CITY UNIVERSITY
LONDON

Authentication and Privacy in Mobile Web Services

Kalid Elmufti

A Thesis Submitted for the Degree of

Doctor of Philosophy

to the School of Engineering and Mathematical Sciences,

City University, London

October 2008

Abstract

This thesis looks at the issue of authentication and privacy in mobile Web services. The work in this thesis builds on GSM and UMTS security framework to develop security protocols for mobile Web services environment. The thesis initially highlights some core principles of designing security protocols in such environment. The next two chapters look at the core technologies and building blocks in Web services systems and the core security features in mobile networks mainly GSM and UMTS. Registration and authentication were identified as security issues in federated systems. Proposed solutions were developed utilizing XML security mechanisms with SIM card security in GSM environment to address these issues. Also a novel system was proposed in which it is possible for a mobile user to securely authenticate and have full anonymity as far as the service providers are concerned; however it is possible for a trusted authority to reveal the identity of the user if he or she is suspected of illegal activities. The next section analyze in detail the Generic Authentication Architecture from 3GPP. Combining SAML with the Generic Authentication Architecture, we propose a novel "generic mobile Web service platform" for M-Commerce. Various solutions have been proposed to address privacy concern in distributed networks; the Platform for Privacy Preferences is one of the popular proposal, though it has many desirable features, it is not easy to enforce it. We argue that this limitation can be managed in federated system such as the Liberty Alliance framework. In the final chapter we make the case for using timestamp based authentication protocol in mobile Web service on the ground of efficiency gain.

Contents

List of Figures

Acknowledgements

I would like to thank my supervisors Dr.M. Rajarajan and Dr. Veselin Rakocevic for their guidance and support with my work and for constant feedback on my progress. I also would like to thank Prof. Sanowar Khan who helped me to start the PhD in the first place and for his valuable suggestions on my research.

I am very grateful to Prof. Chris Mitchell from Royal Holloway, University of London for his valuable time that he give me and for the precious discussions which had significant positive impact on my work.

Finally I would like to than everyone in CM217 for making the work fun and very enjoyable.

Abbreviations

3GPP	3rd Generation Partnership Project
AKA	Authentication and Key Agreement protocol
AuC	Authentication Centre
BS	Base Station
BSF	Bootstrapping Server Function
GAA	Generic Authentication Architecture
GSM	General System for Mobile communications
HLR	Home Location Register
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
MO	Mobile Operator
MWSS	Mobile Web Services Security
NAF	Network Application Function
OASIS	Organization for the Advancement of Structured Information Standards
P3P	Platform for Privacy Preferences
SAML	Security Assertion Markup Language

SIM	Subscriber Identity Module
SOAP	Simple Object Access Protocol
SP	Service Provider
SSO	Single Sign On
UBR	UDDI Business Registry
UDDI	Universal Description, Discovery and Integration
UMTS	Universal Mobile Telecommunications System
WAP	Wireless Application Protocol
WCDMA	Wide Code Division Multiple Access
WSDL	Web Services Description Language
WSA	Web services architecture
WSS	Web Services Security

Introduction

Web services are becoming a model on which to build distributed Internet applications. These services predicate a set of standards that provide a simple and consistent way to access the functionality of diverse systems via the World Wide Web. Web services are new type of middleware for distributed computing. However, it is different from other middleware as it is; simpler, standards based, and more loosely coupled technology. It is because of the nature of Web services all middleware needs strong security practices. Web services are not just being used to integrate internal systems, but they also integrate data sources from outside the organization.

Since the early 1990s there has been a huge growth in wireless networks, because of this flexibility to both network operators and users. This has provided the users with not only "anytime, anywhere" network access, but also the freedom of roaming while networking. One of the major concerns in wireless networking is security. Although most of security threats against the wired network (e.g. threats against the TCP/IP stack) are equally applicable in wireless networks, a number of additional vulnerabilities makes wireless network more challenging

to secure. Such challenges includes, open wireless access medium, limited bandwidth, and system complexity.

Use of identity federation standards can increase security and lower risk by enabling an organization to identify and authenticate a user once, and then use that identity information across multiple systems, including external partner web sites. It can improve privacy compliance by allowing the user to control what information is shared, or by limiting the amount of information shared.

The notion of identity federation is extremely broad, and also evolving. It could involve user-to-user, user-to-application as well as application-to-application use-case scenarios at both the browser tier as well as the web services or SOA (service-oriented architecture) tier. Federated environment is used in Web services, where Single Sign On (SSO) concept plays an important role in building federated systems.

All the above topics play different roles when considering mobile Web services security system. Each chapter in this thesis is dedicated to one or more of these topics. The following section describes the structure of this thesis.

0.1 Motivation

Web Services Security is still in the process of being defined. Web Services look to be a useful paradigm for service access in mobile environments, but mobile environments have their own particular characteristics and requirements. As a result, it may be appropriate to look at Web Services security in the particular context of mobile networks to see if any additional features are required and also to see if solutions designed for fixed networks are still feasible in a mobile context.

Hence; the aim of this thesis is to investigate security systems to try use the best of both worlds to address security issues in mobile Web services environment.

This thesis looks at how to apply and/or utilize security features in wireless systems (mainly General System for Mobile communications (GSM) and Universal Mobile Telecommunications System (UMTS)), to secure mobile Web services systems and applications.

0.2 Aims and Objectives

The main aims and objectives of this research are driven from the motivation section earlier. These are outlined as follows:

- To identify the limitations/weakness of current Web services security technologies/techniques when applied in GSM/UMTS environment.
- To identify the limitations/weakness of current security system in GSM/UMTS technology, to be addressed when developing Mobile Web services system.
- Propose novel protocols to address the issues of authentications and privacy in mobile web services environment.
- Design and build prototype to test the various techniques and protocols developed by this research.

0.3 Structure of the Thesis

The thesis is structured into six chapters and a conclusion section;

0.3 Structure of the Thesis

In chapter one we review some core principles for designing security protocols. The chapter describes best practices and highlights core security techniques. We also list key security threats that must be considered when designing such protocols. These security design principles are used throughout the thesis.

Chapter two looks at Web services security. The chapter start by defining the basic building blocks of Web services such as XML and SOAP. This is followed by a review of the most used security technologies in Web services. WS-Security and common Web services security considerations are also discussed here.

Mobile security technologies are discussed in chapter three, specifically GSM and UMTS systems; the chapter mainly focus on the security features and limitations of both systems.

Chapter four looks at issues of federated system, after reviewing related work we identified that registration and authentication in addition to protecting user privacy as key security issues in federated environments. We propose various solutions and develop novel protocol that using GSM security to tackle the issue of protecting user's anonymity in mobile Web services environment.

The challenges of portable identity is one of the big challenges in mobile Web services, in chapter five we first review in some details SAML, which is a technology that was developed to address this issue of exchange authentication and authorization credentials across different security domains. And after reviewing related work.

Chapter six also looks at the issue of privacy in mobile Web services in some details. After reviewing some of the current technology standards and related work in the area. We identify the Platform for Privacy Preferences as key in helping to protect user's privacy. Though it has many desirable features, it is not

easy to enforce P3P policy in distributed system. We argue that this limitation can be managed in federated system such as the Liberty Alliance framework. We propose a system to enable privacy with P3P and Liberty Alliance.

In chapter seven we make the case for using timestamp based authentication protocol in mobile Web service on the ground of efficiency gain. After analyzing some of the limitations and issues such as clock synchronization, and trusted clock's and related work. We how timestamp based authentication can be used in mobile Web services.

Finally a conclusion is given, summarizing this research and identifies some challenges and direction for further research.

0.4 Publications and main contribution

This thesis contains previous research that has been published in the proceedings of a number of refereed conferences, as follows.

- Weerasinghe, D., Elmufti, K., Rajarajan, M. and Rakocevic, V. (2007) Securing electronic health records with novel mobile encryption schemes, Int. J. Electronic Healthcare, Vol. 3, No. 4, pp.395416.

- Dasun Weerasinghe, Kalid Elmufti, Muttukrishnan Rajarajan, Veselin Rakocevic. The International Workshop on Security and Privacy in Mobile HealthCare, "XML Security based Access Control for Healthcare Information in Mobile Environment, Innsbruck, Austria, November 29 - December 1, 2006.

- John A. MacDonald, Kalid Elmufti, Dasun Weerasinghe, M Rajarajan,

Veselin Rakocevic, Sanowar Khan. Proceedings for The 4th IEEE European Conference on Web Services, "A Web Services Shopping Mall for Mobile Users, 4-6 December 2006, Pages 99-108.

- Kalid Elmufti, Dasun Weerasinghe, M Rajarajan, Veselin Rakocevic, Sanowar Khan. 2nd International Conference on Pervasive Computing Technologies for Healthcare 2008, workshop "Connectivity, Mobility and Patients' Comfort (CMPC)" Timestamp Authentication Protocol for Remote Monitoring in eHealth, Tampere, Finland, Jan 29th, 2008.

- Kalid Elmufti, John A. MacDonald, Dasun Weerasinghe, M Rajarajan, Veselin Rakocevic, Sanowar Khan. The International Journal of Information Security "Mobile Web Services Authentication using SAML and 3GPP Generic Bootstrapping Architecture. (under review)

- Kalid Elmufti, M Rajarajan, Veselin Rakocevic, Sanowar Khan. International Journal of Mobile Communication, "GSM for mobile Single Sign-On to protect user privacy . (To be published 2008)

- Kalid Elmufti, Dasun Weerasinghe, M Rajarajan, Veselin Rakocevic, Sanowar Khan. The International Workshop on Security and Privacy in Mobile HealthCare, "Privacy in Mobile Web Services eHealth, Innsbruck , Austria , November 29 - December 1, 2006.

- K. Elmufti and C. J. Mitchell, "GSM for mobile SSO to protect user privacy", presented at WEWoRC 2005, The Western European Workshop on Research in Cryptology, Leuven , Belgium , July 2005.

- Kalid Elmufti, M Rajarajan, Veselin Rakocevic, Sanowar Khan, PREP

2005. "Authentication protocol for an M-Commerce application using OASIS SOAP message security standard, Nottingham , UK . April 2005.

Chapter 1

Protocol Design and considerations

1.1 Introduction

Cryptography has been used throughout the history of civilizations to protect secret messages. Though the techniques changed overtime, the core requirement has not changed [1]; how to send a message from A to B without anyone else able to discover the message. For a long time cryptography was mainly used by governments, military, and to some extent business/financial communities. It was not of a major concern to the average people. With the developments of the digital age and the growth of the Internet, however, many people started looking to cryptography to protect their privacy and important data, and cryptography became of greater importance.

This Chapter analyzes the main cryptographic techniques used throughout the Thesis. It also analyzes the design principles and the methodologies used to

develop the security protocols in the Thesis. Section 1.4 discusses the major security challenges and how to verify the credibility of our design, and finally the Chapter lists some of the main security threats in the domain of Web services.

1.2 Terminology and Definitions

The following is an informal introduction and an overview of definitions to some of the well established techniques used in digital cryptography. The following terminologies and definitions are used throughout the Thesis. [1, 2] present a more detailed introduction.

1.2.1 Random Numbers and Nonce

Random Number or Nonce can be defined as a finite sequence of bits that is used only once within a given context. A nonce is a value used not more than once for the same purpose. Nonces are used to protect from a reply attack. Generally, they are introduced in cryptographic protocols to provide freshness. The random numbers used as nonces have to be hard to guess. However, a simple type of a nonce could be a counter where the parties involved would keep track of all the integers used. Another popular type of nonce is a *timestamp*. a Timestamp can be used to limit the period of validity of the message.

The security of many cryptographic systems depends upon the generation of unpredictable quantities such as the secret key in DES [3] or the public/private key pair in RSA [4]. The quantities generated must be of sufficient size and be random in the sense that the probability of any particular value being selected must

be sufficiently small to preclude an adversary from gaining advantage through optimizing a search strategy based on such probability.

1.2.2 Key Derivation Functions

Key derivation functions are used to derive cryptographic keys, usually from secret values such as passwords or other cryptographic keys. A Key derivation function can also be used to ensure that the derived keys have other desirable properties, such as avoiding "weak keys" in some specific encryption systems. Another common use is in deriving session keys from long term secret key such as in GSM using the A8 derivation function [5].

1.2.3 Encryption

Encryption is the process of transforming information to make it unreadable to anyone except those possessing special knowledge, also known as a key. The message in its human readable form is referred to as plaintext (P). The process of transforming or hiding this message is known as encryption (e), and the resulting message is known as ciphertext (C). The encryption function e operates on P to produce C:

$$e(P)=C$$

The reverse process is known as description (d), where the decryption function d operates on C to produce P:

$$d(C)=P$$

The processes of encryption and decryption are achieved through the use of cryptographic algorithms also known as ciphers. All modern ciphers use a key (K), where the value of this key affects the cryptographic algorithms:

$$\begin{aligned} \mathbf{e(K,P)=C} \\ \mathbf{d(K,C)=P} \end{aligned} \tag{1.1}$$

There are two types of encryption depending on how the cryptographic key(s) are used: symmetric encryption and asymmetric encryption

Symmetric encryption

The distinguish feature of symmetric encryption is that only one key is used for both encryption and decryption processes, and this key must is shared between the sender and the receiver. There are many techniques for deploying symmetric encryption. The one most frequently used is known as block cipher. A block cipher algorithm takes a block of plaintext and a secret key and produces block ciphertext. The decryption will take the ciphertext with the same secret key to reproduce the plaintext. Some of the most used block ciphers are the Data Encryption Standard (DES) [3] and triple DES, the Advanced Encryption Standard (AES) [6], and the KASUMI cipher algorithm [7] which is important in a mobile context. Stream cipher is another important type of symmetric encryption. It

is different from block cipher in that it encrypts data bit by bit. Stream cipher uses what's known as keystream generator which takes a secret key as the input and generates a pseudo random sequence of bits as the output. This sequence is EXORed with the plaintext bit sequence to produce the ciphertext. The same process is repated for the decryption. An example of stream cipher is the A5 algorithm used in GSM [8].

Asymmetric encryption

The main issue with symmetric encryption is key distribution, since the sender and the receiver must share a common key. This is possible with a small number of users and in closed environments but in large distributed networks, such as the Internet, this is a major issue. The problem was addressed in 1970s by Whitfield Diffie and Martin Hellman with their Diffie-Hellman cryptographic system [4], the start of asymmetric cryptography. Also known as public key cryptography, it is a process where each user gets a pair of keys, called the 'public key' and 'private key' (or 'secret key'). The public key is published while the private key is never revealed. In the encryption, the recipient's public key is used to encrypt the plaintext and the recipient's private key is used to decrypt the ciphertext. The most know public key algorithm is the RSA scheme [4]. Its important to note that, although asymmetric encryption solves the issue of key distribution effecting symmetric encryption, managing these key pairs is not an easy task. Furthermore, asymmetric encryption tends to be much more computationally intensive than symmetric encryption [9] and this can be an issue for devices with limited computing power such as mobile phones.

1.2.4 Hash Functions

A hash function (h) takes a message of any length and gives an output of a short fixed length value known as digest (d), such that for a message (m):

$$d=h(m)$$

The main property of any hash function is that it must be a one-way function, that is its infeasible to obtain the message m from the digest d. Another important property of hash functions is that they are usually efficient to compute. Hash functions are used to protect the integrity of messages, so after computing the digest the recipient can check the integrity of the message received by recomputing the digest and compare the two digests. Typically this is done using the Message Authentication Code (MAC). The MAC is computed by encrypting the digest with a secret key so then the MAC can be attached to the message and sent to the recipient, who then needs to recompute the MAC and compare it with the one attached to the received message to verify its integrity. Hash functions are also used in digital signature.

1.2.5 Digital Signature

A digital signature of a message is an output of a mathematical process that is dependent on some secret known only to the signer, and, additionally, on the content of the message being signed. The process of generating the digital signature is very much like the generating MAC. The main difference is that a private key (instead of a symmetric secret key)is used, which has a corresponding

public key that the recipient possesses.

Signatures must be verifiable; and this is possible by anyone possessing the signer's public key. In MAC, the verification process is just recomputing the MAC, however in digital signature a special verification function is used, which takes as input the signature, the message, and the public key, the output is a boolean indication as to whether the signature is valid or not. If a dispute arises as to whether a party signed a document (e.g. a lying signer, or a fraudulent claimant), an unbiased third party should be able to resolve the matter equitably, without requiring access to the signers secret information.

Digital signatures have many applications in information security, including authentication, data integrity, and non-repudiation. One of the most significant applications of digital signatures is the certification of public keys in large networks. Certification is a means for a trusted third party to bind the identity of a user to a public key, so that at some later time, other entities can authenticate a public key without assistance from a trusted third party [2].

1.3 Design Principles and Methodology

In later chapters we propose novel security protocols and schemes. The design principles of these protocols are defined in this section. However, the main purpose of proposing these protocols is to demonstrate and introduce new concepts in mobile Web services security, in particular for authentication and privacy purposes. Therefore, there was no need to use formal methodologies to prove the accuracy of these protocols, but on the other hand we wanted our proposed protocol to be reasonably secure and to address most of the known security issues.

For these reasons we found the design principles presented in "Prudent Engineering Practice for Cryptographic Protocols" [10] by Abadi and Needham, and "A Logic of Authentication" [11] by Burrows and Abadi and Needham, very relevant and suitable to be used as general guides for our protocol design principles and methodology.

1.3.1 Notation and Naming

- Each entity used in the proposed protocols is referred to it by its descriptive name (e.g. Mobile Device or Service Provider) to encapsulate some of its characteristic such device limitation or level of trust.

- The mobile device sometime is referred to as the 'User' as a reference that the end user is using the mobile device.

- The signature on data X using private key K is written $s_K(X)$.

- In asymmetric encryption algorithm, the encryption of data X using public key P is written $e_P(X)$.

- In symmetric encryption algorithm, the encryption of data X using secret key K is written $e_K(X)$.

1.3.2 Assumptions

The following are assumed true throughout the Thesis. There are other specific assumptions which are presented in the relevant sections.

- Security Algorithms: all standard security algorithms used in our protocols

are assumed to be 'secure' and they operate as expected, unless otherwise stated.

- All entities/principals know exactly how the cryptographic algorithms in use work.

- Mobile Operators are trusted: all Mobile Operators (also referred to as Network Operators) are trusted by all other elements who interact with them directly or indirectly. The trust here indicates that the Mobile Operators will act as expected and that their system can't be compromised.

- Mobile Operators Clock: it is assumed that the Mobile Operators run trusted clock, whose time can not be changed by anyone other than the operators.

- The terms Encryption/Decryption, Signature/Verification are used in their basic definitions as described in the previous section, and that no special characteristics is required unless otherwise specified.

- All mobile devices have limited memory storage and computing power, although enough it is possible to run PKI on them [9], we try to avoid it as much as possible.

- In our security analysis we assume the existence of a malicious entity, an attacker or a hacker, who can monitor the network traffic, create fake messages and introduce them in the traffic.

1.3.3 Core Principles

The security protocols proposed in this thesis present new mechanisms on how to develop security protocols which provide means of authentication and protection of user privacy in the context of Mobile Web services. While when developing these protocols we have addressed other security concerns such as confidentiality, integrity, and non-repudiation, it is by no mean comprehensive and other important security threats such as denial of service and social engineering were not considered in details. The protocols developed are considered suitable for many applications, such as M-Commerce, and health care however more testing is recommended before deploying in real applications; as they have not been tested in real systems.

There have been many proposals and formalisms on how to design cryptographic protocols, although sometimes useful, these formalisms do not of themselves suggest design rules and hence they are not directly beneficial in preventing trouble [10]. since the aim of this work was mainly to demonstrate the possibilities of integrating Mobile and Web services systems and how can this improve security; we have found the methodology and the design principles presented in "Prudent Engineering Practice for Cryptographic Protocols" [10] by Martin Abadi and Roger Needham are the most suitable for our purposes. The following is a summary of main principles from [10] related to our designs:

Principle 1 Every message should say what it means; the interpretation of the message should depend only on its content. It should be possible to write down a straightforward English sentence describing the content though if there is a suitable formalism available that is good too. In other words, the

messages in the protocols only depend on the information presented in the protocol or already in the possession of the recipient.

Principle 2 The conditions for a message to be acted upon should be clearly set out so that someone reviewing a design may see whether they are acceptable or not.

Principle 3 If the identity of an entity or a principal is essential to the meaning of a message, it is prudent to mention the principal's name explicitly in the message. This principle is of greater importance in our proposed security protocols, as there are general assumptions associated with the various entities or principals for example the Mobile Operator is always a trusted entity or that the Mobile Device has limited capability in terms or memory storage or computing power. For that reason we used naming notation that distinguishes each entity for the others.

Principle 4 Be clear about why encryption is being done. Encryption is not wholly cheap, and not asking precisely why it is being done can lead to redundancy. Encryption is not synonymous with security, and its improper use can lead to errors.

Principle 5 When a principal signs material that has already been encrypted, it should not be inferred that the principal knows the content of the message. On the other hand, it is proper to infer that the principal that signs a message and then encrypts it for privacy knows the content of the message.

Principle 6 Be clear what properties you are assuming about nonces. What may do for ensuring temporal succession may not do for ensuring association and

perhaps association is best established by other means. This is also true when using random numbers or timestamps.

Principle 7 If timestamps are used as freshness guarantees by reference to absolute time, then the difference between local clocks at various machines must be much less than the allowable age of a message deemed to be valid.

Principle 8 The protocol designer should know which trust relations his protocol depends on, and why the dependence is necessary. The reasons for particular trust relations being acceptable should be explicit though they will be founded on judgment and policy rather than on logic.

1.4 Protocols Security threat Analysis

While it is a matter of current research to devise a satisfactory list of properties for defining secure protocols in the context of modern distributed networks, all of these properties implicitly assume the existence of a malicious entity, who can monitor and effect the network traffic. While history tells us that security was already an important issue in ancient times [1], now in this digital age the problem is on much bigger scale.

The vast majority of security protocols for computer networks are based on cryptography. These are sequences of messages, possibly encrypted, exchanged between different entities in order to make their subsequent communication secure. Messages include entity names, cryptographic keys, random numbers, timestamps, concatenations of those components and ciphertexts obtained from them. Each protocol attempts to achieve certain goals at the time of its completion, namely the set of properties that define security.

1.4 Protocols Security threat Analysis

In the following subsections we list the main categories of possible security threats.

1.4.1 Entities Security

It is assumed at the start of any proposed protocol that all devices used in the system are free from any malicious programs such as viruses or back-doors (unless specified). Since the beginning of computer networks viruses and other malicious programs have been developed to exploit these network entities or network devices. There are many techniques to attack such device. Such techniques includes; infecting network devices with viroses, modifying or stopping certain softwares from behaving properly, installing worms to spread viruses, installing trojan horse and back-doors to monitor device activities and open protected ports [12].

This is not different in mobile phones. Most current generation of mobile phones have the ability to download 'code' such as ring tones, personal organiser, or a game. This code needs to interact with the device operational system to execute it; if someone can hide a virus inside these codes it can easily infect such devices [13]. The subject of entities security is out of the scope of this Thesis.

1.4.2 Communication Security

Because of the nature of the protocol security design, communication security analysis is very important, and most of the security analysis of the proposed protocols in this Thesis is based on communication security. There are two types of communication attacks: passive attacks and active attacks.

Passive attack

In this type of attack the attacker is just monitoring the network traffic using techniques such as footprinting and scanning [12]. The main aim of passive attacks is to gather information. Sometimes, this information is very valuable by itself (for example just knowing that A is talking to B could be of significance). Alternatively, the information can be used later to launch active attack.

Passive attacks are very hard to protect against. This is made even harder in mobile networks where most of the network traffic is travelling on air, easy for anyone to intercept.

Passive attacks are of major concerns when trying to protect user privacy. An attacker capable of monitoring network traffic could compromise the user's privacy. Digital cryptography plays important part to protect against such attacks. Some data, however, (e.g. packet headers) can't be encrypted for routing reasons and other techniques must be used to protect user privacy, such as changing the user identity.

Active attack

There is a wide range of attacks that can be classified as active attacks, varying from guessing a password, replay attack, denial of service, and many others [12]. In active attacks, the attacker will 'actively' interact with the system to expose its weaknesses or to cause some damage. Different types of attacks have different level of risks depending on the system and/or the application of concern, therefore it is important when developing/reviewing security requirements that they are aligned with the system or the application main objectives.

1.4 Protocols Security threat Analysis

In this Thesis the focus is on security of communication protocols, prevention or limitation of attacks that could compromise authentication, confidentiality, and integrity of the system in addition to improving user privacy. There are many techniques to compromise weaknesses in security protocols, some if the common techniques includes:

Security Keys One of the biggest challenge of any security protocols is the management of security keys, from keeping these keys secret to key distribution of symmetric keys. Also to make sure that keys are not used after their lifetime.

Mutual Authentication This is when only one entity is required to authenticated itself to the other entity, though it is not a requirement by itself in many communication protocol, many serious attacks exploit this property that exist in many protocols such as the "false base station attack" in GSM networks [14].

