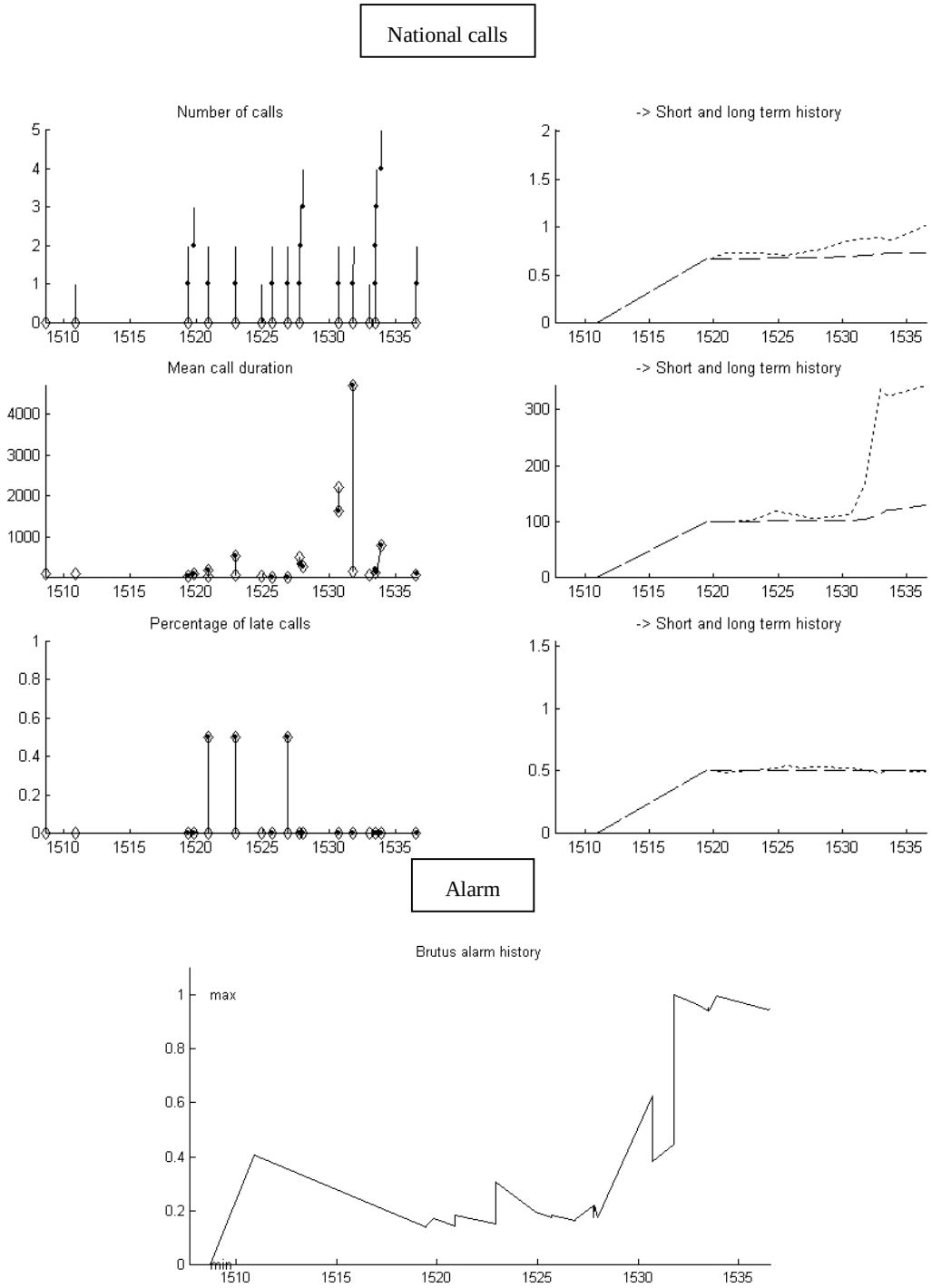


Figure 8.23 - Fraudster F23415526754463d06522b66.
Display of (from top to bottom) national calls and alarms. For national calls, we see (from left to right) daily quantities and their short-term plus long-term averages. The quantities displayed are the number of calls, mean duration of a call and percentage of late calls.