Timestamps and other nonces In appropriate use of timestamps and nonces such as sequence numbers can exploit the system to replay attacks.

Security Algorithms Some serious security attacks exploit some weaknesses in security algorithms that are used in the various cryptographic operations. Care must be taken on selecting which algorithm to be used when developing security solution. Though the issue of security algorithm is an important one, it is out of the scop of this thesis, and only few references will be mentioned on this topic.

1.4.3 Privacy Considerations

The scope of digital privacy is very big, and it is a very difficult challenge to protect against all possibilities. For example for some users all what they care about is that none knows their financial data. Other users may not want to reveal their true identity online, and in more special cases such as in the military they may not want anyone to know that they are having a communication at all.

There are different levels of privacy protection, Garg and Wilkes [15] have classified privacy into four categories as follows:

Level 0: None No privacy security enabled, anyone with proper monitoring equipment could monitor the communication.

Level 1: Equivalent to wireline Here some level of protection is available and is harder to launch than level 0, such as using simple username/passowrd protection.

Level 2: Commercially secure The need for privacy in such application is obvious and cryptography is very likely to be used to protect the confidentiality of such communications, at level 2 more sophisticated cryptographic techniques would be used to protect the data privacy and also the use of some privacy protection protocol such as P3P.

Level 3: Military and governments secure The highest level of possible privacy and security protection, because of the sensitivity of information in such applications.

The main issue in digital privacy is that in many cases it works against security. A common example could be in healthcare, in a scenario where a user

would like to access his healthcare records online. In this case the user needs to give his true identity to the healthcare authentication server to obtain a username/passowrd to access the system. Even though this action was meant to provide only the user with secure authentication, the fact that he gives his details online and these security credentials are stored in a remote server bring a risk to his privacy.

One of the most common techniques to exploit user privacy is by monitoring user's 'identity'. As mentioned earlier in distributed network environments (e.g. Web services environment) protecting user privacy is not easy, therefore guaranteeing identity pseudonymity, such that the identity does not include any personally identifying information is a desirable property. Though total 'unlinkability' may be feasible in theory it may not be possible (or desirable) in practice. Protecting personal information is harder to do if for example the user expecting physical goods as the user address is required, in the other hand this could be possible when providing goods in digital format.

1.5 Summary

There is a clear need to protect digital communications. This is usually achieved through secure communication protocols. In this chapter we listed the main definitions and terminologies used in this Thesis and reviewed key cryptographic technologies. A list of assumptions with core design principles were presented which are used throughout the thesis, and finally a an overview of protocol threat analysis and security consideration for Web services were given.

Chapter 2

Web Services Security

2.1 Introduction

Web Services were developed to improve the functionalities of web applications by providing seamless integration of systems and services and an increased number of service options. Web Services gained a lot of momentum since the term was introduced in 2000. Many software vendors have started to produce various Web services products, each with their own adopted definition of what is meant by the term Web Service. Currently the most adopted definition is the one defined by the Web Service Architecture working group of the World Wide Web Consortium [16]: "A Web Service is a software system designed to support interpretable machine-to-machine interaction over a network. It has an interface described in a machine processable format specifically WSDL. Other systems interact with the Web Service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards".

2.1 Introduction

To a business person, Web Service is mainly about integration; integrating application functionality within an organization or integrating applications between business partners. From a technical perspective Web services is trying to address the fundamental challenge of distributed computing. A Web Service is nothing more than a collection of one or more related operations that are accessible over a network and are described by a Service description. What is new in Web services is that they combine the loosely coupled component-based approach to software development using open standards to achieve ubiquitously deployed infrastructure capabilities.

The industry is embracing and implementing the Web services model mainly to reduce cost, ensure compliance, and increase security. However, this will never be realized unless they are proven to be reliable, available and have the appropriate level of security.

As the Internet has developed into an internationally accepted basis for e-commerce it has seen an evolution of security practices and challenges [17]. This accounts for the exponential growth of hackers, virus developer, and others types of attacks. Web services uses web technologies, which results in the fact that many of the potential attacks against web sites are of concern for web services.

The security issues that apply to the Web Services are similar to those in other technology solutions and systems; the main difference is that in the Web services organizations must focus on building secure application from ground up to protect their data in storage and transit. This chapter looks at the building blocks of Web services and the main technologies and standards used to secure Web services.

2.2 Web Services Basics

The concepts behind Web services are complex [18], and many people disagree about what Web services are and what they mean to the computing industry. However most agree that a Web Service is any piece of software that makes itself available over the Internet and uses a standardized XML messaging system. Because all communication is in XML, Web services are not tied to any one operating system or programming language. Another definition of Web services [19] is: "Web services are self-contained, self-describing XML based software components, they provide architecture for distributed loosely coupled services that can be published, located and invoked remotely over Internet protocols, by services clients written in a different language".

Web services is built using four building blocks; XML, SOAP, UDDI, and WSDL. The following subsections look at these building blocks.

2.2.1 XML

XML (eXtensible Markup Language) was defined by the W3C as an open standard technology, and it became W3C Recommendation in 1998. XML defines a standard way to structure information for describing, storing and exchanging data via Web services. There are no predefined semantics and because of that the definition of data must be agreed in advance between the communicating parties.

The XML specifications defined by the W3C in [20] specify the XML syntax that all XML documents must follows. These documents that follows the correct syntax are known as *well formed* XML documents. An XML document optionally can reference another document that defines the XML document structure (known

as Document Type Definition (DTD) or Schema), if the XML document adheres to the structure defined in the DTD/Schema then the XML document becomes *valid*.

Web services uses XML. The following is a set of XML technologies that are the foundation of Web services [16]:

- XML Instances: the rules for creating syntactically correct XML documents.
- XML Schema: a standard that enables detailed validation of XML documents as well as the specification of XML datatypes.
- XML Namespaces: definitions of the mechanisms for combining XML from multiple sources in a single document.
- XML processing: the core architecture and mechanisms of creating, parsing, and manipulating XML documents from programming languages as well as mapping Java data structures to XML.

Note: the above are not only XML technologies exist, other technologies such as XPointer/Xlink, Resource Definition Framework, XPath are less relevant to Web services and was not introduced. Other technologies such as XML Signature and XML Encryption will be discussed in more details later on.

2.2.2 SOAP

Simple Object Access Protocol (SOAP) is one of the most common standards used to deliver Web services. Initially developed by representatives from DevelopMentor, Userland Software and Microsoft [18], now SOAP 1.2 is a W3C

Recommendation [21]. SOAP was created as a way to transport XML from one computer to another via a number of standard transport protocols. SOAP is defined using XML, and it provides a simple consistent and extensible mechanism that allows one application to send an XML message to another.

SOAP provides an envelope into which an XML message is placed. This envelope is just a container to hold XML data. the idea is for SOAP to create a uniform container that can then be carried by a variety of transports. The SOAP model separates between infrastructure processing and application processing [22]. Figure 2.1 shows the basic structure of a SOAP message.

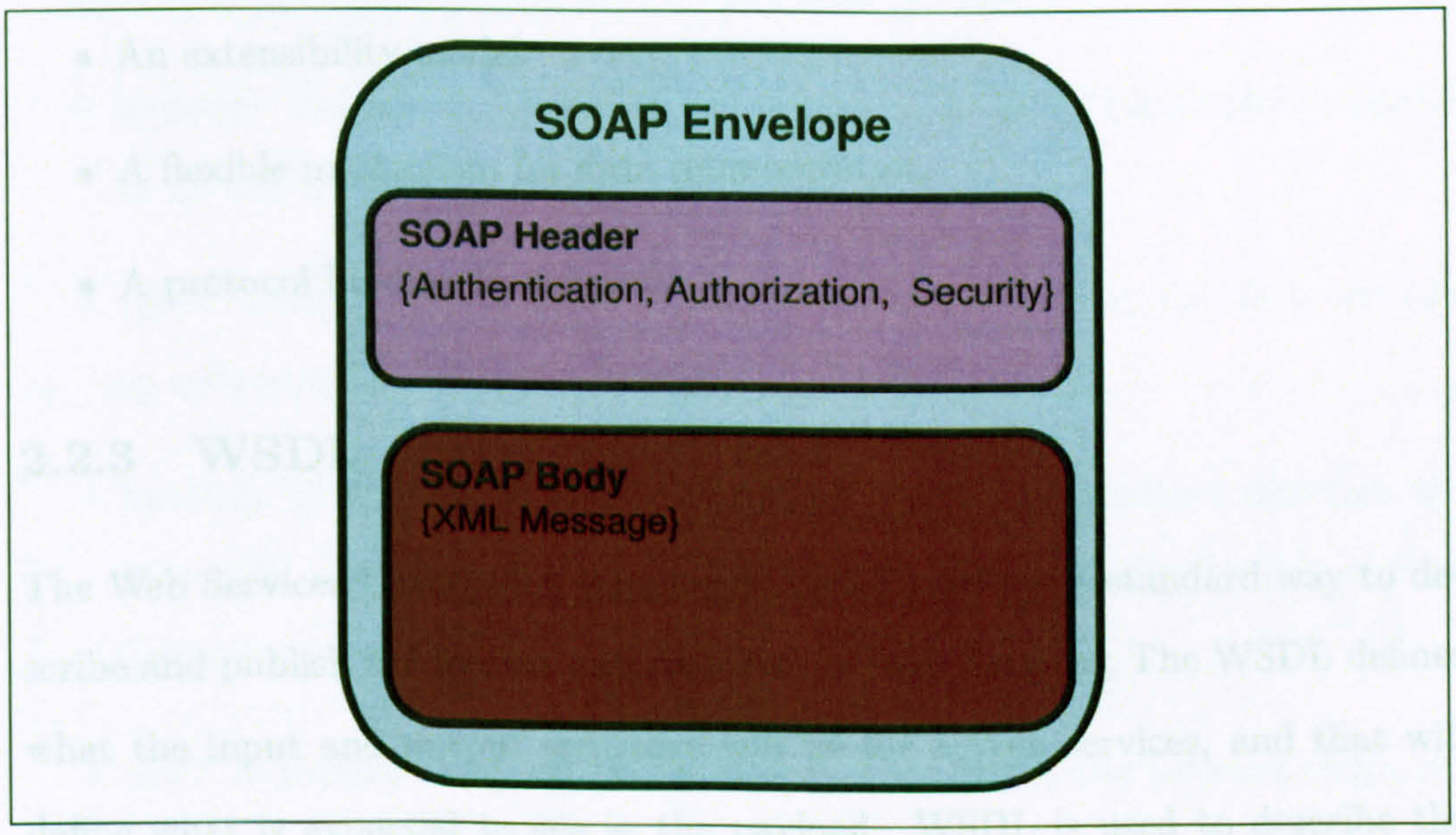

Figure 2.1: Basic SOAP structure

The SOAP envelope contains:

- SOAP Header: contains information about the SOAP message. This information is used to manage or secure the package.
- SOAP Body: contains the message payload. This information is being sent

from one application to another. It could be a full document or a description of remote procedure call information.

SOAP is very important as it is the industry standard for cross platform XML distributed computing, providing:

- A mechanism for defining the unit of communication.
- A processing model.
- A mechanism for error handling.
- An extensibility model.
- A flexible mechanism for data representation.
- A protocol binding framework.

2.2.3 WSDL

The Web Services Description Language (WSDL) defines a standard way to describe and publish the format and protocols of Web Services; The WSDL defines what the input and output structure will be for a Web services, and that will define what is expected to see in the payload. WSDL is used to describe the message syntax associated with the invocation and response of a Web Service. WSDL is a W3C Note since March 2001 [23]. WSDL description describes three fundamental properties of a Web Service [16]:

- What a Service does: the operations the Service provides, and the data needed to invoke them.

- How a Service is accessed: details of the data formats and protocols necessary to access the Service's operations.

- Where a services is located: details of the protocol specific network address, such as URL.

WSDL is defined in XML. The WSDL schema defines several major elements in the language:

- portType: A Web Service's abstract interface definition where each child operation element defines an abstract method signature.

- message: Defines the format of the message, or a set of parameters referred to by the method signatures.

- types: Defines the collection of all the datatypes used in the Web services as referenced by various message part elements.

- binding: Contains details of how the elements in an abstract interface are converted into a concrete representation in a particular combination of data formats and protocols

- prot: Expresses how a binding is deployed at a particular network endpoint.

- Service: A named collection of ports.

2.2.4 UDDI

The Universal Description, Discovery and Integration (UDDI) enables developers and businesses to publish and locate Web services on the network. It stores

WSDL files which define Web services interfaces by defining and implementing a registry for finding Web services. This Web services registry is communicated by SOAP and is intended to act as a search engine for services.Web services will use UDDI to publish services and the Web Service client will use the UDDI registry to obtain location, description and binding information from WSDL files stored in the registry. The purpose of UDDI is to facilitate Service discovery both at design time and dynamically at runtime. UDDI version 3 became an OASIS Standard in February 2005 [24].

There are two primary types of UDDI registries: public and private. The public registry is referred to as the UDDI Business Registry (UBR). The private registry is more widely used and usually has a specific purpose.

2.2.5 Web Services Driving Committees

There is a wide diversification in the businesses and technical committees that are driving standards for Web services. In fact one of the main reasons for the success of Web services is the joint effort from the business and technical world to standardize Web services. The following is a list with the main organizations and committees:

- W3C: The World Wide Web Consortium. Its main goals are to provide standards that keep the Internet accessible, easy to use, secure, innovative, and to encourage the expansion and increase of creativity in web site development. (www.w3c.org)

- IETF: The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers

concerned with the evolution of the Internet architecture and the smooth operation of the Internet (www.ietf.org)

- OASIS-Open.org: The Organization for the Advancement of Structured Information Standards is a non-profitable global consortium dedicated to web standard. WS-Security is an OASIS standard.

- WebServices.org: An organization that was formed to gather information regarding Web services, and is dedicated to provide these information in an organized, unbiased manner. (www.webservice.org)

- Web Services Architect: An organization that is dedicated to gather information about Web services. They provide technical reports specifically regarding the architecture that supports web services. (www.webservicesarchitect.com)

2.3 Web Service Security Basics

No new technologies were invented to secure Web services. Instead, existing security technologies were proposed to be applied for message level security [21]. This section analysis the main technologies used to secure Web services systems.

2.3.1 XML Digital Signature

Digital Signature provides the key functionality needed to promote the trusted exchange of data between web services. The specifications for digital signature in the XML environment were developed by a joint project between the W3C and the IETF [25]. XML Signature is a core foundation for Web services security and it was built on top of digital signature technology.

The ability to digitally sign a document is not a new concept and many methodologies can be used to apply digital signatures to a whole XML document such as RSA Public Key Cryptography Standard. However, what XML digital signature specifications allow is the ability to sign only specific portions of a document and to be able to attach multiple signatures that apply to different portions of the same XML document.

In XML signature the signature is not directly applied to the original or the digest form of the data. Instead the digital content or contents to be signed are first digested and placed in an XML element. This element is then digested and cryptographically signed.

The digital signature is represented in XML and is identified by the root <Signature> element. There are four ways of relating the data objects and their XML signature:

- Enveloping signature: the data object is embedded within the XML signature, and the <Signature> element becomes the parent of the local data object.

- Enveloped signature: the data object embeds the XML signature within itself, and the <Signature> element becomes the child of the original data object.

- Detached signature: the data object resides within the same XML document containing the XML signature. The <Signature> element is a sibling of the original data object.

- Detached signature and external reference: the data object resides external to the document containing the XML signature The <Signature> element

carries a reference to the external original data object [17, 25].

Issues with XML signature

One issue with digital signature in general is that both the signature and the validation must occur on the same bits. The validation of a simple ASCII text could fail because for example there are three different end-of-line sequences. If the line ending sequence of the signed text changes form the one convention to another, the validation would fail. The XML Signature working group has recommended three specific principles when using Transforms with XML Signature:

- Only What Is Signed Is Secure
- Only What Is Seen Should be Signed
- See What Is Signed

As the original data may change after the signature has been created, and the 'old' signature may still be valid. In addition, the generation of an XML signature involves many processing steps and many algorithms. The strength of a signature depends on the weakest among the chain of the processing steps. There are possibilities to introduce unplanned behaviour of algorithms, such as excessive memory requirements by passing wrong parameters, which could be a major issue in mobile web services applications where mobile devices used could have limited memory capabilities [25].

2.3.2 XML Encryption

XML Encryption was developed by the W3C XML Encryption work group [26] to define a standard mechanism for encrypting XML entities.

The main goal of XML Encryption is to ensure end-to-end confidentiality of messages. XML Encryption allows for different parts of a document to be encrypted with different keys and therefore to be seen by different recipients.

One of the main goals to develop XML Encryption was to be able to encrypt specific parts of a document. Another goal was to apply multiple encryptions to different parts of the same document.

Additional important part of the XML encryption specifications was the XML representation of encrypted data. This is specially important for web services as WS standard protocols such as SOAP and WSDL are based on their use of XML format and therefore it was important to represent the encrypted content in XML format. The encrypted data in XML format is identified by an <EncryptedData> element or by an <EncryptedKey> element. The <EncryptedData> element is used to represent any encrypted content other than the encryption key, which is represented by the <EncryptedKey> element.

Issues and security considerations with XML Encryption

It is possible to introduce cryptographic vulnerabilities when combining XML encryption and signatures, this is because the digest value of the encrypted signed document will still appear in clear text in a <...:Reference> child of the <...:Signature> element, which may reveal information over encryption data [17].

Furthermore, the XML Encryption specifications permits recursive processing. For example an EncryptedKey may depend on another EncryptedKey, which

in turn may depend on another EncryptedKey and so on. That could lead to a denial of Service vulnerability.

2.3.3 Security Assertion Markup Language (SAML)

SAML is the XML based security standard created to enable portable identities and the assertion of these identities. SAML is used to exchange authentication and authorization credentials across different security domains. Because SAML is XML-based it is not tied to any transport protocol or platform, also it is not dependent on any central certificate authority to issue certificates and this is very important in web services environment.

SAML V1.0 became an OASIS standard in November 2002. SAML V1.1 followed in September 2003 and has seen significant success, gaining momentum in financial services, higher education, government, and other industry segments. SAML has been broadly implemented by all major Web access management vendors. SAML support also appears in major application server products and is commonly found among Web services management and security vendors. SAML V2.0 builds on that success.

SAML V2.0 unifies the building blocks of federated identity in SAML V1.1, and was developed by the Security Services Technical Committee of OASIS. The SAML V2.0 OASIS Standard specification set was approved on 15 March 2005 [27].

A more detailed description of SAML including its main components, the motivation behind SAML, and some design issues will be discussed in chapter five when dealing with federated environment.

2.4 WS-Security

In April 2002 three software giants; Microsoft, IBM, and VeriSign submitted a proposal for Web Services Security Roadmap [28] to OASIS; since then the OASIS Web Services Security Technical Committee have used this Roadmap to define a specification for WS-Security and to form the necessary technical foundation for higher level security services.

WS-Security is an overarching conceptual model that abstracts different security technologies into "claims" and "tokens". The WS-Security standard is a specification specifically created for using various security technologies in the context of a SOAP message [22].

Developed by OASIS, WS-Security defines a SOAP extensions providing quality of protection through message integrity, message confidentiality, and message authentication [29]. WS-Security targets the problem of securing the Web Service message itself; this is different for example than SSL/TLS which creates a secure pipe between two servers though which messages travel.

WS-Security mechanisms can be used to accommodate a wide variety of security models and encryption technologies. The WS-Security mechanisms provides a general mechanism for associating security tokens with messages. The specification does not require a specific type of security token. It is designed to support multiple security token formats. WS-Security describes how to encode binary security tokens. The specification describes how to encode X.509 certificates and Kerberos tickets. Additionally, it also describes how to include opaque encrypted keys. The WS-Security specification defines an end to end security framework that provides support for intermediary security processing. Message integrity

is provided using XML Signature in conjunction with security tokens to ensure that messages are transmitted without modifications. The integrity mechanisms can support multiple signatures, possibly by multiple actors. The techniques are extensible such that they can support additional signature formats. Message confidentiality is granted by using XML Encryption in conjunction with security tokens to keep portions of SOAP messages confidential. The encryption mechanisms can support operations by multiple actors.

WS-Security is a foundational technology for a set of follow-on Web services security standards [22]:

- **WS-Policy**: Defines how to express capability and constraints of security policy
- **WS-Trust**: Describes the model for establishing both direct and broker trust relationships.
- **WS-Privacy**: Enables users to state privacy preferences and Web services to state privacy preferences and Web services to state and implement privacy practices
- **WS-SecureConversation**: Describe how to manage and authenticate message exchanges between parties
- **WS-Federation**: Describe how to manage and broker the trust relationships in a heterogenous federated environment.
- **WS-Authorization**: Defines how Web Services manage authorization data and policies.

Figure 2.2 shows the Web Service security framework. WS-Security defines a SOAP security header that provide a common format for security in a SOAP message. There are three major elements make up a SOAP security header: security tokens, XML Encryption, and XML Signature.

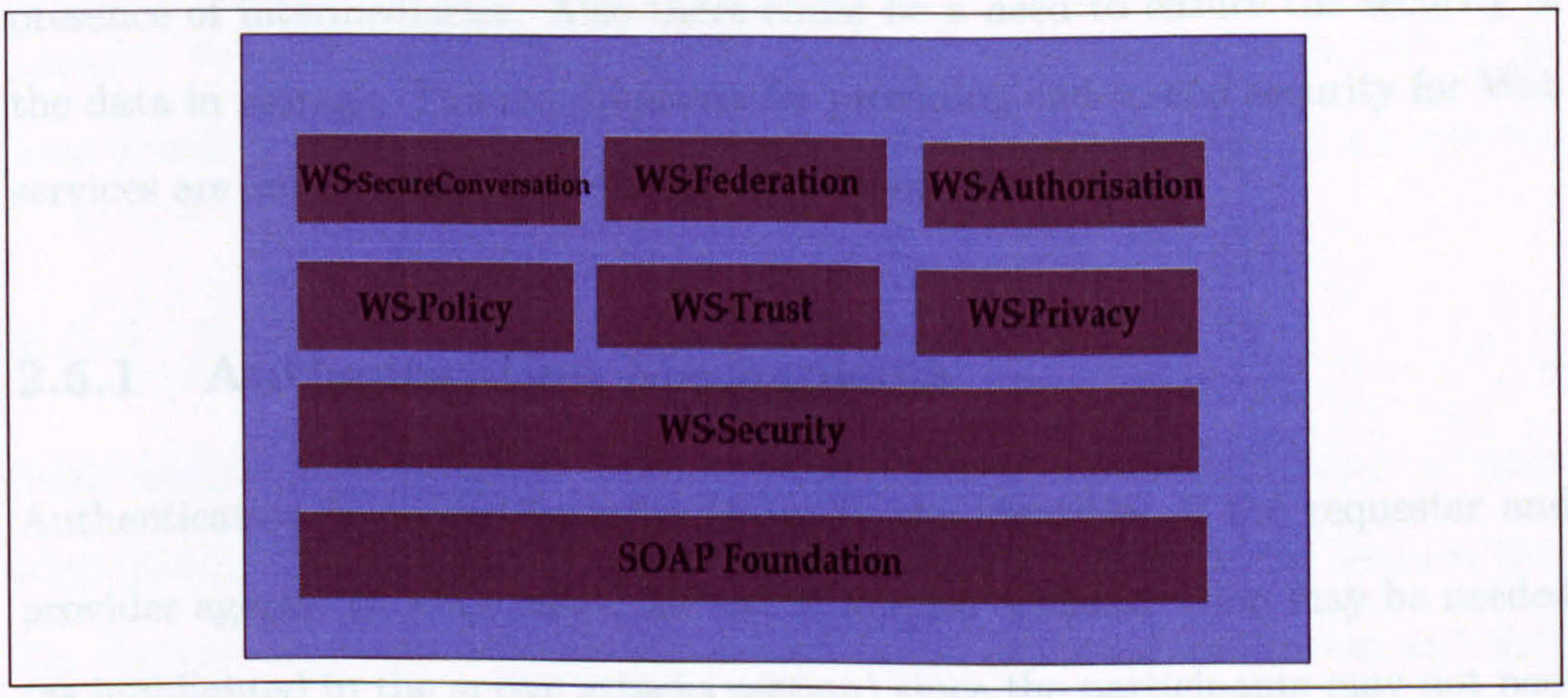

Figure 2.2: The Web Service security framework

The security tokens are pieces of information used for authentication or authorization. The current standard is WS-Security 1.1 [29] which consists of seven documents: item WS-Security Core Specification 1.1, Username Token Profile 1.1, X.509 Token Profile 1.1, SAML Token Profile 1.1, Kerberos Token Profile 1.1, Rights Expression Language (REL) Token Profile 1.1, and SOAP with attachments (SWA) Profile 1.1.

2.5 Security considerations for Web Services

A Web service can be defined as a middleware that uses the web infrastructure. It integrates applications inside and outside the organization. Distributed computing has always had a challenging set of security issues.

There are many security challenges for adopting Web services. At the highest level, the objective is to create an environment, where message level transactions and business processes can be conducted securely in an end-to-end fashion. There is a need to ensure that messages are secured during transit, with or without the presence of intermediaries. Also there could be a need to ensure the security of the data in storage. The requirements for providing end-to-end security for Web services are summarized in the following subsections.