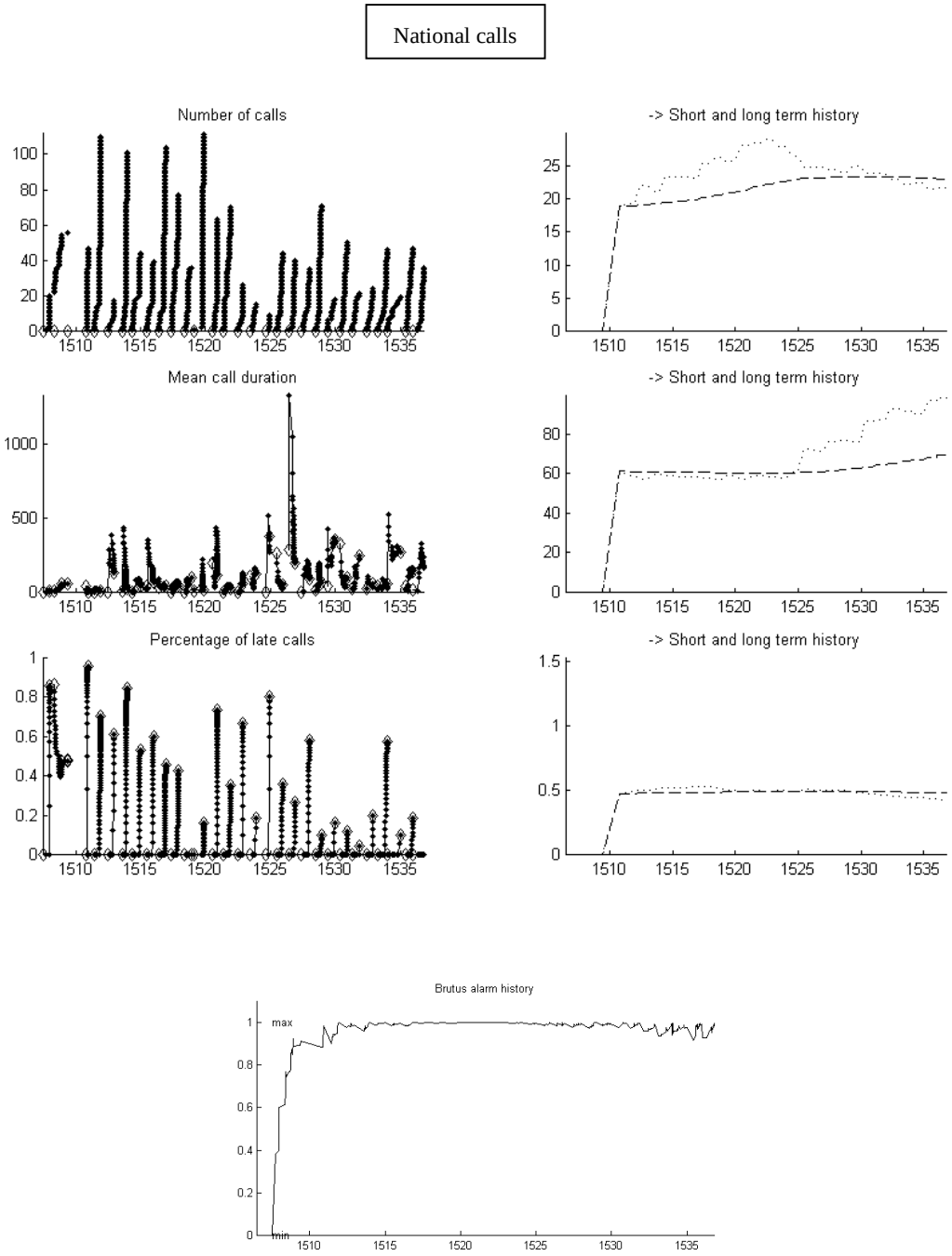


Figure 8.24 - Fraudster F2341548203340076a532f3d.

Display of (from top to bottom) national calls and alarms. For national calls, we see (from left to right) daily quantities and their short-term plus long-term averages. The quantities displayed are the number of calls, mean duration of a call and percentage of late calls.

8.8 Conclusions

The ASPeCT fraud detection and management work has made a detailed study of the requirements for a mobile Telecoms, fraud detection system. The first stage in this process was to identify the different fraud scenarios that are known to have occurred and also monitoring to ensure that any novel scenarios are incorporated into our analysis. Based on the fraud scenarios, a set of fraud indicators was identified and the relevant information contained in the toll tickets determined. At the same time a study was made of existing fraud detection techniques that have been developed in other areas as well as the basic tools available for mobile telecoms.

Based on all of the information gathered the underlying design principles of profiling and differential analysis were adopted. At the same time the three approaches to fraud detection were identified, namely rule-based, supervised learning and unsupervised learning, each of which was recognised to have specific strengths and weaknesses.

The three systems were developed and trialled on data supplied by the network operators involved in the project. The results of the trials were very encouraging showing that a significant proportion of the unseen fraudulent data was being detected by all three of the systems. Two things, however, were recognised. Firstly, as anticipated each system had different strengths and hence by combining them a more effective tool could be developed, and secondly, that there were potentially useful features of the B-numbers called by fraudsters that the current systems were unable to exploit.

The final stage of the project therefore involved two main activities, the development of a tool for B-number analysis, and the integration of all four tools into an overall fraud detection system which would allow the operator to study the pattern of calls of users who were raising significant alarms. This tool was developed under the name BRUTUS and was demonstrated on the available data. During the demonstration fine-tuning of the relative weightings of the different tools was made. The users who raised the most significant alarms were then given to the operators for them to instigate an investigation and ascertain whether any genuine fraud had been detected.