2.5.1 Authentication Mechanisms

Authentication is needed in order to verify the identities of the requester and provider agents. In some cases, the use of mutual authentication may be needed (as highlighted in the active attacks section) since the participants may not necessarily be directly connected. For example the participants might be the initial requester and an intermediary. Depending on the security policy it may be possible to authenticate the requester, the receiver or to mandate the use of mutual authentication. Several methods can be used to authenticate services. Techniques include: passwords, one time pass and certificates. Password-based authentication must use strong passwords. Password authentication alone may be insufficient. Based on vulnerability assessment it may be necessary to combine password authentication with other authentication and authorization process such as certificates, and Public Key Infrastructure (PKI) [4] (in later chapters we proposed novel mechanisms based on GSM/UMTS security).

2.5.2 Authorization

Authorization is needed in order to control access to resources. Once authenticated, authorization mechanisms control the requester access to appropriate system resources. There should be controlled access to systems and their components. Policy determines the access rights of a requester. The principle of the least privileged access should be used when access rights are given to a requester.

2.5.3 Data Integrity and Data Confidentiality

Data integrity techniques ensure that information has not been altered, or modified during transmission without detection. Data confidentiality ensures that the data is only accessible by the intended parties. Data encryption and digital signature techniques can be used for this purpose. Confidentiality protection techniques plays important roles in assisting to protect user privacy.

2.5.4 Non-Repudiation

Non-repudiation is a security service that protects a party to a transaction against false denial of the occurrence of that transaction by another party. Non-repudiation technologies provide evidence about the occurrence of transactions that that may be used by a third party to resolve disagreement. The same techniques could be used to protect against reply attacks.

2.5.5 End-to-End Integrity and Confidentiality of Messages

The integrity and confidentiality of messages must be ensured even in the presence of intermediaries and may be even when the data is in storage.

2.6 Summary

Web Service play an important part in the development and integration of advanced electronic solutions. In this chapter we introduced the importance of Web services and briefly described the main building blocks of a Web Service.

One of the core part for the success of any Web Service system is security. In this chapter we reviewed the main security technologies used to secure Web services such as XML Encryption and XML Signature among others. We also looked at WS-Security which the standard and the foundation for securing Web services.

Chapter 3

Mobile Network Security

3.1 Introduction

The growth of the mobile telecommunication system industry in the last 15 years has been extraordinary, and it does not look like that this growth will stop soon. Having said that, the development of the mobile telecommunication system is not without its challenges. Other than the communication engineering issues (e.g. sending and receiving of calls and data) there have been two main challenges to the telecommunication industry: the privacy of the conversation and the billing accuracy [15].

Many mobile phones available today have enough memory and process power to be able to access the Internet. Having said that, mobile phones are restricted with their screen display area and the network link is considerably slower and more expensive than that available to a user of the fixed Internet. As the number of people owning such devices increases an entire industry is developing the concept of mobile commerce or M-Commerce.

3.1 Introduction

Mobile commerce is currently mainly used for the sale of simple 'Value Added Services' (VAS) such as: mobile phone ring-tones and games, although as 3G/UMTS services roll out it is increasingly used to enable payment for location-based services such as maps, as well as video and audio content.

Financial Institutions such as banks see mobile commerce as offering new channels of service to customers as well as offering them new and innovative products. These financial institutions are working to design and implement new applications that will offer mobile payment and mobile brokering. The travel industry, realizing the possible benefits of m-commerce, is working on technologies that will take care of travel arrangements, update customers on flight status, notify them when this information changes and will offer to make new arrangements based on preset user preferences requiring no input from the user.

This chapter reviews in some detail the security mechanisms of popular mobile telecom systems, namely the GSM and the UMTS systems; and highlight some of their security issues. In later chapters we examine the integration of these systems in the Web services environment.

Since many of the mobile phones used are based on the General System for Mobile communications (GSM) [30], users can take advantages of some of the unique security features of the GSM system, in particular the use of the Subscriber Identity Module (SIM). The following sections in this chapter will look into related security features in more details.

The main idea behind the GSM specification was to define several open interfaces, which determine the standardized components of the GSM system [31]. The great difference between 1G and 2G is the presence of a data transfer possibility; basic GSM offers 9.6 kb/s symmetric data connection between the network and

the terminal. However; this was still slow and various modules and specifications were developed to increase the data transfer rate, in which saw the evolution of GSM to The Universal Mobile Telecommunications System (UMTS).

During the migration path from GSM to UMTS, an intermediate phase is General Packet Radio Service (GPRS), also known as 2.5G technology. In GPRS, the existing GSM Radio Access Network (RAN) is augmented with packet-switching capabilities for data services, and a new packet-switched core network (PS-CN) is added in parallel to the legacy circuit-switched core network (CS-CN) to carry data traffic. This GSM/GPRS RAN, the PS-CN are later on connected to the new UMTS RAN (UTRAN) which is based on WCDMA (Wide Code Division Multiple Access).

Because in wireless network data travels in air anyone can capture this data weather it is a voice or other type of data. To protect the security and privacy of any wireless system there is a set of requirements that should be considered when designing such systems [15]:

- Privacy requirements: includes the privacy of the call setup information (e.g. calling number or service requested), privacy of speech, privacy of data, privacy of user identification, privacy of calling patterns and privacy of user location.

- Theft resistance requirement: this covers issues such as clone resistant design, installation and repair fraud, and unique user ID.

- Radio system requirements: because of the nature of mobile system they are subject to; multi-path fading, interference, jamming and various handoff issues.

- Physical requirements: such as equipment installation, cabling, physical access to servers and control system.
- Software requirements: this include general software updates, security patches, and optimization and control software.
- Law enforcement requirements: different countries have different set of telecom enforcement requirement, such that operators must record some calls for sometime, sharing of information, privacy policy, and the cryptographic algorithm used.

3.2 The General System for Mobile communications

The General System for Mobile communications (GSM) dates back to 1982 when the Group Special Mobile was established within the European Conference Post and Telecommunication Administrations, Now GSM has over one billion subscribers world wide and still expanding [32]. In this section we review the GSM architecture focusing on the security features provided by the GSM system, and highlight some of the main security challenges of the system.

3.2.1 Overview of the GSM Architecture

The GSM Architecture consists of subsystems which are required to support the services and functionality of the GSM system. The basic subsystem are: Base Station subsystem (BSS), Network and Switching Subsystem (NSS), and Operational Subsystem (OSS) [15].

The BSS manages and provides transmission paths between the mobile stations (MSs) and the NSS. The NSS has the responsibility of managing communications and connecting MSs. The OSS is a service provider to control the MSs, BSS, and NSS. The following subsections provide more information about the GSM subsystem entities.

Mobile Station

Mobile Station (MS), also known as User Equipment (UE),allows the subscriber to access the GSM network. The MS consists of two parts: the device which contain both the hardware and the software of the MS, and the Subscriber Identity Module (SIM) which contains terminal/user-specific data which act as a smart care. The SIM is issued by the Network Operator after subscription.

Each MS has some unique identities, such as the International Mobile Equipment Identity (IMEI) and the International Mobile Subscriber Identity (IMSI). The IMSI is set inside the SIM.

Network and Switching Subsystem

The Network and Switching Subsystem (NSS) is the main switching that provide a function of the GSM network. It manages the communication between the GSM and other network users. The NSS contains two sets of subscribers databases; the Home Location Register (HLR) and the Visitor Location Register (VLR). Also in the NSS is the Authentication Center (AuC) and the Equipment Identity Register (EIR) which are key to the system security.

3.2.2 GSM security features

One of the main contributors to the success of GSM was a very good set of security features. The main security features which can be found in [5] are:

- Subscriber identity confidentiality.
- Subscriber identity authentication.
- User data confidentiality on physical connections.
- Connectionless user data confidentiality.
- Signalling information element confidentiality.

The GSM security provides three main mechanisms; a cryptographic authentication to verify subscription using challenge/response authentication protocol; radio interface encryption to prevent eavesdropping using symmetric stream cipher; and helps to provide user anonymity by allocation and use of temporary identities.

As it was mentioned the GSM subscriber is uniquely identified by the IMSI, also each subscriber holds authentication key K_i which is kept securely in the SIM. The AuC also has a copy of K_i. The concept of the GSM authentication and encryption schemes is that this sensitive information is never transmitted over the radio channel. The following sections describe each of the security features of the GSM system.

Only the subscriber identity confidentiality and the subscriber identity authentication are described because of their relevance to our research. Good description of the other features can be found in [5].

Subscriber identity (IMSI) confidentiality

The aim of this function is to protect the identity of the subscriber from interceptor of the mobile traffic. Therefore, the IMSI should not be transmitted in clear text. To achieve this, the Temporary Mobile Subscriber Identity (TMSI) was developed to identify the subscriber over the radio path. The TMSI is updated frequently (at every location update).

Subscriber identity (IMSI) authentication

The GSM network authenticates the identity of the user (IMSI or TMSI) using a challenge-response mechanism, which is performed in the following steps, and presented in Figure 3.1:

1. A 128 bit random number (RAND) is sent to the MS.

2. The MS computes the 32-bit signed response (SRES), based on the encryption of the RAND using the authentication algorithm (A3) using the subscriber authentication key K_i.

$$\text{SRES}=A3_{K_i}(\text{RAND})$$

   Where $A3_K(\text{X})$, refers to the output of the algorithm A3 using input key K and input data X. At the same time, the MS computes the encryption key K_c using the A8 algorithm such that:

$$K_c=A8_{K_i}(\text{RAND})$$

3. The MS sends SRES to the GSM network.

4. The network operator repeats the calculation to verify the identity of the subscriber.

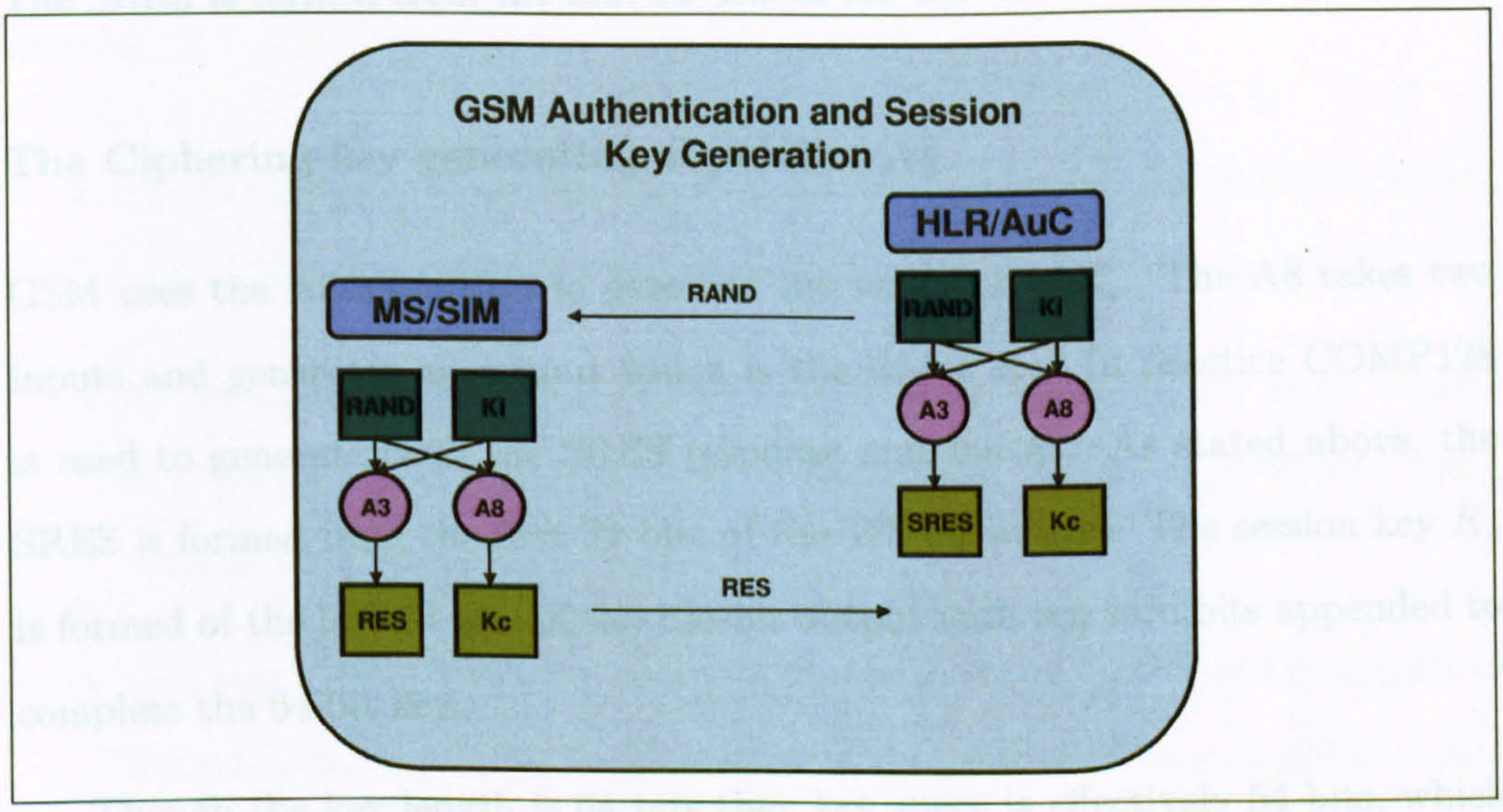

Figure 3.1: GSM Authentication and Session Key Generation

The key K_i is stored in the SIM and in the AuC, therefore K_i will never be transmitted. Instead, all the calculations are processed within the SIM.

3.2.3 GSM security algorithms

This section gives an overview of the three main security algorithms used in the GSM system, namely A3, A8, and A5.

The Authentication algorithm A3

The A3 algorithm can be described as one-way hash function that takes two 128 bits inputs, which are the RAND and the secret key K_i and generates a 32 bits output which is the SRES.

The most used algorithm among GSM operators is known as COMP128. The COMP128 takes the two inputs RAND and K_i, and generate a 128-bit output. The SRES is formed from the first 32 bits of the 128 bits.

The Ciphering key generating algorithm A8

GSM uses the A8 algorithm to generate the session key K_c. The A8 takes two inputs and generates an output which is the 64-bit K_c. In practice COMP128 is used to generate both the SRES response and the K_c. As stated above, the SRES is formed from the first 32 bits of the 128-bit output. The session key K_c is formed of the last 54 bits of the 128-bit output with ten zero bits appended to complete the 64-bit key.

Though the key length is 64 bits they key space is effectively 54 bits, which arguably reduces the strength of the key. Both the A8 and the A3 algorithms are stored in the SIM.

The Ciphering Algorithm A5

The A5 is an encryption algorithm , which works in three modes, the unencrypted mode A5/0, and the A5/1 and A5/2 algorithms to secure the data. Both the A5/1 and A5/2 are considered to be fairly weak. This has led to the development of the A5/3 [8] by the 3rd Generation Partnership Project 3GPP.

3.2.4 Security challenges with GSM

Though security is one of the main strength of GSM, the system does have some security weaknesses. The main security issues are considered in this section.

GSM algorithms security consideration

The following are some of the main issues regarding the Algorithms used in GSM security:

- The GSM cipher algorithms are not published as part of the standard, which lead to the criticism from the research and the academic communities.

- In the COMP-128 algorithm, carefully chosen values for the input RAND will provide enough information to determine the K_i in relatively small number of attempts [33] [34].

- The way COMP-128 has been implemented, it reduces the key length of the ciphering key K_c form 64 bits to 54 bits as the 10 least significant bits are fixed to zeros; this is a reduction of a factor of 1024.

The false base station attack

In the GSM standard only the MS is required to authenticate to the base station (BS), the BS is not required to authenticate itself to the MS. This enable an attack to accomplish the false BS attack by imitating BS. The attacker would page the mobile phone, either using its IMSI or TMSI. If the mobile phone was paged by its TMSI, the IMSI can easily be found out by sending the phone the IDENTITY REQUEST command (to which the phone must respond at any time).

Following this, the attacker can keep choosing RANDs to exploit the COMP128 algorithm flaws and can keep submitting them to the phone via the AUTHENTICATION REQUEST messages (imitating a legitimate network asking the phone to authenticate itself); the phone simply returns the SRES. The attacker could

then repeat the authentication requests many times, collecting the SRESes until he/she has gained enough information to learn the K_i.

Once the K_i and IMSI are known the attacker can impersonate that user, and make and receive calls in their name. It can also be used to eavesdrop, since RANDs from a legitimate network to a legitimate user can be monitored, and thus combined with the known K_i can be used to determine the K_c used for the encryption.

There are few other possible attacks resulting from false base station attack, which are described well in [14].

SIM cloning attack

The GSM SIM card can be cloned [34]; this will lead to two possible scenarios. The first is when attacker uses the SIM card pretending to be the legitimate user. The second is when the attacker exploits the weakness in the COMP-128 algorithm to extract the secrete key K_i.

3.3 The Universal Mobile Telecommunications System

As the need to improve the service of 2G/2.5G mobile systems a lot of effort was put to develop the Third Generation (3G) of mobile phone systems. The main idea behind 3G is to prepare a universal infrastructure able to carry existing and also future services [35]. The infrastructure should be designed so that technology changes and evolution can be adapted to the network without causing

uncertainties in the existing services using the current network structure. Different parts of the world have adopted different technologies, In Europe 3G has become UMTS (Universal Mobile Telecommunication System), following the European Telecommunications Institute (ETSI) perspective. This is now developed by the 3G Partnership Project (3GPP). The 3GPP defined the very first version of the European-style UMTS network by the official name 3GPP System Release 99 [31].

A simplified version of the UMTS network architecture is described in Figure 3.2, the UTRAN is divided into subsystems, each consisting of one radio network controller (RNC) connected to several base transceiver stations (BS). The BSs maintain the air interface in the cells, while the RNC controls the radio connections with the mobile station and the wired interface to the CN. . The PS-CN embeds several elements: SGSN,GGSN, and multiple information servers. The information servers play an important role in the control plane in 2.5G/3G networks. The MS (also known as the mobile device MD)embeds two components that are physically and logically distinct: a software/hardware terminal (e.g.mobile phone) and a subscriber identity module (SIM), which is a tamper-resistant smart card storing a unique identifier and associated secret keys. The UMTS SIM (USIM) is capable of internal processing, and cryptographic algorithms involved in authentication are executed directly on it. The secret keys in the USIM are known to the home AuC (located in the core network), and a trust relationship is in place between the USIM and the AuC.

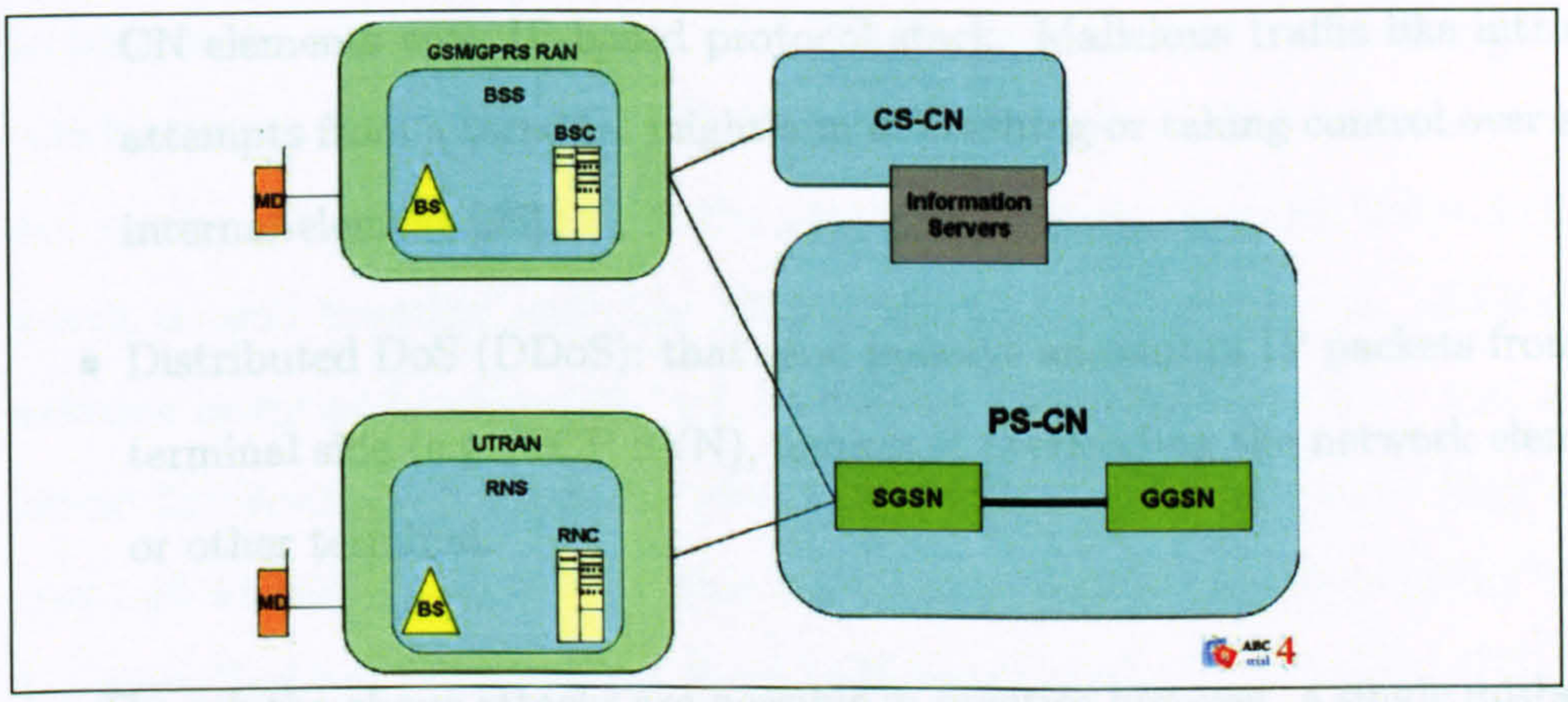

Figure 3.2: UMTS Network Architecture

3.3.1 UMTS Security Threats

As in GSM one of the big security threats is eavesdropping on the radio link, as the USIM is typically associated to a single subscriber, eavesdropping the signaling messages may reveal the subscriber's identity and location, which could compromising the user privacy beyond the content of the data communication.

Because of the complex architecture of the UMTS network various possible attacks against the network are possible:

- Attacks from the Internet: a successful attack on the Internet element of the CN could have big impact on then network. It is possible for a TCP/IP attack to be attempted against the MS and/or some internal IP-based element. This could be damaging because: the radio capacity is limited and is shared which could effect the services for all MSs in the same cell. Also because many UMTS users may be billed per volume, thus unsolicited traffic can cause billing problems.
- Attacks on the GGSN: since the terminals communicate with the internal

CN elements with IP-based protocol stack. Malicious traffic like intrusion attempts from a terminal might aim at crashing or taking control over some internal element [36].

- Distributed DoS (DDoS): that send massive amount of IP packets from the terminal side (e.g. TCP SYN), aiming at overloading the network elements or other terminal.

Though the above attacks are possible in practice however, a single misbehaving MS cannot impose a critical load unless hitting a server loophole or triggering a cascade of failures. This is due to the large resource between the CN and the MSs, and the limited power of the MS [35]. A possible way for an attacker to overcome the issue the limited power of the MS is to group a large pool of MSs and attempt a large scale DDoS. The DDoS can be launched using malicious code (e.g. viruses, worms), an example of such malicious codes is already exist (e.g. Cabir [37] malware, and Mosquito [38] malware). However, its more difficult to infect viruses or worms in 3G networks than in the Internet because of: The MSs makers present high degree of platform heterogeneity that makes it harder for malicious code to spread over a very large set of MSs; users are often charge per volume and will take more care when detecting unsolicited activities.

3.3.2 UMTS security features

UMTS security model has been built upon the GSM security model, The UMTS security features are specified in TS 33.102 [39], which is part of the 3GPP release 99.

The UMTS security system consists of various elements each playing a role

to achieve the required level of security. These elements are: a mobile station MS which contains a removable UMTS-SIM (USIM) which will handle authentication and session key establishment at the user side, a Radio Network Control (RNC) which is used together with the MS to handle encipherment; in addition the network integrity functionality will be located in the RNC, and an AuC in the Home Environment (HE) which stores the long-term cryptographic keys of the user and will produce authentication information which it forwards to the Visitor Location Register (VLR) in the Serving Network (SN).

As UMTS adopted the GSM security model, there was a need to address the security challenges in GSM, such challenges with the cipher algorithm and the 'false base station attack' [33] which we highlighted in section 2.2.4. The following is a summary of these security enhancements [40]:

- New integrity mechanism are added to protect critical signalling information on the radio interface
- Enhanced authentication protocol provides mutual authentication and freshness of cipher/integrity keys
- Stronger algorithm using longer key
- Encryption terminates in the radio network controller rather than the base station
- Some protection of signalling between network nodes

Security in the UMTS network is based on three security principles, which are described in some details in the following subsections. The complete list of

the UMTS security can be found in [39]. We only review the security features relevant to our research.