The philosophy behind BRUTUS has been to give the operator the core information about the alarms raised and provide a system to allow them to analyse the calls made by the suspicious user in order to ascertain the level and type of threat posed.

A further output of the work was the organisation of a Fraud Detection workshop which brought together expertise from the world leaders in fraud detection in credit cards as well as mobile telecommunications. Representatives from the UK government, the US navy, Neural Network companies and network operators were also present. The workshop was proved to be a very rewarding exchange of ideas and generated very positive feedback. In particular the representative of the US navy wrote to say that he would recommend that the ASPeCT team be added to the US navy database of centres of European research expertise.

The work has undoubtedly achieved its goal of demonstrating the effectiveness of novel AI techniques in detecting fraud in mobile telecoms networks. The system developed has proved efficacy but would need further development and refinement before it could be used on-line.

In addition many interesting developments of the research work could be countenanced. We will mention two particular avenues, but these are certainly not the only areas in which developments could be made.

The first is the automation of the longer-term adaptation of the 'focus' of attention mentioned in Section 8.5.4. This represents the introduction of a higher level of intelligence into the system. The second area in which interesting work remains to be done would be the use of more of the toll ticket features. The prototype system has relied on a relatively small number of features, which were identified as the most salient. There was however other features that was considered potentially relevant but was excluded from the current system. Inclusion of these features will present a number of interesting challenges. There is clearly a potential for more useful information if more features are added, but there is also a danger that the information already included is clouded by irrelevant attributes. The learning strategies will therefore have to be adapted to enable them to focus in on the relevant features and extract the most useful information.

9 Overall project conclusions

The ASPeCT project has made significant contributions to enhancing the security of the next generation of mobile communications systems in a number of areas, as was intended in the project's original objectives. It is worth reviewing these briefly to get an idea for what the project has achieved, and how the results will be carried forward.

The project can be divided into three main areas: work focussing on authentication and the UIM, work to do with trusted third parties and their use to support a secure billing protocol for value-added services, and work on the detection of fraud in mobile systems.

In view of the continuing uncertainty during the project's lifetime regarding the evolution from GSM to UMTS, the original objective of securing the evolution and migration between the generations was refocussed. The project worked on achieving a flexible protocol which allowed maximum choice of authentication algorithm and permitted negotiation on dynamic roaming agreements as well. It has demonstrated the use of a public-key based authentication procedure for authenticating UMTS users, establishing this as a contending technology for the third generation. This included showing that UMTS and GSM applications can coexist on a single smart card. Biometric demonstrations showed that speaker recognition could be used to authenticate users to their smart cards, with the support of the terminal.

Extensive investigations have been undertaken into the use of trusted third parties for mobile communications networks. The services that could be provided using TTPs have been assessed. A compact certificate format has been developed, suitable for the mobile environment. A systematic approach to the analysis of key escrow schemes has been developed.

The micropayment system used in ASPeCT is applied to pay for the provision of valued added services which provide information to the user based on WorldWideWeb technology. The novelty is not the payment protocol itself, but the way in which it is integrated with the authentication protocol proposed for the mobile system UMTS and the payment scenario for basic and value added services in UMTS.

To overcome these problems a new protocol was developed in ASPeCT for authentication between user and network; it was particularly designed to fit the performance constraints of mobile networks. Its design exploits the advances in two fields: Crypto-controller smart cards (which have a co-processor which efficiently supports public-key cryptographic mechanisms) and elliptic-curve cryptosystems (which permit the use of smaller cryptographic parameters). The new protocol was successfully implemented and tested in ASPeCT.

The ASPeCT project demonstrated the feasibility of a solution to the problem of securely billing for the provision of services using the aforementioned authentication protocol to initiate a payment scheme.

Major results of the project include the three fraud detection tools based on separate approaches to the problem of detecting and identifying instances and patterns of possible fraudulent behaviour.

- a rule based tool;
- a neural network based tool using supervised learning;
- an unsupervised learning tool utilising neural networks.

A major result of the project is the integration of these three tools, together with a fourth tool using an unsupervised learning approach to B-number analysis, into a combined tool - BRUTUS - with its own monitoring and management GUI.

All the tools adopt an approach based on analysis of *user profiles* based on comparison of recent and longer-term behaviour histories derived from toll ticket data. The neural network-based tools use a differential analysis; the rule based tool also allows absolute analysis against fixed criteria.

A report has been written with the objective of the determination of the legal rules applying in the various fields of law affected by the use of fraud detection systems by mobile communications operators or service providers.

Each of the results described has had an impact, whether on public standards, on partners' internal development plans, or on the direction of future research. The true value of ASPeCT's work will only become apparent when mobile communications are used as the standard communications tool for electronic commerce, and users' confidence in the security of the system is founded on a sound technical basis.