3.3.3 Authentication and Key Agreement protocol

The Authentication and Key Agreement protocol (AKA) performs authentication and session key distribution [41] in UMTS networks. The AKA is a challenge response mechanism that uses symmetric cryptography. This allows the network to authenticate the user and also allows the user to authenticate the network (this was not available in GSM). AKA is performed when one the following events happen:

- Registration of a user in a Serving Network.
- After a service request.
- Location Update Request.
- Attach Request.
- Detach request.
- Connection re-establishment request.

There are three main parties communicating in the AKA protocol: the AuC, the Visitor Location Register (VLR), and the user Universal Subscriber Identity Module (USIM). There is one secret key K which is shared between the AuC and the USIM. The AKA operate as follows:

1. A shared secret K is established beforehand between the SIM and the AuC.

2. The AuC produces an authentication vector AV based on the shared secret K and a sequence number (SQN). The AV contains RAND, authentication token (AUTN), expected response (XRES), integrity key (IK), and confidentiality key (CK). The AV is then downloaded to a server.

3. The server creates an authentication request, which contains the RAND and AUTN; the authentication request is then delivered to the client.

4. The client verifies the AUTN with the SIM using its own shared secret K and the SQN. If successful the client produces an authentication response RES, using the shared secret K and RAND, RES is then delivered to the server.

5. The server compares the client authentication response RES with the expected response XRES. If they match, the user has been successfully authenticated, and the session keys IK and CK can be used for protecting further communication between the client and the server.

In total five one-way functions are used to compute the authentication vector. These functions are denoted f1, f 2, f 3, f4 and f5. The first differs from the other four in the number of input parameters, taking four input parameters: master key K, random number RAND, sequence number (SQN) and, finally, an administrative Authentication Management Field (AMF). The other functions (f2 to f 5) only take K and RAND as inputs. The ETSI have developed a common cryptographic core engine to compute f1-f5 [42], the result was the MILENAGE framework [43] which is block cipher with 128-bit blocks under control of a 128-bit key. The MILENAGE framework is based on the Rijdael block cipher because it has good performance on platforms with limited computing resources [42].

The Figure 3.3 outlines the main process of the AKA protocol. Note that the VLR was omitted from the diagram for simplicity.

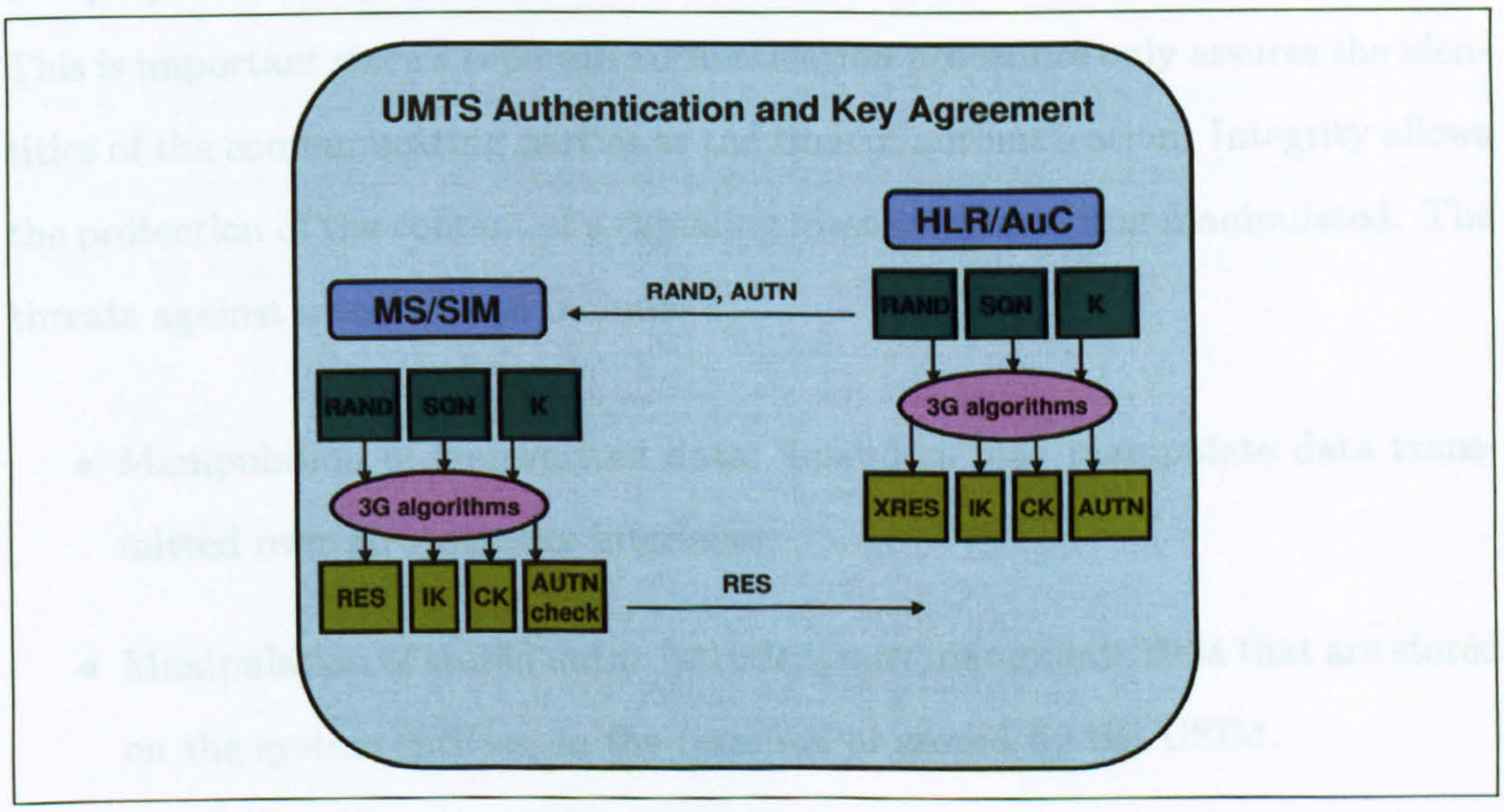

Figure 3.3: Over view of the UMTS AKA protocol

The AKA protocol is a strong protocol and it has been tested [41], however there are few issues with the AKA protocol which include the following:

- Synchronization Failure: this is caused if the client detect that the sequence numbers between the client and the server have fallen out of sync. In this case, the client needs to produce a synchronization parameter AUTN, using the shared secret K and the the client sequence number SQN [39].

- Securing the SIM: the AKA protocol assumes that the SIM application resides in a tamper resistant smart card; as no interfaces is allowed to access the long term secret.

3.3.4 Integrity In UMTS

The purpose of integrity protection is to authenticate individual control messages. This is important since a separate authentication procedure only assures the identities of the communicating parties at the time of authentication. Integrity allows the protection of the content of a signaling message from being manipulated. The threats against integrity can include:

- Manipulation of transmitted data: Intruders may manipulate data transmitted over all reachable interfaces.
- Manipulation of stored data: Intruders may manipulate data that are stored on the system entities, in the terminal or stored by the USIM.
- Manipulation by masquerading: Intruders may masquerade as a communication participants and thereby manipulate data on any interface.

For various reasons UMTS networks must be able to instruct the MS to use an unciphered connection. To protect against man-in-the-middle attack the user must be able to verify instruction from the network to establish an unciphered connection [33], and this is why integrity should be made mandatory in UMTS.

The integrity protection in UMTS is implemented between the Radio network controller (RNC) and the MS. Therefore, IK must be distributed from the AuC to the RNC; The IK is part of an authentication vector which is sent to the VLR from the AuC following an authentication data request. To facilitate subsequent authentications, up to 5 authentication vectors are sent for each request. The IK is sent from the VLR to the RNC as part of a message called security mode command.

The integrity protection mechanism is based on the concept of a MAC. This is a oneway function that is controlled by the secret key IK. The function is denoted by f9 [44] and its output is MAC-I: a 32-bit random-looking bit string. The MAC-I is appended to each RRC message and is also generated and checked at the receiving end. Any change in input parameters are known to influence the MAC-I in unpredictable ways [31, 45]. This algorithm takes five inputs:

- The 128 bits integrity key IK.
- A 32 bits integrity sequence number (COUNT-1).
- A 32 bits random value generated by the radio network controller (FRESH).
- A direction identifier (DIRECTION).
- The radio resource control (RRC) signalling message content (MESSAGE).

The output is a 32-bit message authentication code (MAC-I) computed by the sender for data integrity. The MAC-I will then be appended to the RRC message when sent over the radio access link. The receiver will verify the message by computing the expected MAC-I (XMAC-I) on the message received.

3.3.5 Confidentiality in UMTS

The UMTS encryption mechanism is based on a stream cipher concept between the MS and the RNC. This is an improvement from the GSM system which only encrypted data between the MS and the BS. Confidentiality is very important in UMTS as it protect from various threats such as: eavesdropping on user traffic, signalling or control data on the radio interface; passive traffic analysis, in

which intruders may observe the time, rate, length, sources or destinations of messages on the radio interface to obtain access to information; or confidentiality of authentication data in the USIM.

The core of the encryption mechanism is the mask generation algorithm that is denoted as function f8 [44,45], which is used to encrypt plain text. This block cipher transforms 64-bit input to 64-bit output. Transformation is controlled by the 128-bit CK which are generated by the AKA procedure [31], The f8 takes five inputs:

- The 128 bits cipher key CK.
- A 32 bits time dependent input COUNT-C.
- The bearer identity BEARER.
- The direction of transmission DIRECTION.
- The length of the required key stream LENGTH.

The output will be the key stream block KEYSTREAM, which is used to encrypt the input plaintext block PLAINTEXT to produce the output ciphertext block CIPHERTEXT. A ciphering indicator is also present in the MS to indicate to the user whether encryption is applied or not. Note that although the use of ciphering is highly recommended it is still optional for the UMTS network.

The 3GPP has defined an algorithm called KASUMI [7], which can be used in two different modes, one is the confidentiality mode to construct f8 and the other is the integrity mode to construct f9.

3.4 Summary

The growth in M-Commerce is picking up pace very fast; however one of the main issues is security. Therefore it make sense to try to see if current security features provided by the current mobile network infrastructure can be used in improving/developing security solutions for M-Commerce applications. In this chapter we reviewed the GSM and UMTS security models, we looked at the operational mechanism of the various security algorithm used and described possible security vulnerabilities in both systems.

Chapter 4

Federated System Authentication in Mobile Environment

4.1 Introduction

A common way nowadays when booking a holiday is to go to a favorite airline's web site, log in with username and password, enter authentication information and book the reservation. Then you remember you are going to need a car, so you search for a car rental site, log in again with a different user name and password, and reserve your car. Then a similar process at the hotel's Web site, log in with yet another user name and password, and book your room.

Because of the above there some issues when using Username/Password for authentication, such as: using easy to guess passwords (e.g people names), or using short passwords (less than 8 characters) which makes it easy target to dictionary attacks, or worse still to writing the password down and having them next to their computers, and also using the same password for many different

applications [46].

Wouldn't it be easier if you could login to one site and then be authenticated at associated sites? Single Sign On (SSO) tries to solve some of the issues regarding username/password authentications described above.

Mobile commerce or M-Commerce is growing fast due to the increased functionality available in modern mobile devices. Service providers can now provide a wide range of services such as as Vodafone Live! which allow users to download digital content or allowing users to buy goods using their mobile devices such . However, one of the key issues for M-Commerce applications is the protection of user privacy, such as personal details, financial data or even user behaviour (e.g. shopping habits). Theoretically it is possible to have systems with full anonymity [47], although such systems may not be desirable as it could lead to increased misuse of the system.

Single Sign-On systems are suitable to Web services environment, and can be a secure mechanism to authenticate users to various Web services. However, authenticating the users to the Identity Provider (IdP) or Authentication Server (AS) is challenging and usually it is the weakest point in the system as it depends on human interaction such as remembering username/password. On the other hand, GSM/UMTS network deploys strong "smart card" authentication mechanism as described in chapter three, and since many people these days have mobile phones (using GSM/UMTS technology) it make sense to try to integrate the two systems to improve security and provide better user experience. In the next section we propose a protocol that combines SSO systems with some of the GSM security features to secure M-Commerce system.

The ultimate goal of identity federation is to enable users of one domain to

securely access data or systems of another domain seamlessly, and without the need for completely redundant user administration. Identity federation comes in many flavors, including user-controlled or user-centric scenarios, as well as enterprise controlled or B2B scenarios.

In this chapter we investigate and propose Web service based protocol solutions to address such issues.

4.2 Authentication and The Concept of Identity

The concept of identity has had a long history in social science analysis. When discussing identity, there remains a considerable lack of clarity as to the focus of debate [48]. The concept of Identity is not easy to defined, as to how an individual (or an entity) is represented in a system and how that individual can prove that the information in the system refers back to him has a profound impact on system use [49, 50].

In our research the focus is more on a particular case where a user would like to access different service providers with the same or different identities. Our focus is to investigate how much information should be linked to the user identifier(s) to allow the service providers to deliver the required service without compromising the user security or privacy. In many cases there is a need for a trusted third party to be involved in the process to assist with the identity management and to balance the associated security/privacy risk .

The service provider may or may not know the identity of the user to whom they are providing service the [51]; depending on the service this may not be important, for example the service provider providing general news. However, the

identity of the user could be of some importance when some level of authentication is required, but what if the service provider did not directly authenticated the user? The answer to this question is that it depends on the service provided. To explain further, let us consider the following three real life scenarios:

The Cafe Scenario:

A man goes to his local cafe and asks for a coffee with milk and no sugar. The coffee-maker makes the coffee and the user pays and gets the coffee. The next day again the same man comes to the cafe and asks for a coffee with milk and no sugar, he keeps doing this for few days. The next time he came to the cafe, the coffee-maker provides the 'customized coffee' without asking! The point is that it is possible in some services/application to provide customized service without knowing the true identity of the user.

The Car rental Scenario:

To rent a car you need to provide some kind of identification, such as driving licence. The car rental company are not interested in the full details of the identity of the person who is hiring the car; they are only interested to know that this person can legally hire the car and is able to pay. This information can be provided by a trusted third party such as the DVLA in the UK. In this case a service can be provided without giving away the full details of the user identity through the aid of a trusted third party.

The Bank Scenario:

If someone is trying to withdraw a large sum of money from their personal account from a bank, very likely they will be asked to provide more than one form of

identifications and to answer some security questions before releasing the money. Because of the sensitivity of the service provided the user need to provide their for identity to the service providers.

It is possible, regardless of whether an account exists for the user at the service provider, for an identification to be established between the service provider and the identity provider based upon the direct authentication of the user by the identity provider and the user's account with that identity provider [51].

A user identifier can be a unique string of digits, or a user name, or commonly on the Internet an email address. From a security point view it is important to choose the right user identifier within the context in which it will be used. To use the same user identifier within one domain may not be an issue; however, to use the same user identifier across multiple domains could compromise user privacy. This particular feature is addressed in the Liberty Alliance framework through the creation of "opaque privacy protect name identifier" [52–54]

4.3 Federated System with Single Sign-On

SSO is a mechanism whereby a single action of user authentication can permit a user to access many services providers (SPs). This is typically done through an "Authentication Server" (AS), also known as Identity Provider (IdP) [55]. There are different SSO systems, however they generally require a User to authenticate itself to an AS/IdP to obtain access to the SPs. The process of authentication varies from system to system; overall there are two main SSO schemes, the first is where a strong trust relationship exist between the user and the IdP, the second scheme is where the trust relationship exists between the IdP and the SPs.

4.3.1 SSO based on User-IdP trust relation

In this scheme the IdP manages the user-SP authentication credentials, in such a way that a separate user authentication occurs every time the user is logged into an SP as shown in Figure 4.1. For this, the IdP needs to have a database for each user with a list of all the SPs the user has a relation with and their authentication credentials. For this reason, a trusted relation between the user and the IdP is a must. Therefore, the registration process must be managed very carefully to ensure secure authentication at later stages. From a business perspective this is a very desirable model as it will allow the user to access all of his/her accounts (i.e. enabling personalized marketing and promotion) with a single authentication process.

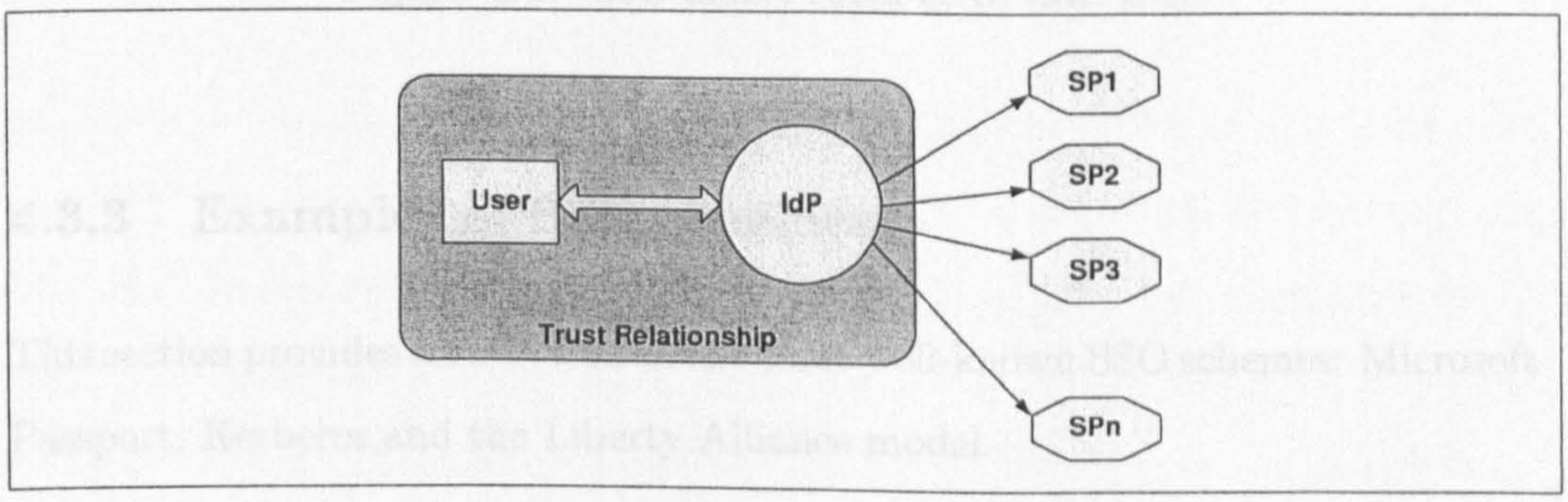

Figure 4.1: SSO model based on User-IdP trust

4.3.2 SSO based on IdP-SPs trust relation

This scheme requires the IdP to have an established relationship with all SPs that are part of the SSO system as shown in Figure 4.2. This relationship requires a level of trust that is typically supported by a contractual arrangement. The main feature of this scheme that makes it different from the previous scheme is that the

only authentication process that involves the user occurs between the user and the IdP; SPs are notified of the authentication status of the user via authentication assertions. These are statements that contain the user's SSO identity and his/her authentication status at the IdP.

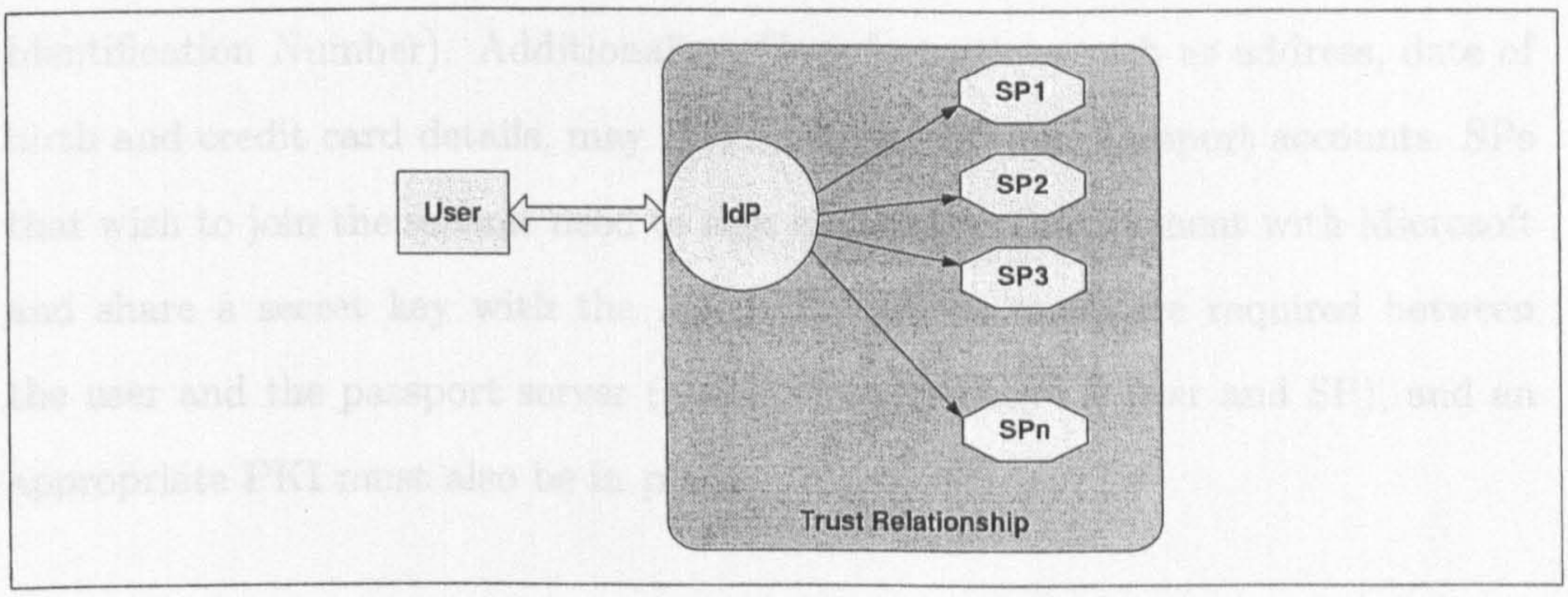

Figure 4.2: SSO model based on SP-IdP trust

4.3.3 Examples of SSO schemes

This section provides an overview of the three well-known SSO schemes: Microsoft Passport, Kerberos,and the Liberty Alliance model.

Microsoft Passport

Microsoft Passport [56] is a web-based SSO service that has been offered by Microsoft since 1999. The Microsoft Passport provides authentication services for Passport-enabled sites called 'participating sites'. A centralized Passport server is the only IdP in the Passport model and contains users' authentication credentials and the associated unique global identifier called Passport Unique Identifier (PUID). Cookies play a major role in the Passport architecture [56] where the

Passport server stores and reads identity information in the form of session and browser cookies stored securely at a client side.

Users register with the IdP by supplying a valid e-mail address and a password (or, if they register from a mobile phone, their phone number and a Personal Identification Number). Additional profile information, such as address, date of birth and credit card details, may also be stored in their passport accounts. SPs that wish to join the scheme need to sign a contractual agreement with Microsoft and share a secret key with the AS. SSL/TLS channels are required between the user and the passport server (and optionally between user and SP), and an appropriate PKI must also be in place.

Kerberos

Kerberos [57] is an authentication system. It provides evidence of a principals identity. A principal is generally either a user or a particular service on some machine. Kerberos is a single security domain, or realm, consisting of a set of users, an Authentication Server, a 'Ticket Granting Server' and a set of relying SPs [58]. The Authentication Server and the Ticket Granting Server can be combined into a single entity called the 'Kerberos server'. The security infrastructure of Kerberos relies solely on symmetric cryptography; every user and every SP share a long-term secret key with the AS. All secret keys are used to perform an encryption operation. A detailed description of Kerberos can be found in [57].

Kerberos is suitable for supporting authentication, authorization, and confidentiality within a network or small set of networks [59]. Encryption in the implementation of Kerberos mainly uses the data encryption standard (DES). It is a property of DES that if a ciphertext is decrypted with the same key used

to encrypt it, the plaintext appears. If different encryption keys are used for encryption and decryption, or if the ciphertext is modified, the result will be unintelligible, and the checksum in the Kerberos message will not match the data. This combination of encryption and the checksum provides integrity and confidentiality for encrypted Kerberos messages.

Some new versions of Kerberos start using PKI, which could make it easier to deploy in Internet applications. Though Kerberos has been around for some time it has some known limitations which were discussed in [59, 60].

The Liberty Alliance

The Liberty Alliance [52] is a consortium of over 140 companies who recently developed a set of open specifications for web-based SSO [52, 53, 61]. The specifications use the Security Assertions Markup Language (SAML), a platform-independent framework for exchanging authentication and authorization information. Liberty is based on the notion of 'Circles of Trust ' which are formed by trusted ASs and sets of relying SPs. The AS/SP trust relationship has to be supported by contractual agreements outside the scope of the specifications.

According to the specifications, users first authenticate themselves to the AS, which subsequently conveys authentication assertions to the relying SPs. The assertions contain 'name identifiers' that allow SPs to differentiate between users. For any given user, the AS has to use a distinct identifier with each SP in the trust circle.

The SSO name identifiers must be constructed using pseudo-random values

that have no discernible correspondence with the Principal's identifier (e.g. username) at the Identity Provider (also known as AS); SSO identities are therefore potentially unlinkable. This unlinkability, however, can be compromised in a number of ways. Firstly, as the AS knows all the user identifiers, SPs could collude with the AS to link the pseudonyms of a user. Secondly, SPs may be able to correlate SSO identities based on the user network addresses.

Thirdly, profile information that individual SPs may maintain (such as shopping habits, telephone numbers or credit card details) can also be used to link identifiers. Although this last point lies outside the scope of a user authentication scheme, the specifications acknowledge that, for the time being, the only protection is for the users to be cautious when they choose service providers and understand their privacy policies.

The Liberty Protocols and Schema Specification define generic requirements for the protocols for conveying assertion requests and responses between parties. Concrete protocol bindings are only specified in the context of a Liberty profile. All currently specified profiles rely on the Secure Socket Layer (SSL) or the Transport Layer Security (TLS) [62] protocol to provide secure channels between parties. Hence, a Public Key Infrastructure (PKI) must be in place.

The Liberty system provides flexibility and scalability without the need for one single IdP/AS and for that we think it is very suitable to be deployed in the Web services environment.

4.4 Related Work

A substantial research effort exists on developing solutions for authentication and payment solutions for mobile commerce systems. These solutions range from a simple WAP connection between a mobile device and a server using user-name/password [63,64] to adding a second dedicated chip for authentication and payment to the mobile device as introduced by Japan NTT DoCoMo [65], and pure Java security application [66].

One of the key areas of research related to SSO is the Cross-domain identity management such as the Liberty Alliances specifications and Microsoft Passport. In [67,68], the author looks at the users privacy issues in the Liberty Alliance, and highlight some ambiguities and propose privacy policy together with a few changes to the Liberty processing rules.

The SSO is the key platform for many distributed systems. It is typically accomplished by the creation of cryptographic token based on initial manual authentication by the user (e.g. entering a password). This token can be utilized to authenticate subsequent actions. Creating instances of such tokens allows for delegation to services on remote systems. In [69] suggest that it may be difficult to modify a particular legacy application to utilize an authentication scheme other than username/password. While the existing protocols such as Kerberos [58] and X.509 [70] provide such capability, the author highlights that in order to use a protocol of this type, all processes involved must understand and utilize the technologys protocol. This could be an issue in the Web services environment. A reference to that was provided by the author in [69]. An interesting approach to solve the problem was proposed in [69], where a "session password" is introduced.

These session password are short-lived passwords transmitted in place of a user's private password.

One of the things holding back the widespread use of cross-domain Services Oriented Architecture is the delay in reaching consensus on how to secure the service between different organizations [71, 72]. Solutions to address these issues can be achieved using federated identity management framework (which is widely adopted in Web services [51]). The work in [73] suggests that important limitation exists with federated identity management systems, especially in associating access policies with a given identity when crossing organizational boundaries. The authors suggest to use federated access management instead, where the request conveys the authorization information instead of or in addition to the requesters identity. Though proposal in [73] is interesting and is built on top of the current standards [21, 74], the proposal is using these standards in a way not referred to by the standards committees and further work is needed to insure that these standards were not misused.

The work in [75] addresses the issue of scalability in WS-* specification. The work looks at the authentication issue of messages exchanges in large scale decentralized systems, composed by different authentication domains. The authors argue that there are scalability and flexibility limitations due to the fact that the acquirement of the identity claims requires online interactions with security token services, which introduces communication overhead and creates performance bottlenecks. The services policies, containing its requirements, must directly point to the issuing security token services, limiting the flexibility of the trust relations.

One of the earlier research on anonymous credentials were presented in [76],

where the author considers a situations where a user wants to interact in unlinkable roles with different organizations and nevertheless transfer certified attributes between these organizations.

While there is an increasing number of published research work in the field [47,77–82], there is still a distinct absence of published solutions for this problem, with majority of related solutions covering Web Services SSO. In [77] the model of an authentication and authorization infrastructure supporting single sign-on and federations of trust on the basis of Web Services has been demonstrated. In order to reach this aim, the work focuses on the design of an authentication protocol closely related to the Kerberos protocol. Although this work gives a general authentication framework for Web services it does not take into account the mobile environment. The mobile environment is addressed, among others, in [78, 79]. Both of these works provide GSM for SSO solutions and show similarities to the concept presented in this Thesis. These solutions, however, require direct communication between the User and the service provider, and also require sending of the International Mobile Subscriber Identity (IMSI) unencrypted, which potentially compromises the privacy of the User. In these two papers the authors assume the ability to gain access to the session key which is the property of the mobile operators, and its unlikely that the operators will give a direct access to its security properties to third party service providers. The work proposed in [47] addresses the issue of user anonymity. However, the proposed work makes intensive use of certificates at the client side (i.e. user), which can be an issue in mobile environment, as discussed in more detail in this paper. In addition to these proposals, [81] analyses IP-based services security architecture for wireless and public domain networks, especially where Web services are brought into play,

and [82] proposes an architecture and conceptual security model for enterprise level mobile networks.

4.5 Verification, Validation and Formal Methods

In all engineering designs it is desirable to be able to verify and validate the design before going into production stage, the aim is to save time and production cost. Another important goal of any verification/validation process is to try to find any error, inaccuracy, or system limitation in the system in order to fix them at the design stage.

Verification and validation are the process to check that the system/product meets its specification. Validation is concerning with building the right solution; were validation is concerning building the product/solution right. Within the context of software development, [83] suggest that the ultimate goal of the verification and validations process is to "establish confidence that the software is fit for purpose", the level of confidence vary, depending on the system's purpose, such as software function or user expectation. There are two complementary approaches to system checking and analysis:

- Static verification and validation: where no need to run the software on a computer, this is done by software inspection or peer reviews to check system representation such as the requirement documents, design diagrams, and the program source code.

- Dynamic verification and validation: this involves software testing to run

the implementation of the software with test data to examine the output and operational behavior.

4.5.1 Formal Methods

Formal methods is the mathematical representation of the software, they are concerned with the mathematical analysis of the software specification. The aim is to formally verify that one representation of the system is semantically equivalent to another representation. Mathematical arguments are used to formally verify that code of a software system is consistent with its specification.

The use of formal verification may reveal potential inconsistencies or system flows that may not be easily discovered without running the software. However; as the system size increase, the cost of formal verification will increase greatly. Because of that many analysis think that formal verification is not cost effective [83], and that similar level of guarantees can be achieved by inspection and system testing. In addition, formal verification does not guarantee that the software will be reliable in practical use. This is very true in large distributed systems, such as Web services, where formal specification may not reflect the real requirement of system users. This is also true when the system consists of many interconnected application and services, where a change in one particular application may lead to fundamental change in the overall system behavior.

As described in chapter one, many of the security protocols nowadays depends on cryptography. These cryptographic protocols were designed to secure against various attacks on the network. Over the years many security protocols were developed and many were found vulnerable to attacks that do not require to compromise, they underling cryptographic algorithms, but rather manipulate

the messages in the protocols to compromise confidentiality, integrity, or to gain some other advantages. This has lead to the development of formal description methods to understand these security protocols. One of the earlier proposals for formal method for describing and analyzing authentication protocols was introduced in the late 1980's by Burrows, Abadi, and Needham on what is known as the BAN logic. The aim of the logic of authentication is to formally describe the knowledge and the beliefs of the entities involved in authentication while analyzing the protocols step by step [11].

The BAN logic proposes a formal method to describe authentication protocols, by listing the source, the destination, and the contents symbolically, is replaced with logical formulas. The aim is describe all the information in a particular step in the protocol with this logical formula, this process is referred to as the 'idealization' of the protocol. The beliefs of the entities or the principals are annotated with assertions on the idealized protocol. The protocols is then analyzed step by step using a set of inference rules [11].

Another proposal described by C. Meadows [84] describes another common approaches to this is by using "state machines". The protocol is translated into another executable formalism, which allows its simulation in presence of an aggressive intruder. The intruder is allowed to "randomly" perform actions such as intercepting communications and forging messages. By an exhaustive search one can establish if the protocol is flawed or not (its important to note that this will depend on what "exhaustive search" will include, as it will depend on the experience/knowledge of the system designer). Another interesting proposal is the NRL protocol analyzer [85], the NRL protocol Analyzer is a prototype special purpose verification tool, written in Prolog, which has been developed for the

analysis of cryptographic protocols that are used to authenticate principals and services and distribute keys in a network. Many solutions have been employed, ranging from human intervention to the use of approximations.

Longley and Rigby [86] proposes the use of a rule based system that transforms goals into sub goals and can constantly continue this process. The rule based scheme is used to build a tree, in which each node represents a data item, and the children of a node represent those data items that are required for the knowledge of the data represented by the father node. In this way, they can construct a tree, in which the root node represents the data required by the attacker for an attack (e.g., a cryptographic key), and the leaves represent those data items that are required to know the root item. This tool allows the user to interact with the system. The user can determine whether a data can or cannot be found by the attacker. If the data is judged to be accessible, this information can be inserted into the system, and the generation of the tree can proceed.

4.5.2 Security Assessment

Security is becoming critical factor in most of current software system as these systems are in one way or another are connected to the Internet, such as for user access or system updates. This becomes more essential in Web services were system interconnection is the norm. The ability to assess/analyze that a system can resist against different types of attacks is not easy [87]. It is difficult to assess system security as most of security requirement are "shall not" requirement [83], as they specify what should not happen instead of required system functionality. That makes it hard to define or check these unwanted behaviors.

Even if a system has been deployed for years, it is impossible to prove that

a system does not do something, irrespective of the amount of testing [83], security vulnerabilities may still exist in the system. However, security analysis and assessment is a must to check system security. There are four approaches to security assessment:

- Experienced based verification/validation: here the system is checked against known attacks; usually security checklist is prepared to assist with this process.

- Tool based validation: this is valid with some applications or systems, when a common functionality is deployed in a system. Examples of such tools are password checkers that are used to check for weak password.

- Tiger teams: this is where a team is set to attack the system with the aim to discover its weakness or to find new ways to compromise the system. This is a very effective approach if the team experienced in system hacking.

- Formal verification: a system can be verified against a formal security specification. However, security formal verification is not commonly used. As it is hard for an end user of a system to verify its security. Methods are developed for industrial security validation for software products [88] which gives an industrial security certification. However, such certification is only applicable to individual product or system. When the product or system is used in conjunction with other uncertified systems (e.g. in Web services) the overall system security can not be assessed.

4.5.3 Concluding remarks on formal methods verification

There has been great amount of work on formal methods in computer security, including the logics of authentication which has seen many improvements. However, great challenges still exist in particular the formalization of a general authentication protocol. An important weakness of designing security protocols lies in the formalization process [89], because of ambiguities and of the difficulties in formalizing, its not easy to define what the actual goals of a protocol are. This simplification which usually takes place when formally describing authentication protocol does take the logic away from the real world.

In the following sections of this thesis we propose some novel security protocols which address different mobile Web services challenges. Experience based validation/verification approach together with the protocol design principles presented in chapter one was the chosen methodology for verifying our proposed protocols. The experienced based approach was chosen over formal methods as we believe it's more applicable, due to the limitation of formal methods in large distributed environment such as mobile Web services, as discussed in this section.

4.6 Security in Federated Systems

SSO systems simplify user interaction with service providers. However, security will always be one of the key issues that any SSO system must address to gain user confidence. Several key security issues have to be identified and these are listed in this section. Some of these issues are very specific to certain types of SSO systems, while others are more general due to the nature of the SSO model. The main issues include:

- **The centralization issue**: Most SSO models depend on a central point that all parties trust (i.e. IdP or the AS). From a security point of view this is a potential single point of failure; if the security of the IdP can be broken, the security of the whole system will be effected.

- **UserName/Password dependence**: Most SSO systems will require the user to have a username/password to access the system (e.g. such as in Kerberos and MS Passport). This is not effective against password guessing attacks; if a user chooses a poor password, then an attacker guessing that password can impersonate the user.

- **The registration Process**: In many SSO systems less consideration is given to the security of the registration process; this is of great importance especially in SSO models where the IdP stores users-SPs access information, and where the SPs need to give a customized services based on these access information.

- **Reply attack**:This happens when an attacker tries to resend the messages; to gain access for example.

- **Timing attack**: The aim of this attack is to reveal the user identity or to understand user behaviour by monitoring the network traffic. This attack can be serious if not many users using the system or the IdP doesn't change the "User Identity" regularly. A solution to this is to increase the time intervals between receiving and sending another message at the User and/or IdP side.

- **Tradeoffs between usability and security**: In particular we refer to the work load on the user; such as how often should the session key be changed?

this problem becomes more serious in mobile Web services applications because of the computing limitation of mobile devices.

In the following subsections we propose a methodology to address some of the above issues in particular we look at the registration challenge. We first define the environment in which we apply our proposed solutions.

4.6.1 Environment Architecture

The concept of the system is to allow a user to securely access all his/her online services, with one login, as well as the option to register with new services. The System consists of the following three entities:

- User: the end user who wants access to a service provider.
- Service Provider: an entity selling goods or services to the User.
- Identity Provider: a third party providing SSO functionality to the system.

The system follows a strict protocol in which a series of SOAP messages are sent between the entities of the system to facilitate for the registration and authentication process of the system; in addition the following requirements must be met prior to the use of the protocol.

- All parties have agreed on a specific signature algorithm. The signature of data X using secret key K is written $s_K(X)$.
- All parties have agreed on an asymmetric encryption algorithm, for which the encryption of data X using public key P is written $e_P(X)$.

- All parties have encryption key pairs for encryption scheme , and all the parties possess a trusted copy of the public key of the other two parties.

- All parties have asymmetric key pair for a signature scheme , and all parties possess a trusted copy of the public key of the other two parties.

The following two sections will demonstrate our proposed protocol in both the registration and the authentication stages.

4.7 Registration Process and Mechanism

Registration is a crucial stage in any access control system, as any security breach at this stage could have serious consequences on the system [46] (e.g. if someone manages to fake or steal someone else's identity).

We are here considering the special case where a User wants to register with an IdP, that he/she has an account with. So that when the User is authenticated to the IdP he/she can access their private account at the SP.

At the initial stage of the registration process, the user register with the IdP; for that the user sends his/her authentication credentials (e.g. user name and passwords) to the IdP. These credentials will be used for the future login to the IdP. The message that contains the credentials is encrypted with the IdP public key (PK_{IdP}) for confidentiality and digitally signed by the user private key (SK_{User}) to protect the integrity of the message.

Message 1 User $\rightarrow$ IdP: $s_{SK_{User}}$ ($e_{PK_{IdP}}$(Authentication Credentials))

The IdP needs to verify the message, to add the user credentials to its database and assign a unique "User Identifier" to the user.

The service provider (SP) also needs to register with the IdP to join the SSO system. The SPs follow the same registration procedure as the user, there also may be some kind of contractual agreement. Once the SP is registered with the IdP, the SP will also be assigned a unique "SP Identifier".

For a user to access any of the SPs through the IdP, he/she needs to register his/her 'account' with the SP at the IdP. The concept behind this is to be able to link the user Identity at the IdP with the user Identity at the SP. The registration process commences with the user login to the IdP using the user authentication credentials. The user then requests to register with an SP (this SP must be registered with the same IdP). It is not necessary for the user to be externally registered with the SP system. Upon receiving the user request, the IdP sends a "Registration Token" to both the user and the SP. Registration Token is encrypted with the SP public key (PK_{SP}) and signed by the IdP.

Message 2 IdP → User: $s_{SK_{IdP}}$ ($e_{PK_{SP}}$(Registration Token))

Message 3 User → SP: $s_{SK_{User}}$ ($e_{PK_{SP}}$(Registration Token))

Following that, the user logs in to the SP if he/she has an account with the SP, otherwise the user can create an account with this SP; then the user must show that he/she possesses the Registration Token obtained from the IdP. If the login and the validation of the Registration Token are successful, the SP authorizes the particular registration token to the IdP by sending it with a "user Identifier" (e.g. user name) at the SP system. The SP will sign the message

before sending it to the IdP to ensure the integrity of the message. Note that to improve user privacy the SP should not pass the true user Identifiers to the IdP, rather it should produce a reference to it and send it to the IdP. An overview of this process is illustrated in Figure 4.3

Message 4 SP → IdP: $s_{SK_{SP}}$ ((registration token),user Identifier at the SP)

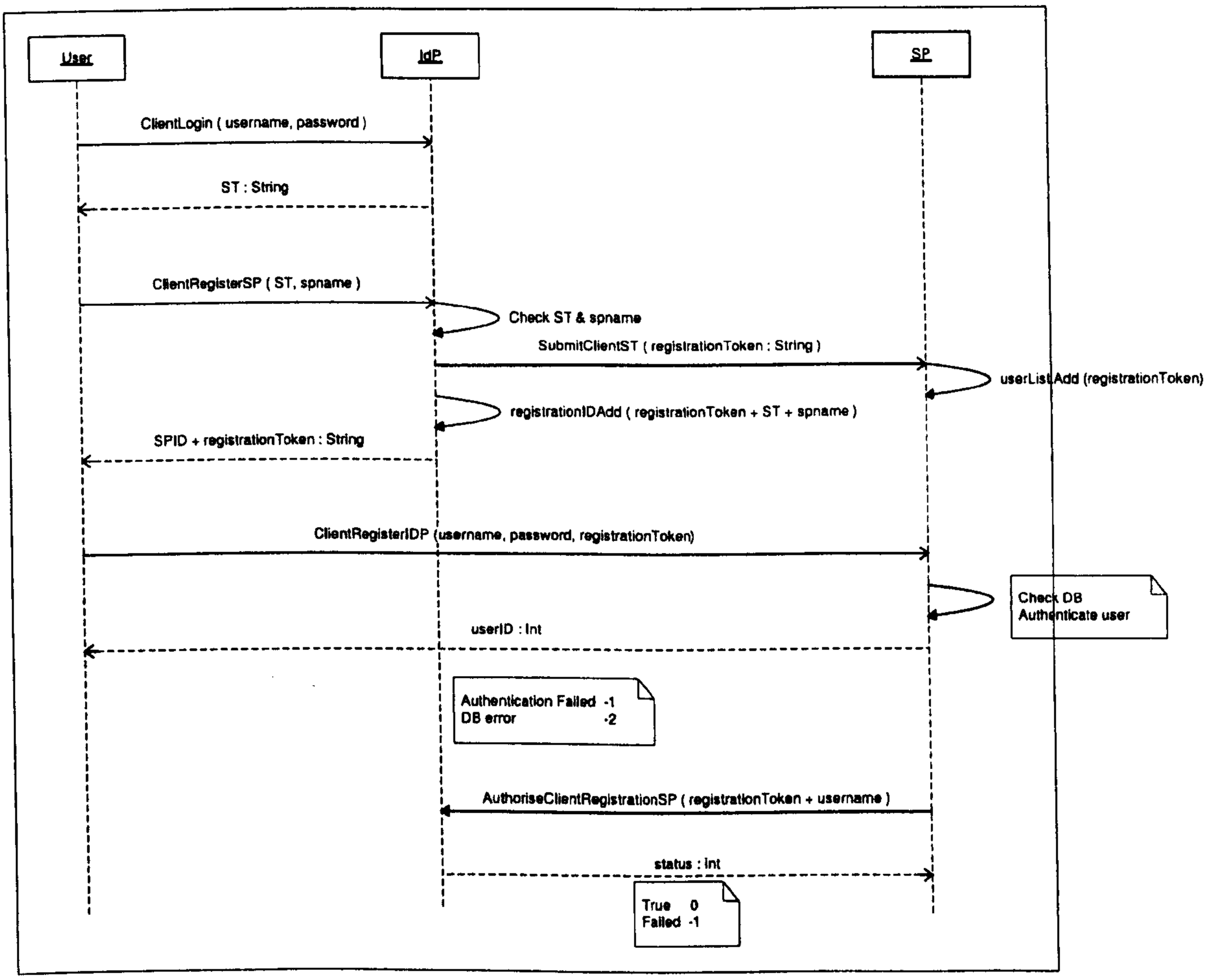

Figure 4.3: User's SP registration with the IdP

Once the authorization reaches the IdP it saves the user's "user Identifier" with SP in a database and this completes the registration process, Figure 4.4 gives an overview of the registration process.

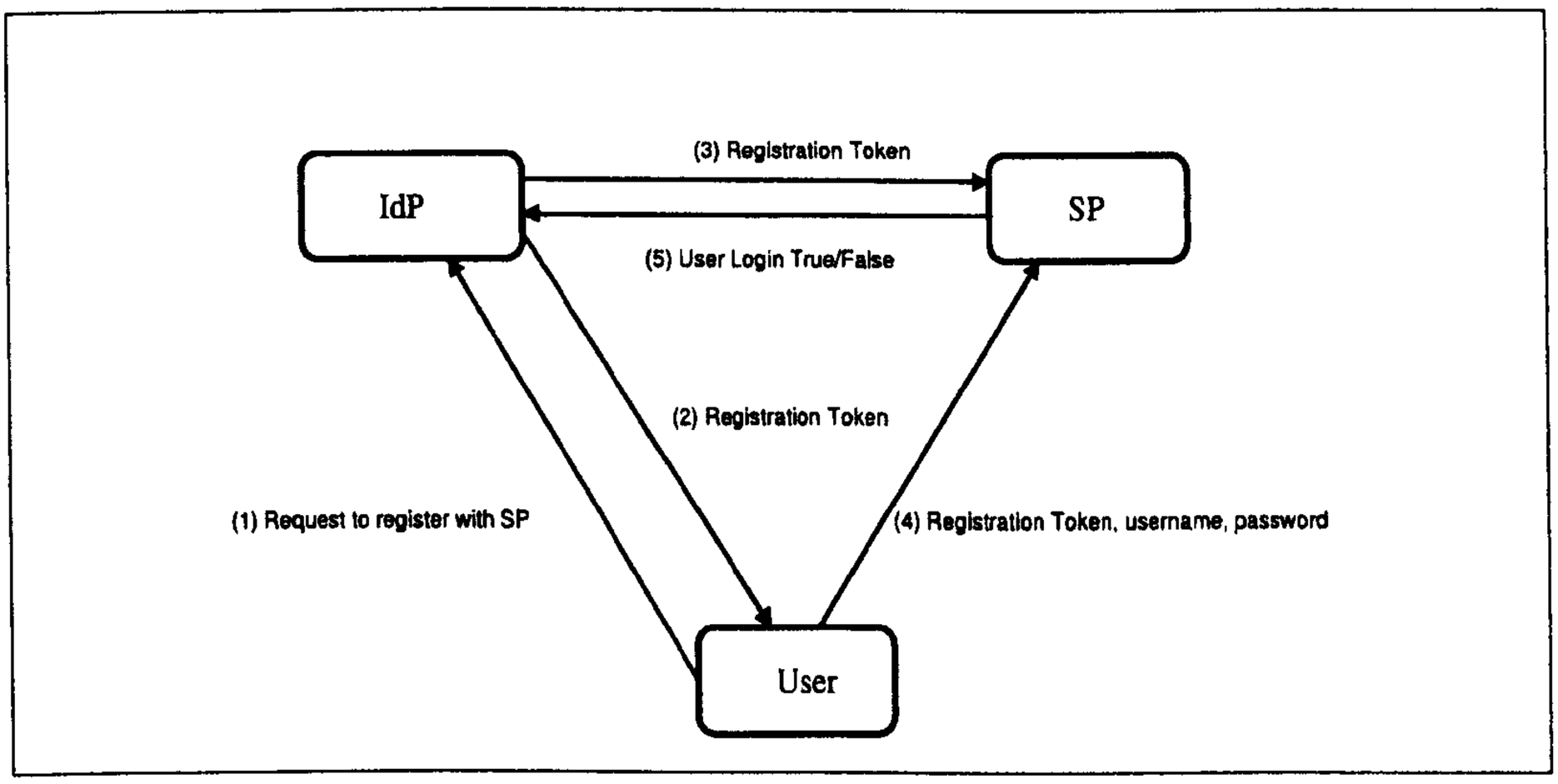

Figure 4.4: Registration process

4.8 Authentication Process and Mechanism

The authentication process starts with the user authenticating at the IdP using his/her authentication credentials usually a username and password. If this is successful, the IdP generates a "Security Token" and sends it back to the user. This Security Token will be used for the rest of the session when communicating with the IdP. The "Security Token" contains a timestamp and a 'token lifetime' after which the token will become invalid. The user's username and password are encrypted by the IdP's public key PK_{IdP} and signed by the user's private key SK_{User} to protect the message confidentiality and integrity. In response the user will receive the Security Token which will be encrypted with User public key and

signed with IdP private key.

Message 1 User → IdP: $s_{SK_{User}}$ ($e_{PK_{IdP}}$(username, password))

Message 2 IdP → User: $s_{SK_{IdP}}$ ($e_{PK_{User}}$(Security Token))

To access an SP the user sends the SP name (SPID) with the security token to the IdP; if the security token is valid and user has registered with the requested SP, the IdP generates a User Token to be used only with the intended SP, and sends to the user. Both messages are secured such that:

Message 3 User → IdP: $s_{SK_{User}}$ ($e_{PK_{IdP}}$(Security Token AND SPID))

Message 4 IdP → User: $s_{SK_{IdP}}$ ($e_{PK_{User}}$(User Token))

When the user approaches the SP with the User Token, the SP will forward the User Token to the IdP for validation. If the User Token has been successfully validated, the SP saves it in its database under the user's username, so next time when the user makes a request under the same User Token, the SP can validate it from the record in its database. The User Token has a defined lifetime after which the token becomes invalid, the login process is illustrated in Figure 4.5. The following is how the various messages have been secured:

Message 5 User → SP: $s_{SK_{User}}$ ($e_{PK_{SP}}$(User Token)) // Access request

Message 6 SP → IdP: $s_{SK_{SP}}$ ($e_{PK_{IdP}}$(User Token)) // Validation request

Message 7 IdP → SP: $s_{SK_{IdP}}$ ($e_{PK_{SP}}$(AccessResponse))

Message 8 SP → User: $s_{SK_{SP}}$(AccessResponse)

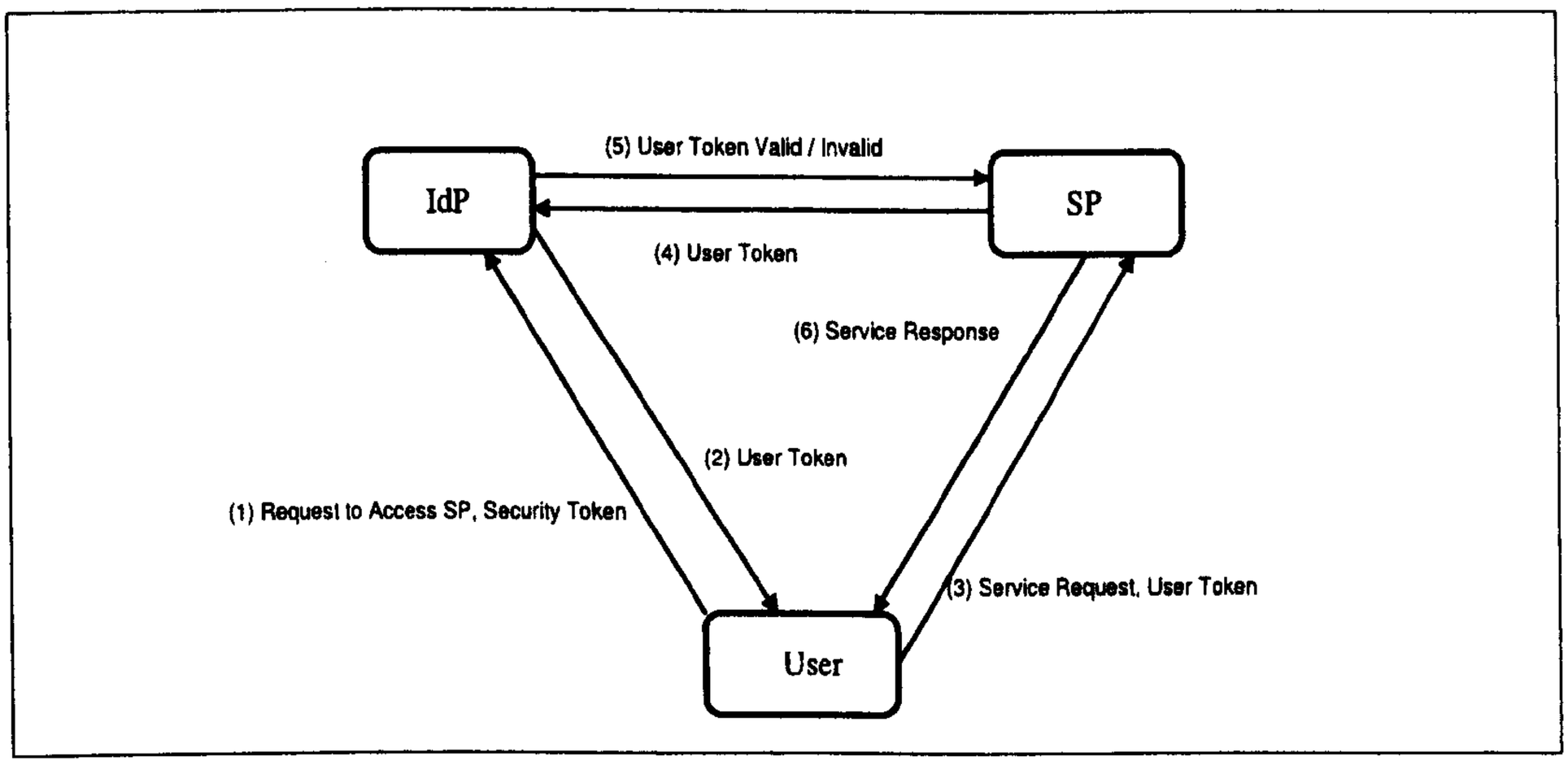

Figure 4.5: User Authentication

Global Logout

Also known as Single Logout, is provided by the system, such that once the user request to logout from the system, the IdP informs all the SPs to logout this user by setting the 'User Tokens' invalid. If the user requests global logout from an SP, the SP sends the logout request to the IdP. This request then informs all the SPs that the user has active sessions with, and sets all the User Tokens issued for this user to be invalid. In the case of the Liberty Alliance a Single Logout framework provides synchronized session logout functionality across all sessions that were authenticated by a particular identity provider in Federated Systems [51, 90].

Single Sign-On systems are suitable for the Web services environment, and can be a secure mechanism to authenticate users to various Web services. However, authenticating the users to the IdP or AS is challenging and usually is the weakest point in the system as it depends on human interaction such as remembering

username/password. On the other hand, GSM/UMTS network deploys strong "smart card" authentication mechanism as described in chapter four, and since many people these days have mobile phones (using GSM/UMTS technology) it make sense to try to integrate the two systems to improve security and provide better user experience. Next I propose a protocol that combines SSO systems with some of the GSM security features to secure M-Commerce system.

4.9 Privacy and Anonymity in Federated System with GSM

In this section we propose a novel system in which it is possible for a mobile user to securely authenticate and have full anonymity as far as the service providers are concerned. A feature of the system is that it is possible for a trusted authority to reveal the identity of the user if he or she is suspected of illegal activities.

A typical scenario is when a mobile user wants to download music or to book a train ticket using his/her mobile phone. To do that the user will login to the trusted third party server (e.g. the Network Operator) from which he will be able to access different service providers securely, confidentially and without the need to remember any username/password for any of these services providers. The system is built on top of the SSO architecture similar to the one presented by the Liberty Alliance; specifically we refer to the SSO work in [91].

4.9.1 System Architecture

The system consists of three players: Mobile Stations, Service Providers, and GSM Network Operators. The User operates a Mobile Station (MS) and wishes to access services provided by a service provider via a mobile network provided by a Network Operator (NO). The NO also provides a SSO service for the MS, and acts as an 'Anonymity Revocation Authority' to reveal the MS identity in some scenarios. Finally, the Service Provider (SP) provides services to the MS.

It is assumed that the NO is trusted by the MS not to reveal the User's true identity to any third parties, except in specified circumstances. It is further assumed that the NO has an asymmetric key pair for a signature scheme and the MS and the SP possess a trusted copy of the public key PK_{NO}.

It is also assumed that the MS is equipped with a SIM or connected to a SIM card, which shares a secret key with the NO, in line with the GSM security standards [92, 93].

In reality the secret key will be stored in a protected database in an Authentication Center which belongs to the NO; for simplicity we refer to the NO as the holder of the secret key, as the NO will also be managing the SSO system.

Figure 4.6 gives an overview of how the different players interact with each other, where the MS authenticates to the SSO system in the NO before accessing the services provided by the SP.

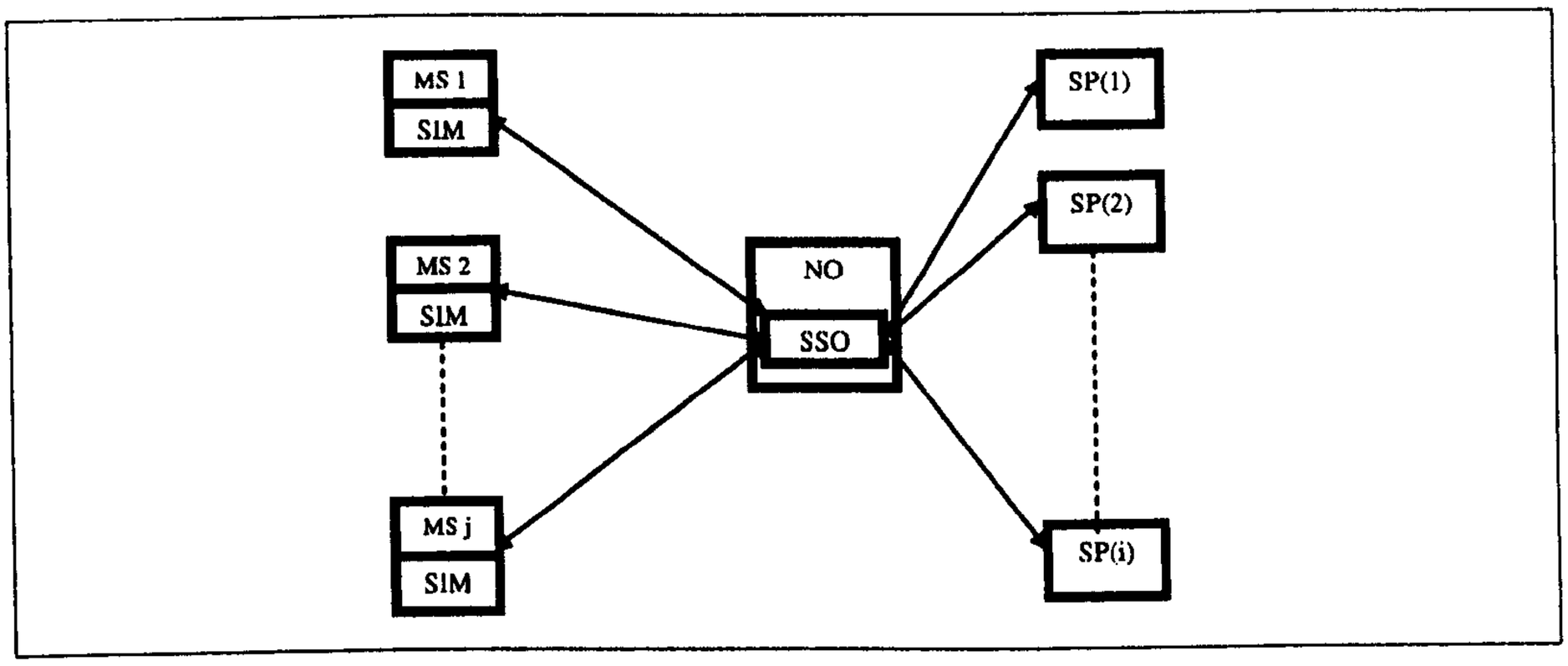

Figure 4.6: Proposed Architecture

4.9.2 The Protocol and System requirements

The main idea behind the system is that the user binds the secret key shared with its GSM network operator to a random ID. This ID acts as the user identifier with the service providers, and the only way to link this random identifier to the user is by knowing the secret key shared by the user and the GSM network operator.

The protocol description is divided into two parts. The 'Normal Scenario' which describes the operations required to allow the User to access the SPs with total anonymity, and the 'Revocation scenario' which describes how the NO can reveal the User identity if needed.

The following conditions must be met prior to the use of the protocol.

- All parties need to agree on a specific signature algorithm. The signature of data X using secret key K is written as $s_K(X)$.

- The NO has an encryption key pair for encryption scheme , and both the MS and the SPs possess a trusted copy of the public key PK_{NO}.

- The MS is equipped with a SIM or connected to a SIM card, which shares a secret key K_i with the NO, and supports the GSM security standards.

- The MS needs to generate a distinct signature key pair for each SP (PK_{SP} is the public key for SP, and SK_{SP} is the private key for SP). The public key does not have to be certified.

- The MS and its NO have agreed on a symmetric encryption algorithm which can use a secret key derived using the GSM A8 key derivation algorithm. The encryption of data X using secret key K is written as $e_K(X)$.

- The NO and SP have agreed on an asymmetric encryption algorithm, for which the encryption of data X using public key P is written $e_P(X)$.

- This protocol will work on top of other communication protocols (UMTS, GSM, etc) and therefore all parties should be able to communicate with each other.

4.9.3 The Normal scenario

The following steps describes the interaction between the MS, the NO and the SP.

1. The MS authenticates to the NO using regular GSM authentication, and then authenticates to the NO SSO system, e.g. using user name and password, to be able to request access to an SP.

2. The NO generates a random challenge (RAND) and sends it to the MS.

3. The MS generates a Security Token, consisting of two parts: the MS ID which is a random string to be used as the MS identifier, and a 'Revocation Parameter' which can be used to reveal the MS identity to a trusted authority, if needed. The Security Token (ST) is generated as follows:

   (a) The MS uses RAND and the long term secret key K_i (stored in the SIM) to compute a secret session key K_c as follows: $K_c = A8_{K_i}(\text{RAND})$, where A8 is a network specific key derivation function used by the GSM system [9, 92, 93].

   (b) The MS will generate the distinct signature key pair for each SP, public key PK_{SP} will act as the MS ID, and the private key will be bound to the true identity of the MS(as described below). This key will be different for every SP.

   NOTE: For a higher level of user privacy, it is possible to generate a different key pair every time a service is requested, even if it is for the same SP. As a result the SP will be unable to link two different acts of the same user, for example what products the same MS is buying.

   (c) The MS generates the Revocation Parameter (RP) by signing a message m with the private key SK_{SP} and then symmetrically encrypting the signed message with the session key K_c, such that,RP=$e_{K_c}(s_{SK_{SP}}(m))$, where m is a short message containing the MS IMSI and RAND, used to re-generate K_c.

   (d) MS will create the Security Token (ST) shown in Figure 4.7, which will include the following attributes:

   - MS ID, which will be the public key used only with a particular SP, i.e. the MS ID = PK_{SP}.

- The Revocation Parameter (RP).

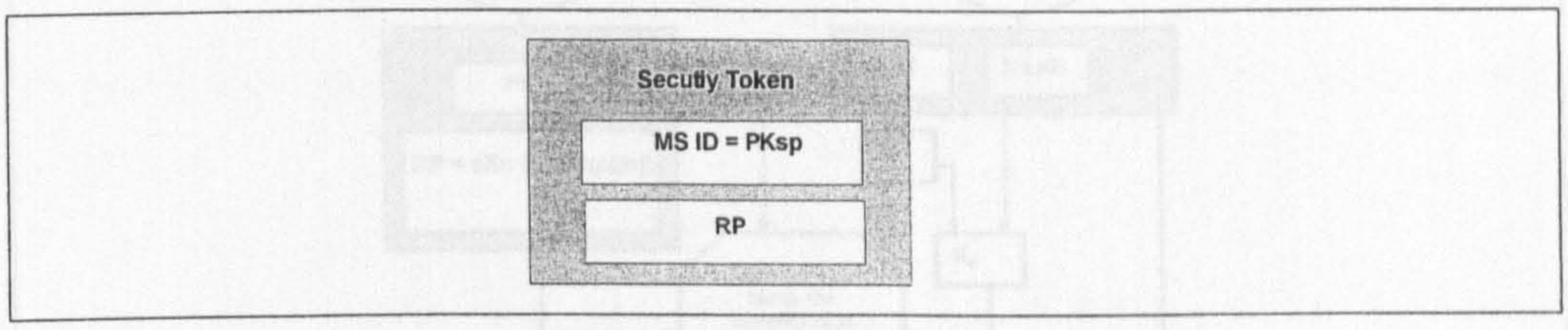

Figure 4.7: Overview of the Security Token

4. The MS sends its request to the NO, including the Security Token.

5. The NO checks the request from the MS. This is done as follows (as illustrated in Figure 4.8):

   (a) The NO uses the secret key K_i associated with the MS IMSI and the random challenge RAND that has been sent to the MS to generate m, which consists of IMSI and RAND, and to generate the secret session key K_c. (For this reason the NO needs to store the RAND it sent to the MS previously in the protocol.)

   (b) The NO will first decrypt the RP obtained from the security token using K_c (that it has just generated) and verify the signature on m using PK_{SP}, obtained from the security token. The NO will then compare the two values of m(i.e. the one it has just generated and the one recovered from the RP) and if they are equal it confirms that the MS has used the same RAND that the NO sent.

6. If this check is successful the NO will create a Revocation Attribute (RA) by adding the RP to m (which consists of the MS IMSI and RAND) and encrypt the new m' (which consists of the MS IMSI, RAND, and RP) with the NO public key, i.e. RA $= e_{PK_{NO}}(m')$.

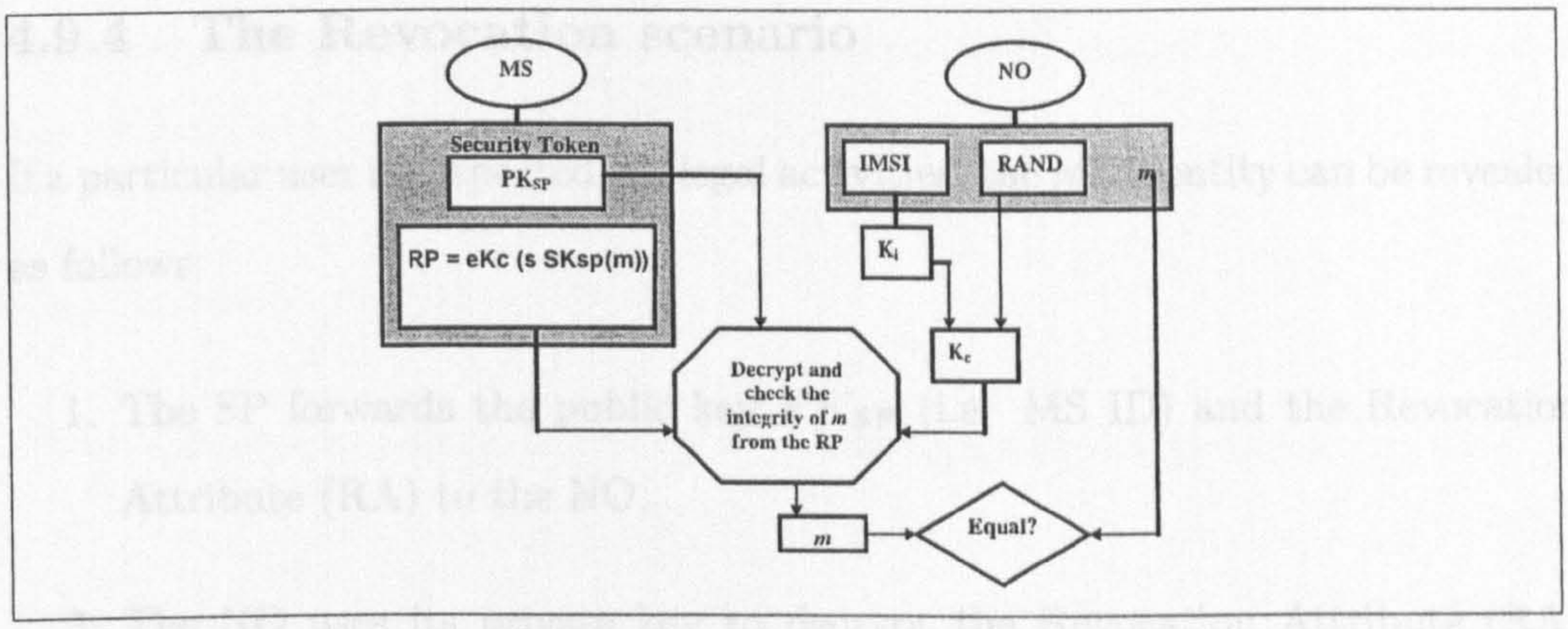

Figure 4.8: Security token verification process

7. Using its SSO system such as [52] the NO forwards the MS request to the SP using PK_{SP} as the MS ID. Attached to the message will be the RA. It is assumed that the NO is trusted to send the correct RA to the intended SP.

8. As the user been authenticated using the SSO system, the SP will deliver the service requested by the user, as shown in step 5 in Figure 4.9.

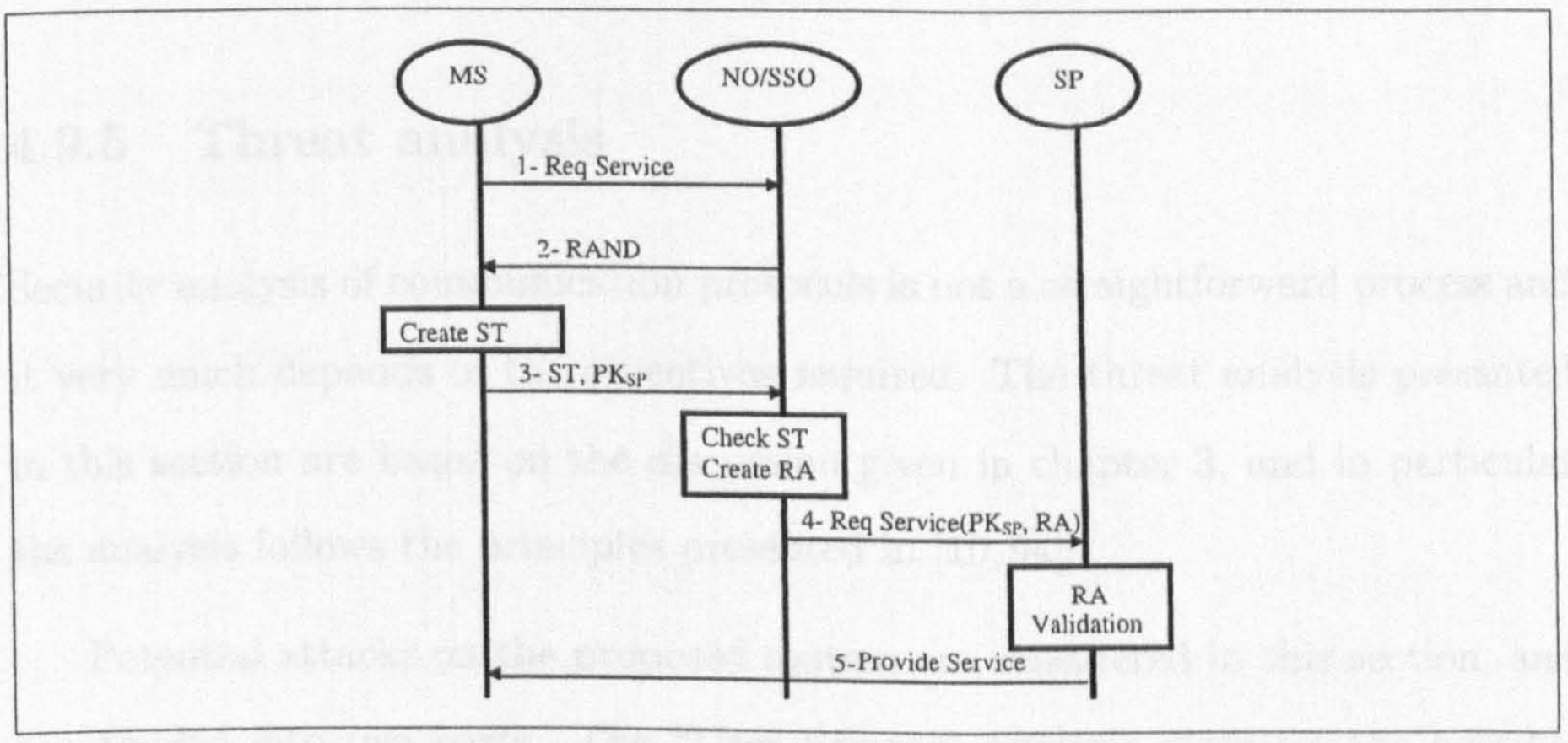

Figure 4.9: Normal scenario overview

4.9.4 The Revocation scenario

If a particular user is suspected of illegal activities, the MS identity can be revealed as follows:

1. The SP forwards the public key PK_{SP} (i.e. MS ID) and the Revocation Attribute (RA) to the NO.

2. The NO uses its private key to decrypt the Revocation Attribute (RA) to get m', then it uses the secret key K_i associated with MS IMSI and the random challenge RAND (contained in m') to re-generate the secret session key K_c.

3. The NO decrypts RP using K_c and check the integrity of m using PK_{SP} from the RP. If successful (i.e. the decryption and the integrity check) will indicate that owner of this public key PK_{SP} is the same as the owner of the secret key K_i who is linked to IMSI number, and therefore the user identity can be revealed.

4.9.5 Threat analysis

Security analysis of communication protocols is not a straightforward process and it very much depends of the objectives required. The threat analysis presented in this section are based on the discussion given in chapter 3, and in particular the analysis follows the principles presented in [10, 94].

Potential attacks on the proposed system are considered in this section, and are divided into two parts. The "User Privacy" analysis discusses the possible attacks that could compromise user privacy. The second part, "Other Attacks",

considers other possible threats to the system which are not directly related to the user privacy.

User Privacy

These are possible attack scenarios that can potentially damage user privacy.

Attack on the communication links

A network eavesdropper who captures any of the communication messages in the system will not be able to gain access to the information about the user as all the user related data are encrypted.

Attack on the MS

If the secret key K_i can be extracted from the SIM, either by stealing or cloning the SIM, the attacker can claim to be the subscriber to the NO; however this is unlikely to be a major threat to the proposed system as the user needs to authenticate to the SSO system by another authentication mechanism, most likely using username/ password mechanism.

Attack on the Network Operator/Anonymity Revocation Authority

It was pointed out earlier that the NO is trusted by the MS not to reveal the user's true identity to any third parties, except in specified circumstances. But as in most SSO schemes if an attacker breaks into the NO, the system will fail as it is the 'single point of failure'. If successful, the attacker could read all the

user private data. However the NOs are highly secure against most attacks and therefore the likelihood of such failure is very small.

Attack on the Service Provider

If an attacker manages to get access to the SP system or even of the SP itself tries to gain information about the user it will not be able to do so, because the user is known to the SP by a random ID with no link to user true identity. To gain any information from the RA the SP needs the NO private key which it doesn't have.

Timing attack

The aim of this attack is to reveal the user identity by monitoring the network traffic. The MS makes a request to access some service and after a short period of time (after receiving the RAND and sending the ST) the NO sends a message to a SP, an eavesdropper who is monitoring the network traffic can conclude with some probability that this User is trying to access this SP, and if the system at that time has limited number of users, the eavesdropper can make the same conclusion with higher probability. This eavesdropper could be a dishonest SP, who wants to know more about its competitor business. Possible solution is to increase the time between the receiving and sending of another message at the MS and/or the NO side. This could increase the possibility that the NO will make more contact with many SPs, which will make it more difficult to conclude which user is trying to contact which SP with some certainty.

Other attacks

These are other possible attacks on the system that are not directly linked to the privacy of the user.

Attack on communication links

An attacker may not be able to read the messages sent in the system, but it is still possible that he/she can modify the network traffic. If for example that attacker manages to modify message 2 in Figure 4.9 (i.e. the RAND sent from NO to MS), the system will fail. Therefore integrity protection is fundamental to the operation of such a system. This problem can be overcome with the use of some established techniques such as SSL/TLS with client/server certificates [17].

Replay attack

A dishonest SP could impersonate to be the NO to another SP, by capturing message 3 in Figure 4.9 (i.e. the Security Token sent from MS to NO) and later replaying the message after creating its Revocation Attribute (RA) to another SP. The effect of such attack may not be great, but it could be damaging in certain applications.

4.10 Summary

Federated environment is used in Web services, Single Sign On do play an important role in building federated system. In this chapter we reviewed SSO systems in some details in particular the issue of trust between the various entities in the systems and the concept of trust.

4.10 Summary

Improving security is one of the SSO systems objectives. This chapter highlighted some of the main security challenges, in particular during the registration and authentication process and we have proposed mechanisms to address these concerns.

User anonymity is an issue in such systems, and the GSM security features can help in protecting user privacy as demonstrated by our novel protocol. The aim of the proposed protocol was not to provide total anonymity to the user as this may be misused in M-Commerce applications, therefore the aim was to provide total anonymity from the point of view of the services providers only. It was described how this can be achieved through the use of some GSM security features in combination with a digital signature scheme for SSO system, with the assumption that the mobile network operators are the trusted entities.

Chapter 5

Secure Authentication for Mobile Web Services

5.1 Introduction

The previous chapter presented Single Sign On as a a way to improve the overall security of interworking systems such as Web services, by securing the authentication process. The management and the handling of the security credentials is a major concern in the developments of the security system .

This is in fact part of a bigger challenge known as the "challenge of portable identity". Web services increasingly cross organizational boundaries, yet there is no standard way to convey security attributes about individuals from one organization to another, especially in terms of how individuals or entities are identified and how permissions for access to resources are specified.

Web Services use SOAP to connect machines and applications. However; when two entities with different trust models want to interact, SOAP has no

standardized and interpretable way to communicate their security properties to establish trust. One of the proposed solutions is the Security Assertion Markup Language (SAML). SAML is used to exchange authentication and authorization credentials across different security domains, which we will address in more details in the this chapter.

Another interesting technology for authentication services is the Generic Authentication Architecture (GAA) [95], it is part of the 3GPP UMTS framework and, can play an important role together with SSO systems to improve the security for Mobile Web services applications. In the second part of this chapter we propose an novel architecture which integrate the Liberty Alliance Federation system with UMTS GAA.

5.2 Security Assertion Markup Language (SAML)

SAML is an XML based security standard created to enable portable identities and the assertion of these identities. SAML is used to exchange authentication and authorization credentials across different security domains. As SAML is XML based its not tied to any transport or platform, also it is not depended on any central certificate authority to issue certificates and this is very important in the Web services environment.

SAML V1.0 became an OASIS standard in November 2002. SAML V1.1 followed in September 2003 and has seen significant success, gaining momentum in financial services, higher education, government, and other industry segments. SAML has been implemented by all major Web access management vendors. SAML support also appears in major application server products and is commonly

found among Web services management and security vendors. SAML V2.0 builds on that success.

SAML V2.0 unifies the building blocks of the federated identity in SAML V1.1, and was developed by the Security Services Technical Committee of OASIS. The SAML V2.0 OASIS Standard specification set was approved on 15 March 2005 [27].

5.2.1 Motivation for SAML

Securing identity is fundamental for Web services security, and as the identity of valid users must move around when information moves from one trust domain to another, and the fact that Web services will be used to cross trust domains makes portable trust an important requirement for Web services security.

One of the biggest challenges for Web services is that of user authentication, and single sign-on across a number of federated systems [17]. As proposed by the Liberty Alliance [52], SAML provides distributed authorization and federated identity management, and does not impose a centralized, decentralized, or federated infrastructure or solution, but instead facilitates the communication of authentication, authorization, and attribute information.

5.2.2 The SAML Specification

There are four main components of SAML. They are:

- Assertion: an XML schema and definition for security assertion.

- Request and response protocol: an XML schema and definition for a request/response protocol.
- Binding: rules on using assertion with standard transport and messaging frameworks.
- Profiles: the rules for embedding, extracting and integrating.

The assertion can convey information about authentication acts performed by subject, attributes of subjects, and authorization decisions about whether subject are allowed to access certain resources. The protocol defines an agreed way of asking for and receiving information. Binding define how SAML messages are communicated over standard transport and messaging protocols. For example, a SAML SOAP binding describes how SAML request and response message exchanges are mapped into SOAP message exchange. This is illustrated in Figure 5.1 below. Full details of SAML specifications can be found in [27, 74, 96, 97].

5.2.3 Operation of SAML

SAML assertion are encoded in a common XML schema, which includes basic information and the claims the requestor is making, for example "I claim to be John". The basic information specifies a unique identifier used for the assertion name, date and time of issuance, and the time interval for which the assertion is valid. Here is a simple SAML assertion

<saml:Assertion>

MajorVersion="1" MinorVersion="0"

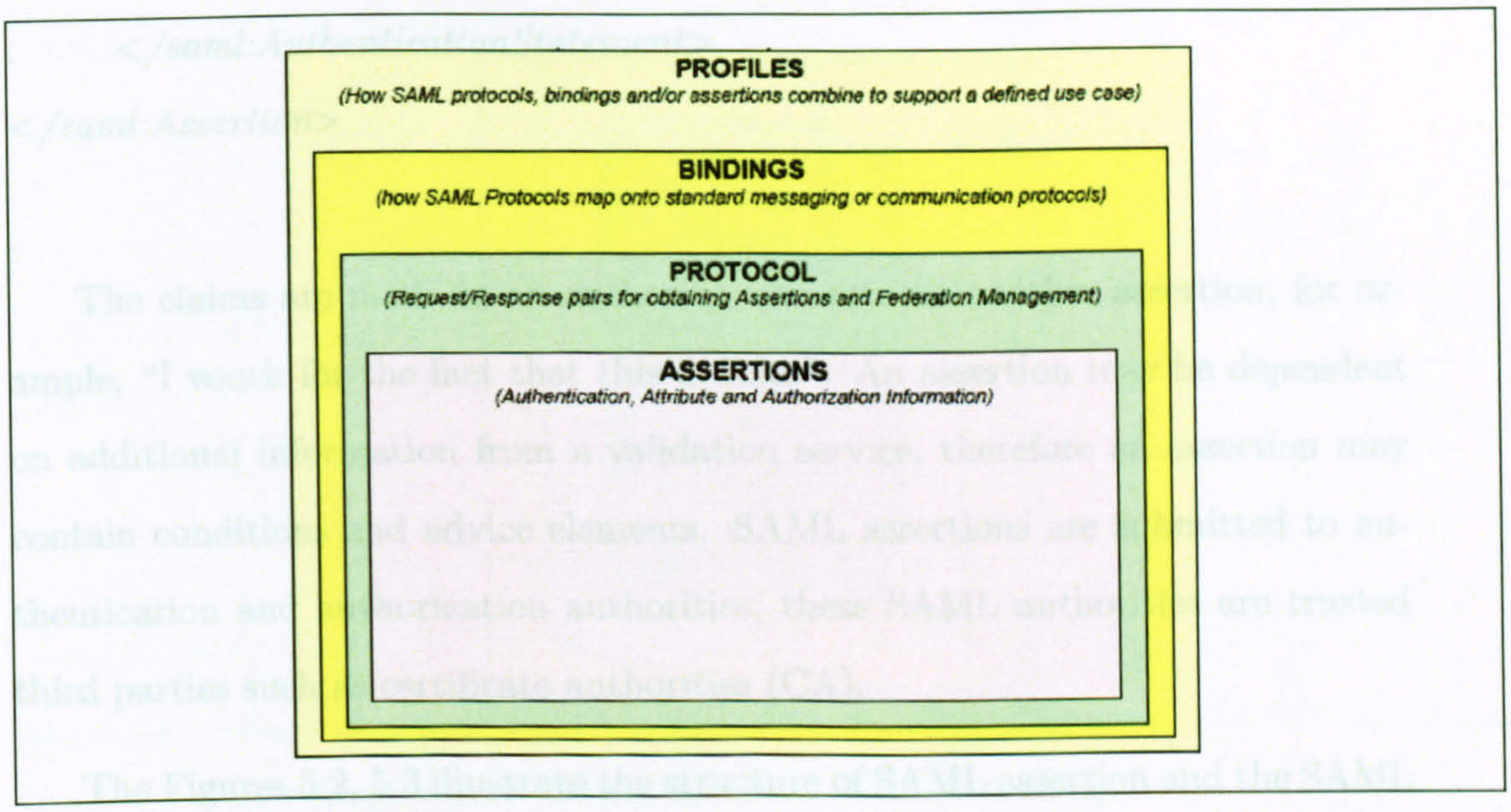

Figure 5.1: An Overview of SAML components

. *AssertionID="138.40.160.163"*

. *Issuer="City.ac.uk"*

. *IssuerInstance="2006-10-30T09:50:00GMT"*

. *<saml:Conditions*

. *NotBefore="2006-10-30T09:50:00GMT"*

. *NotAfter="2006-10-30T09:59:00GMT"/>*

. *<saml:AuthenticationStatement*

. *AuthenticationMethod="password"*

. *AuthenticationInstant="2006-10-30T09:50:00GMT"/>*

. *<saml:Subject>*

. *<saml:NameIdentifier*

. *SecurityDomain=""Lab.city.ac.uk"*

. *Name="John"/>*

. *</saml:Subject>*

. *</saml:AuthenticationStatement>*

</saml:Assertion>

The claims are made to an authority who returns another assertion, for example, "I vouch for the fact that this is John". An assertion may be dependent on additional information from a validation service, therefore an assertion may contain conditions and advice elements. SAML assertions are submitted to authentication and authorization authorities; these SAML authorities are trusted third parties such as certificate authorities (CA).

The Figures 5.2, 5.3 illustrate the structure of SAML assertion and the SAML SOAP/HTTP binding.

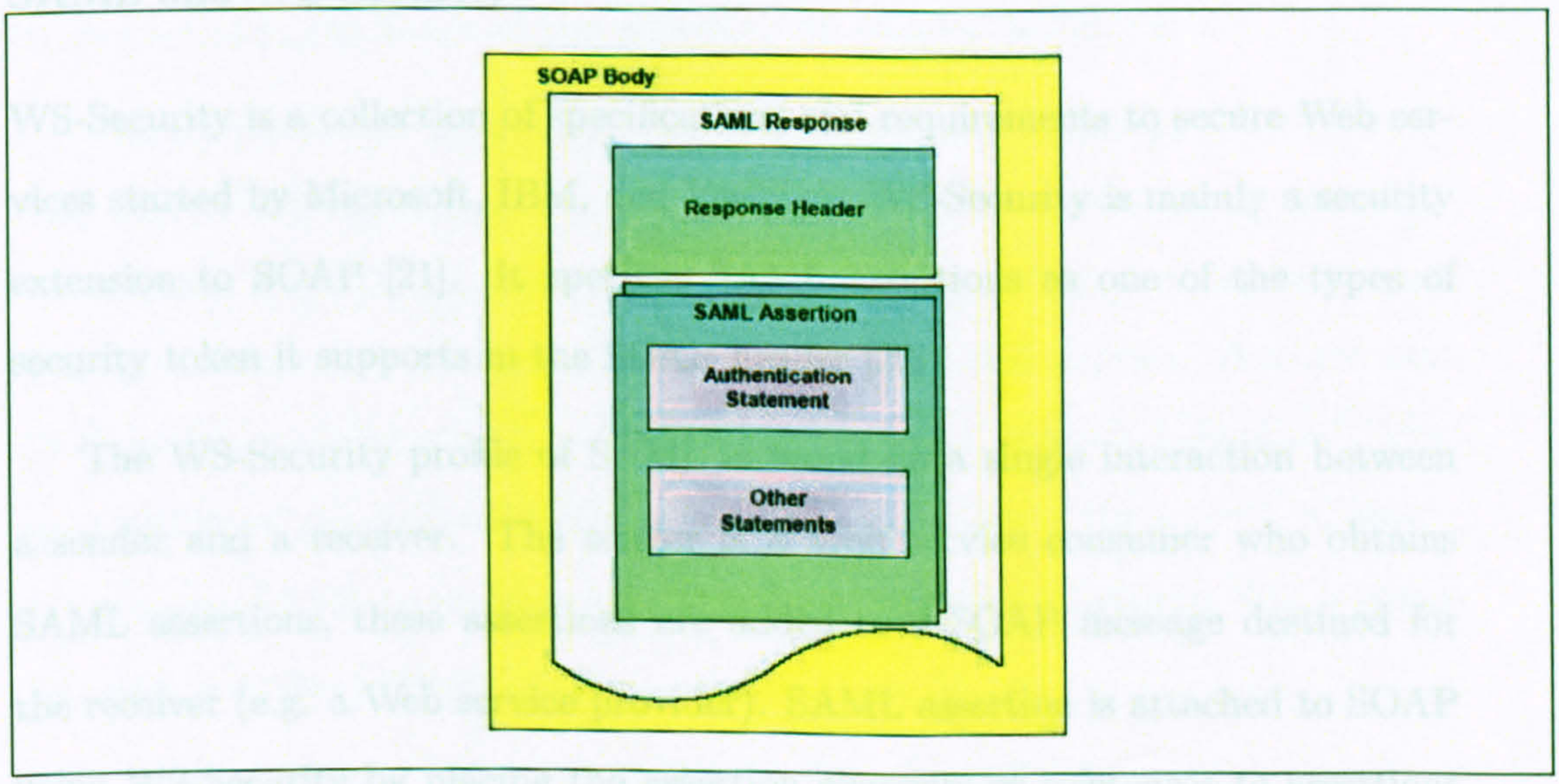

Figure 5.2: SAML assertion structure

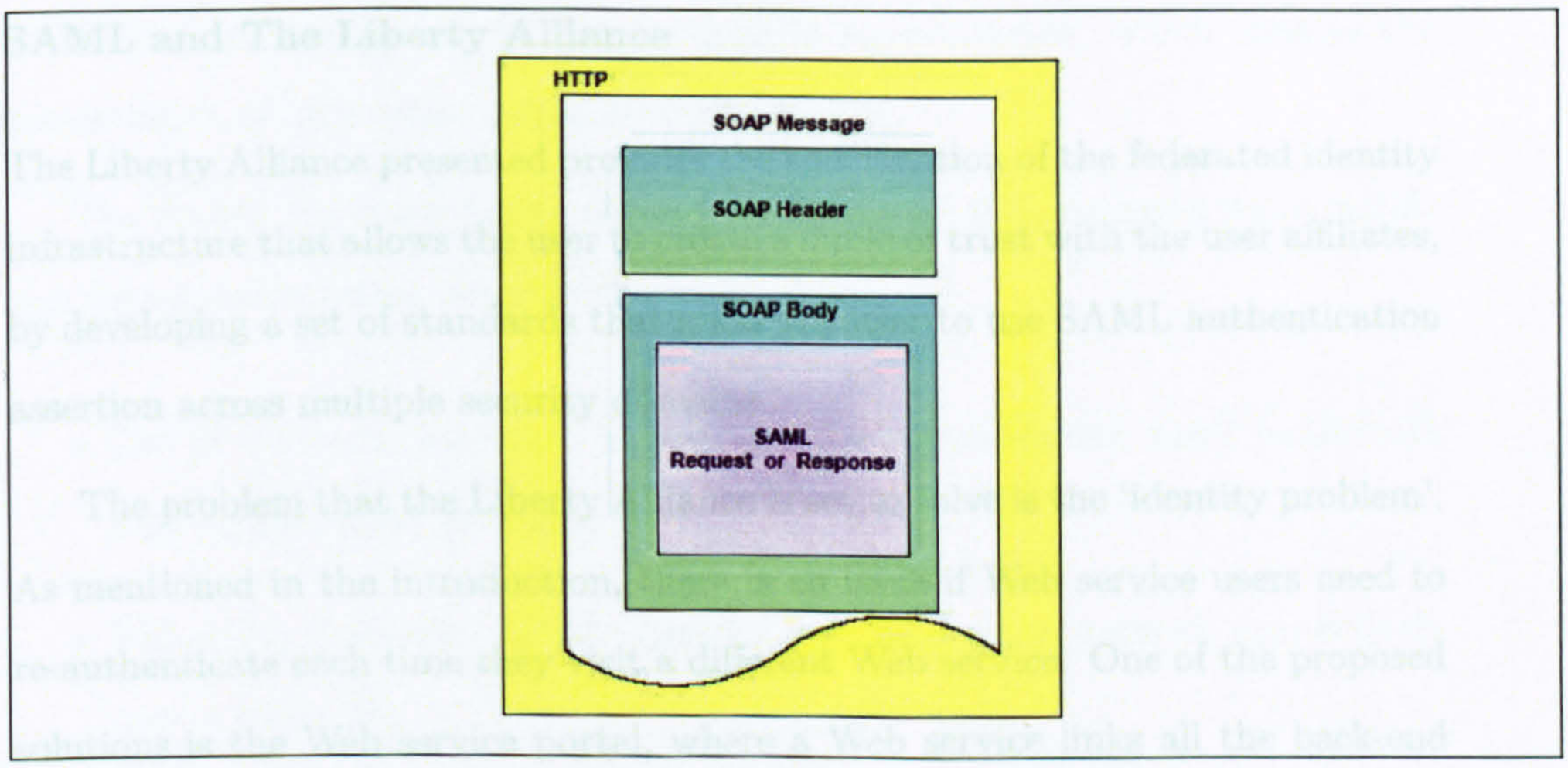

Figure 5.3: SAML with SOAP/HTTP binding

SAML and WS-Security

WS-Security is a collection of specifications and requirements to secure Web services started by Microsoft, IBM, and VeriSign. WS-Security is mainly a security extension to SOAP [21]. It specifies SAML assertions as one of the types of security token it supports in the SOAP header [22].

The WS-Security profile of SAML is based on a single interaction between a sender and a receiver. The sender is a Web service consumer who obtains SAML assertions, these assertions are added to a SOAP message destined for the receiver (e.g. a Web service provider). SAML assertion is attached to SOAP using WS-Security by placing the assertion elements or references to assertions inside a <wsse:Security> header.

SAML and The Liberty Alliance

The Liberty Alliance presented provides the specification of the federated identity infrastructure that allows the user to create a circle of trust with the user affiliates, by developing a set of standards that allow the user to use SAML authentication assertion across multiple security domains.

The problem that the Liberty Alliance is set to solve is the 'identity problem'. As mentioned in the introduction, there is an issue if Web service users need to re-authenticate each time they visit a different Web service. One of the proposed solutions is the Web service portal, where a Web service links all the back-end resources of different Web service providers into a Web front end. Microsoft .Net Passport [56] follows this model. This model has some drawbacks as it puts too much control in the hands of one provider. As detailed in chapter 4.

The alternative to the portal solution is the federated identity which is basically shared authentication. The way this works is: a user will login to one Web service, then when the user would like to visit a different Web service, the second Web service can rely on the work done by the first Web service. How the two Web services establish this shared identity is known as "identity federation". The specification of this process is one of the main objectives of the Liberty Alliance. Liberty uses SAML to transport a set of SAML assertions between Web services to create a circle of trust.

SAML in Mobile Web services

Mobile devices may offer a more constrained environment for the identity management than do other devices such as personal computers on an enterprise network.

SAML defines one specific profile for mobile environments, which enables the transmission of an artifact using WAP/WML [98].

The SAML Implementation Guidelines document [99], suggests some mobile specific implementation considerations:

- Use of the radio resource: In some mobile environments, radio bandwidth may be restricted or costly.
- Reliability/Latency: Mobile devices may have poor network connectivity over a radio link.
- Ease of deployment: To enable handset usage of the SAML profiles may in some cases require the deployment of handsets that utilize additional or improved software.
- Presence of SIM card: GSM-based networks make use of the Subscriber Identity Module (SIM) card. Such cards may provide enhanced security for identity-based transactions. This is also true with USIM in UMTS.
- Network roaming: Mobile roaming business agreements established between network operators provide an important basis for trust between SAML providers.
- Link security: WAP 1.x does not allow for secure, encrypted links at the transport layer between a mobile device and a service provider. WAP 2.0 introduced TLS which does allow for such links.

5.3 Related Work

The different characteristics of wireless networks require looking at the threat differently. An interesting approach for looking at security threats are presented in [100], where the author considers threat from two points of view: the insider and the outsider. The type of threats that should be considered for such environment includes: traffic analysis; passive and active eavesdropping; unauthorized access; man-in-the-middle attack; session highjacking; and replay attacks. The author concludes with counter measure for the above threats by implementing four security components: mutual authentication; encryption for the payload; strong integrity measures; and firewall.

In the other hand when developing a security for a Web services architecture, there are many Web services security standards developed by important consortiums such as W3C, IETF, and OASIS which needs to be taken into account. However there is no process to guide the developer in integrating security through the life cycle of the development. This issue was the focus of the work in [101], where the authors proposes a process for Web services security. This process designed on the basis of traceability and re-usability of the products.

While the majority of works on Web service architecture is based on client/server models (centralized based), in [102] the authors proposes an interesting mechanism for fair exchanges between mutually distrustful yet collaborating web services, based on peer-to-peer settings. The authors propose that the system is useful both at the system level (to build higher-level services) and at the application level (to provide end-user functionality).

The Multimedia Laboratories group at NTT DoCoMo developed a scalable

security description framework for mobile Web services [103], in which SAML is used as an 'e-ticketing service token' that describes service information which includes token issuer, user identifier, timestamp and the service context. Although this work refers to the use of XML encryption and XML signature to improve security, it does not address key security risks associated with such environment. For example WAP has been proposed but there was no reference to the issue of the WAP gap [9] and how to reduce such risk. The authors are treating the mobile device just as a computer terminal with limited computer powers and not taking any advantages of the security provided by the mobile environment.

One of the key papers that analyse SAML from a security perspective is by Thomas GroB [104]. This paper gives general analysis of the security of SAML V1.0 Browser/artifact. The paper presents three attacks and it suggests possible countermeasures. The main attacks presented are, Replay attack, Man-in-the-Middle attack, and HTTP Referrer Attack. Although these attacks are different in nature their origin in SAML protocol V1.0 is the same. Because of the lack of authentication between the different communicating parties (i.e. User/Browser, Source, Destination), the impact of these attacks could be eliminated or greatly reduced by using SSL/TLS to secure the communication parties between the various entities. However, even with that [104] highlights possible vulnerability of the SSL/TLS Binding, specifically when an entity has several services on the same destination site. It is possible thenfor one service holder tries to cheat another. The above analysis was taken into consideration by the OASIS Security Service Technical Committee response (SSTC) in a committee draft [105]. In response to the papers analysis, the SSTC has incorporated some changes into the SAML V2.0 specifications.

Another issue associated with SAML is delegation. Delegation is an important aspect of commercial Web services. Current delegation systems such as X.509 proxy certificates have the problem that commercial Web Services tools fail to recognize these certificates or process them properly. SAML has a broadening commercial support but lacks delegation capabilities. The work in [106] proposes a delegation framework that uses SAML assertion for both direct and indirect delegations. This delegation framework is based on the SAML Attribute statement and SOAP binding.

Yuen-Yan JChan in [107] shows how it is possible to launch a "Weakest Link Attack" on a SSO system that uses SAML V2.0. This work exploits some of the optional requirements in SAML V2.0. A scenario is analysed where an adversary launches a concurrent service request at two service providers that require different authentication levels (e.g. one using X.509 PKI based authentication and the other using a simple username password authentication). Assuming the adversary can legitimately have access to one of the SPs (e.g. the one that uses username password authentication), he will also be able to access the other SP. This attack can be launched if the following conditions are met: User redirection for request and response messages is required in the SSO system. And at least one of the following conditions in the response message is true: The level of authentication is not indicated or implied; the subject being authenticated is not indicated or implied; or there is not integrity for the message segment that binds the subject and the authentication level.

This is mainly because the <AuthnContextClassRef> is contained in <AuthnContext> that is contained in <AuthnStatement> are not mandatory to be included in the <Assertion> in SAML V2.0. This means it is not mandatory to include the

indicator of the required authentication level nor to include the indicator of the authentication subject in the response. The proposed solution is to include the above indicators and to sign the response message.

As it has been discussed in this thesis the process of integration of Web services and mobile network is not straightforward. Example of this is [108] where the authors propose a framework for mobile Web services based on a "Mobile Agent", to make Web services more convenient and efficient by taking advantages of the location information in the Web service. However, this proposal fails to address the basic security needs, such as authentication and privacy; also the "Personal Agent Server" is designed as WAP gateway, where translation between WAP and TCP take place. The authors also make use of the mobile operator ability to provide users' location based data, but does not suggest any measures to insure users' privacy.

Another interesting approach of using "Agents" is proposed in [109]. The agent is run at the service host to minimis the work by the mobile device. However; the paper fails to address important security risks such as passive and active attacks and is subject to man-in-the-middle attacks as no integrity protection in place.

5.4 Global Mobile Telecom Industry Market Trends

The growth in number of users using mobile phones in the last 10 years or so has been phenomenal. Now reaching 1.12 billion units world wide according to 2007 data from Strategy Analytic, there are more mobile phones than PCs!

Mobile phones are becoming more than just devices to make calls; most of

the latest new mobile phones have built in camera, personal organizer, radio, and dozens more applications. There are two main reasons for that. The first is that mobile phones processors are becoming more powerful with more memory built inside them; this has enabled the possibility to add more hardware and software into them, hence; more applications. The second reason which is more importantly is that in many regions in the world (especially the developed world) they have reached over 100 percent user penetrations, which means that the revenue from phone calls almost saturated. And therefore, the leading players in the telecom industry (network operators and handset manufacture) needed to find new streams of revenue. This has lead to the development of a new market known as Value Added Services or VAS.

In telecommunication industry VAS is every service beyond core service such as voice calls and faxes (sometimes even Short Messaging Services SMS is considered core service). Many operators offers their own VAS service however, many gets their VAS through VAS provider or content providers. Type of applications for VAS varies including:

- Mobile gaming
- Mobile personalization (wallpaper, Ring Back Tones (RBT))
- Mobile music
- Mobile TV
- Messaging (Multimedia Messaging Service (MMS), Instant Messaging (IM))
- M-Commerce
- Emails

5.4 Global Mobile Telecom Industry Market Trends

The global VAS market is big; the following table demonstrate the size of the market.

Region	Mobile Subscribers in 07	Data Revenues in 07
EMEA	785m	USD54b
US and Canada	235m	USD18b
Central Latin America	231m	USD4b
Rest of Asia	357m	USD32b
China	521m	USD7b
India	103m	USD1b

Source: Morgan Stanley, IDC 2005

Screen Digest predicts that by 2011 mobile television will be the dominant service in Europe (generating revenues of 4.7 billion). In second place will be mobile gaming with 2.0 billion revenues and music comes in third with 1.47 billion.

According to the Informa Telecom and Media, in2004 UK ringtone sales reached 174m, 216 percent more than the 80m spent on CD singles. Ring tones currently account 34 percent of the mobile content revenues. The Figure 5.4illustrate the worldwide revenue from mobile personalization (Source: Informa Telecom and Media).

According to Frost Sullivan, USD 25B global market for m-commerce by 2006 representing 15 percent of the Global On-line Commerce. The Figure 5.5illustrate the scope of mobile commerce transaction.

As the above figures shows the mobile telecom market is serious business and

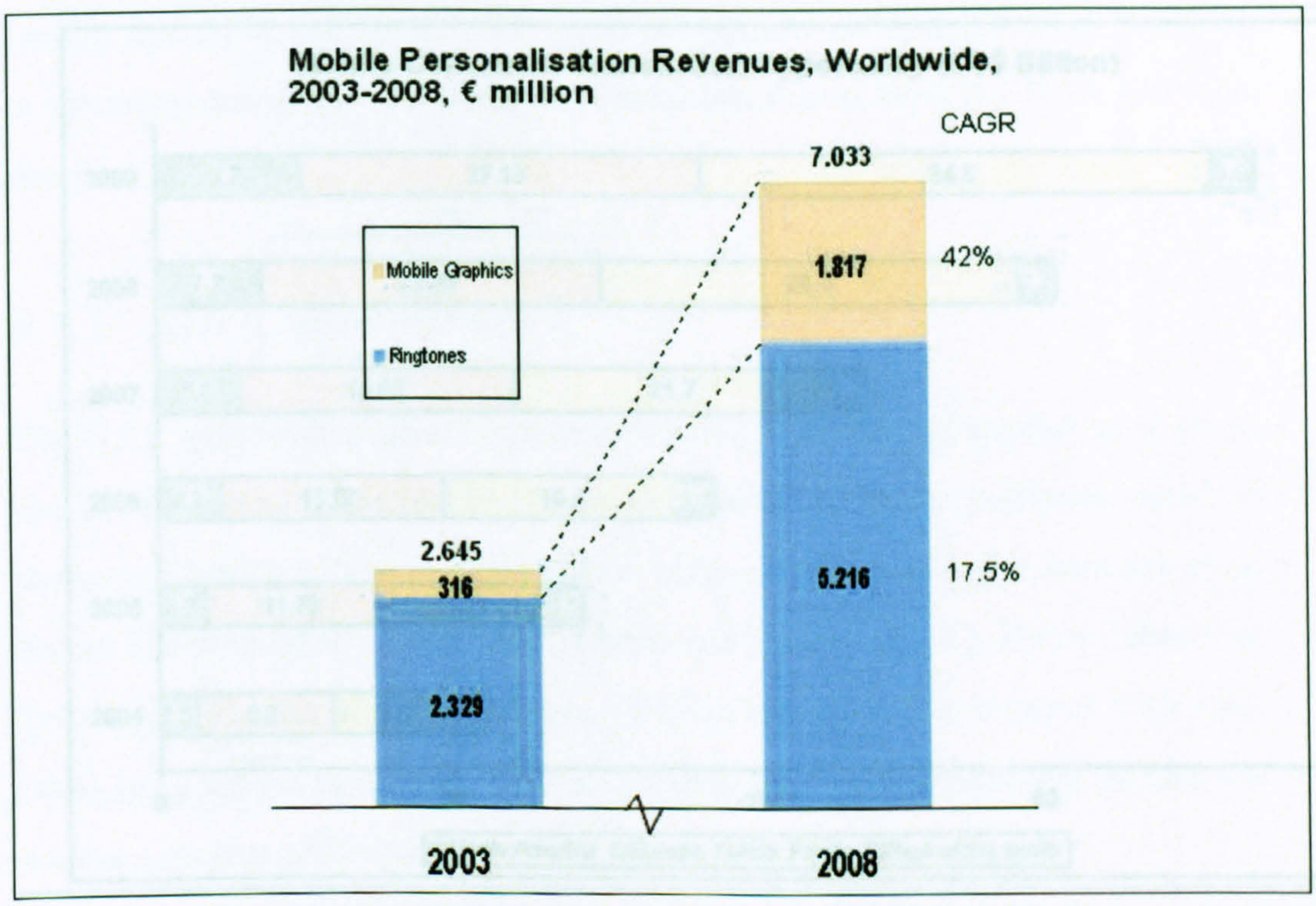

Figure 5.4: Mobile Personalization Revenue Worldwide

is growing, and as mobile user penetration raises the trends is moving toward VAS. Innovative application in particular in M-Commerce will require the solutions to be more flexible and secure. For these reasons we believe mobile Web service will play a more leading role in M-commerce platforms; however, such applications will only succeed if they have the right level of security protection. Therefore, we do think that the topic of this thesis and its finding is very relevant not only to the research and academic community but also to the business sector, as it touch's key issue of potential future mobile Web services solutions.

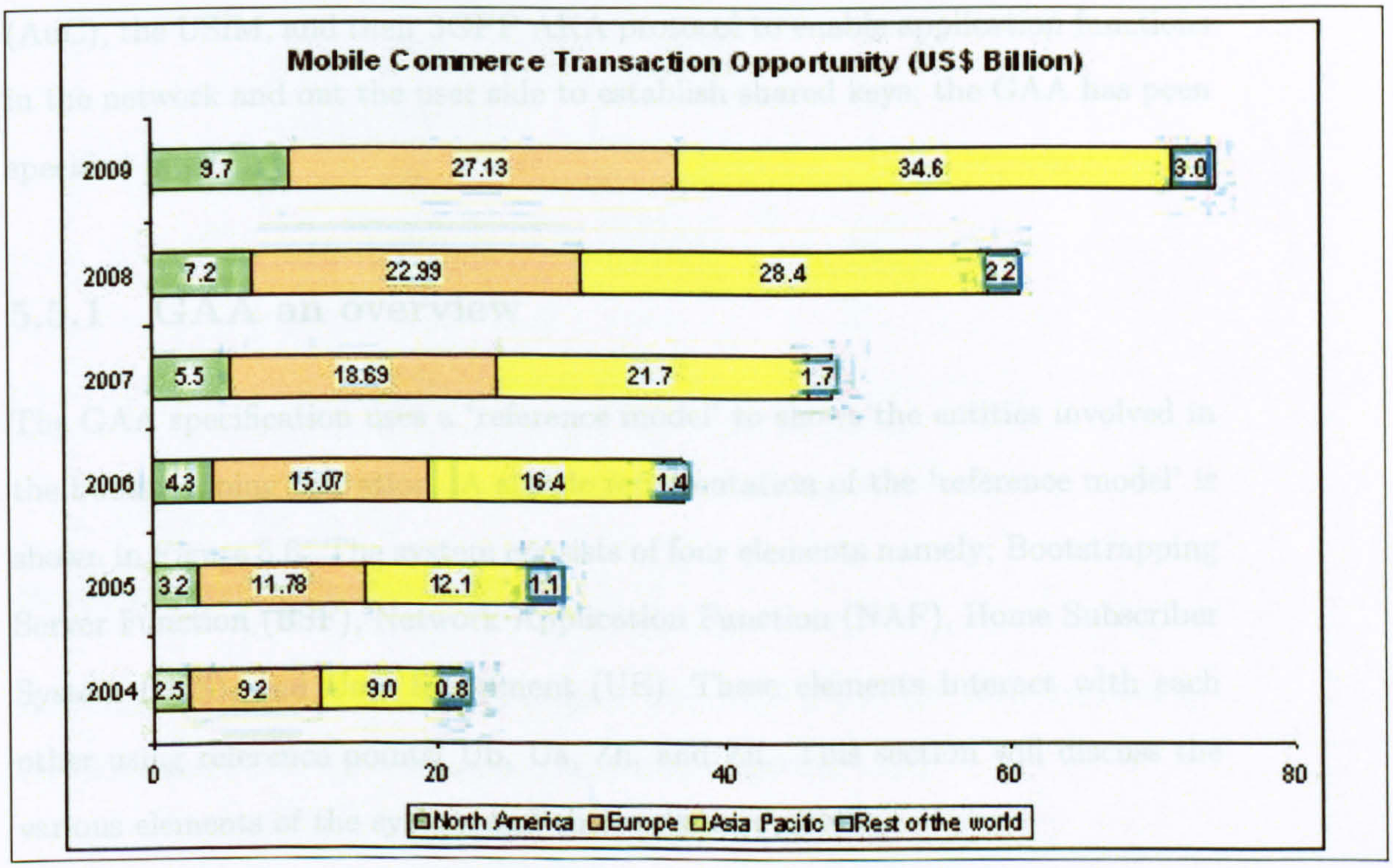

Figure 5.5: Mobile Commerce Transaction Worldwide

5.5 3GPP Generic Authentication Architecture (GAA)

Authenticating users to application servers is a big security issue for most E/M-Commerce system. In the other hand, the GSM authentication infrastructure has proved itself to be a very successful system. Recently the 3rd Generation Partnership Project (3GPP) started working on a framework that enables mobile operators to extend 3G authentication as a service, this framework is known as the Generic Authentication Architecture (GAA). We introduce the GAA in the next section.

The main concept behind GAA is to use the 3GPP Authentication Center

(AuC), the USIM, and their 3GPP AKA protocol to enable application functions in the network and ont the user side to establish shared keys; the GAA has peen specified in [95].

5.5.1 GAA an overview

The GAA specification uses a 'reference model' to shows the entities involved in the bootstrapping operation. A simple representation of the 'reference model' is shown in Figure 5.6. The system consists of four elements namely; Bootstrapping Server Function (BSF), Network Application Function (NAF), Home Subscriber System (HSS), and User Equipment (UE). These elements interact with each other using reference points; Ub, Ua, Zh, and Zn. This section will discuss the various elements of the system and their reference points.

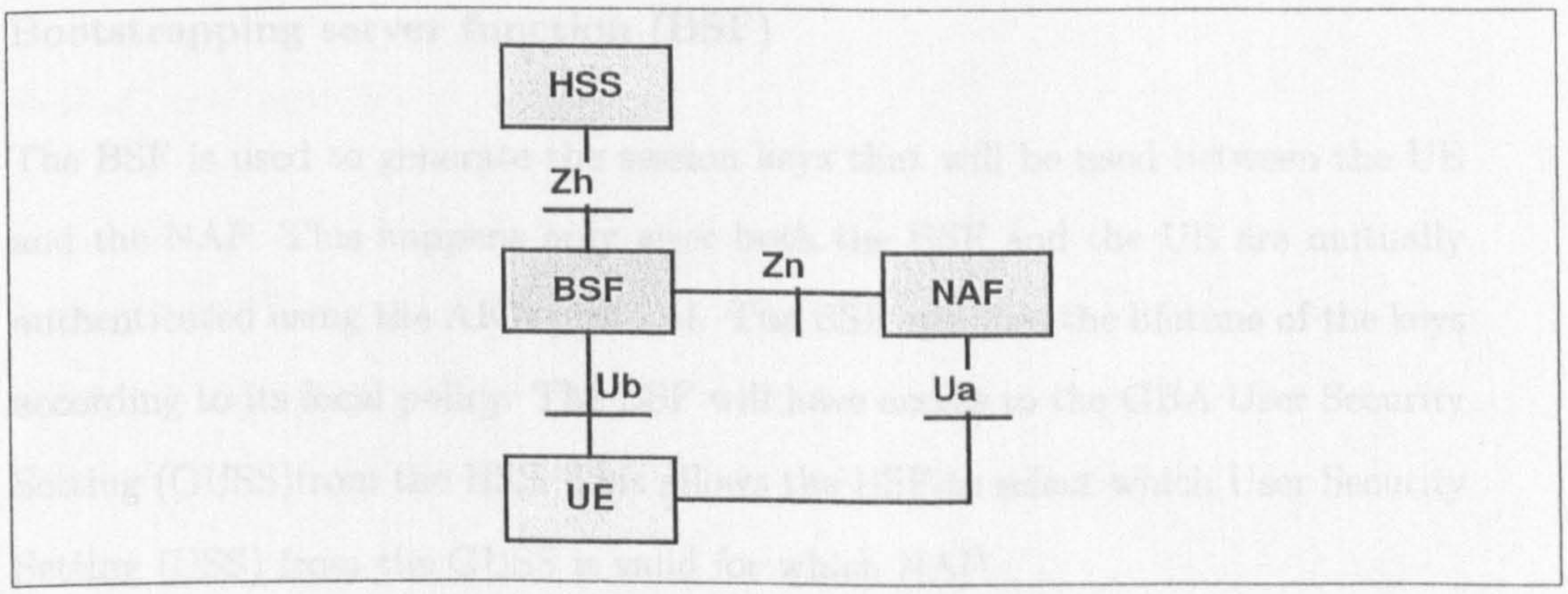

Figure 5.6: Bootstrapping reference model

The NAF can be located at a different network from the BSF, in this case a Diameter Proxy (D-Proxy) element is added to the reference model which acts as a proxy between the visited NAF, and the subscriber "Home" BSF, if this is the reference model will be adjusted as shown in Figure 5.7.

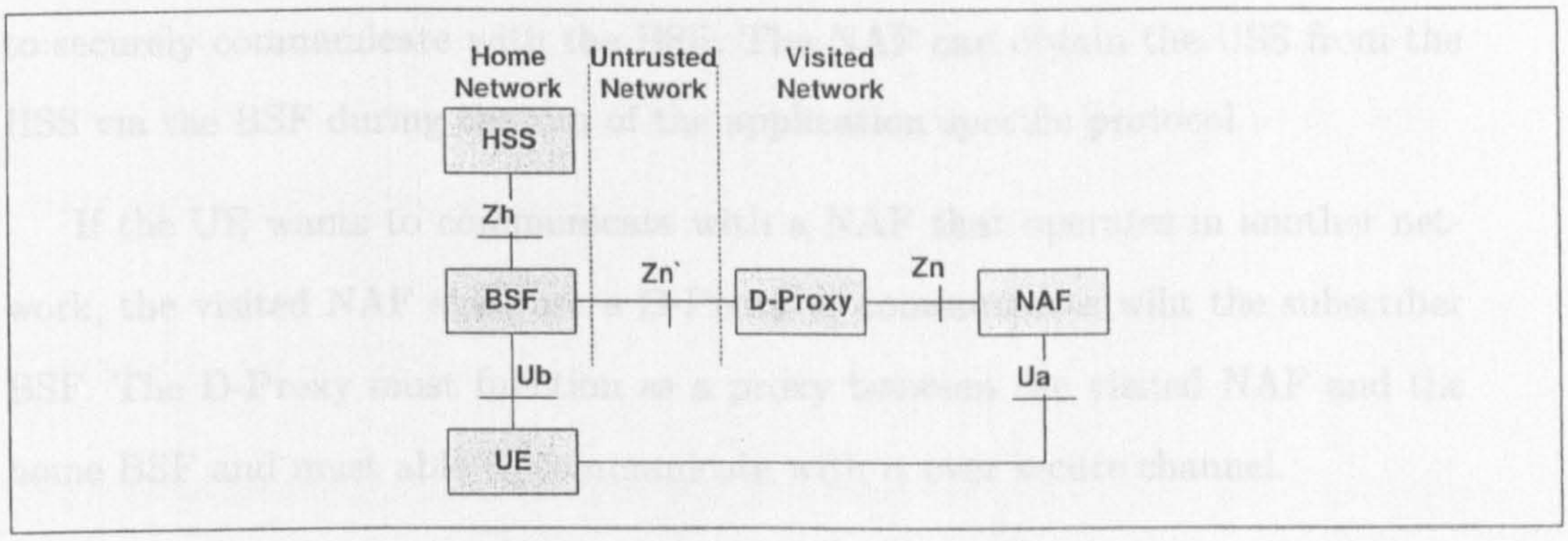

Figure 5.7: Bootstrapping reference model in visited network

Home subscriber system (HSS)

HSS stores all the user security settings (USSs), HSS is the only persistent storage for GUSSs. The GUSS can contain application specific USSs.

Bootstrapping server function (BSF)

The BSF is used to generate the session keys that will be used between the UE and the NAF. This happens only after both the BSF and the UE are mutually authenticated using the AKA protocol. The BSF specifies the lifetime of the keys according to its local policy. The BSF will have access to the GBA User Security Setting (GUSS)from the HSS. This allows the BSF to select which User Security Setting (USS) from the GUSS is valid for which NAF.

Network application function (NAF)

NAF uses the session keys generated during the bootstrapping to communicate with the UE, and to be able to run the application specific protocol. It is assumed that the NAF does not have any security association with the UE, but it is able

to securely communicate with the BSF. The NAF can obtain the USS from the HSS via the BSF during the run of the application specific protocol.

If the UE wants to communicate with a NAF that operates in another network, the visited NAF shall use a D-Proxy to communicate wiht the subscriber BSF. The D-Proxy must function as a proxy between the visited NAF and the home BSF and must able to communicate with it over secure channel.

User Equipment (UE)

The UE must be able to support the HTTP Digest Protocol [41], and should be able to derive the new key material from the Confidentiality Key (CK) and the Integrity Key (IK) to be used with the protocol over the Ua interface. Also, it should be able to support the NAF specific application protocol.

Reference Point Ua

This reference point is used between the UE and the NAF, the application protocol is carried over this reference point. The communication in the Ua is secured using the keys generated between the UE and the BSF. The NAF should be able to indicate to the UE to use newer key if the key has expired.

The Ua reference point can be secured using HTTPS as defined in TS 133.222 [110]. The UE will establish a TLS tunnel with the NAF. The NAF is authenticated to the UE by means of a public key certificate which the user needs to verify (checks that the server certificate corresponds to the FQDN). The UE (i.e. the client in TLS terms) does not need to authenticate itself to the NAF.

Once the tunnel is established the UE will indicate to the NAF that GBA

authentication is supported by adding a constant string "3gpp-gba" to the HTTP header. The rest of the messages used in this process are discussed in the bootstrapping procedure section.

Reference Point Ub

The Ub reference point is between UE and BSF. This reference point provides mutual authentication between the UE and the BSF. It allows the UE to use the 3GPP AKA to bootstrap the session keys.

The Ub reference point specifies some requirements, including; the BSF should be able to identify the UE and that both should be able to authenticate each other based on the AKA; the BSF shall be able to send a bootstrapping transaction identifier to the UE and be able to send (BSF to UE) the expiry time of the key.

Reference Point Zh

The reference pint Zh used between the BSF and the HSS to allow the BSF to fetch the required authentication information and all the Generic Bootstrapping Architecture (GBA) user security settings from the HSS through an internal interface to the 3G AuC.

There are some requirements on the Zh reference point, including; the BSF to initiate all the procedures; mutual authentication, confidentiality and integrity shall be provided; the BSF shall be able to send subscriber bootstrapping information; the HSS shall be able to send 3GPP AKA vectors and the complete set of the subscriber GBA USS to the BSF. Its important to note that if the subscriber GUSS is updated in the HSS, it will not be updated in BSF until the next time

the BSF fetches the authentication vectors and GUSS from HSS.

Reference Point Zn

This reference point is used by the NAF to fetch the key material agreed over the reference point Ub by the UE and the BSF. It can also be used to fetch application specific user security settings from the BSF if requested from the NAF.

The Zn reference point will be secured according to NDS/IP [111] if both the BSF and the NAF are located within the same operator; if they are in different operator the Zn‘ reference point between the D-Proxy and the BSF will be secured using TLS [62, 112]. The NAF will send a key material request to the BSF, containing the NAF's public hostname used by the UE's; the NAF should be able to indicate to the BSF if it requires the USSs for a single application or several applications.

As with the Ub reference point the BSF should be able to indicate the lifetime of the keys, thus indicating the expiry time of the keys. This lifetime that must be identical to the lifetime sent by the BSF to the UE over the Ub.

Bootstrapping Transaction Identifier

The bootstrapping transaction identifier (B-TID) is used to bind the subscriber identity to the keying material in reference points Ua, Ub, and Zn. The B-TID must be globally unique and could be used as a key identifer in protocols used in the reference point Ua. The B-TID should inform the NAF about the home network and the BSF about the UE

5.5.2 Bootstrapping procedure

The following list specifies the format of the bootstrapping procedure. Figure 5.8 shows the overall messages communication during the GAA bootstrapping procedures:

1. The UE will request communication with the NAF without any GBA related parameters.

2. The NAF replies with a bootstrapping initiation message; both messages 1 and 2 will be sent over the reference point Ua.

3. The UE contact the BSF using HTTP request.

4. Using the Zh reference point the BSF retrieves the complete set of GBA user security setting from the HSS, including one Authentication Vector (AV) (where AV= RAND||AUTN||XRES||CK||IK). Where:

   - RAND: Random challenge generated by the AuC.
   - AUTN: Authentication Token, a 128 bit value generated by the AuC and used with RAND to authenticate the server to the client.
   - XRES: Expected authentication response.
   - CK: Cipher Key used for encryption.
   - IK: Integrity Key used for integrity check.

5. To perform authentication the BSF only forwards the RAND and AUTN to the UE.

6. The UE verifies the AUTN, and generates RES and the session keys IK and CK. Using RES the UE calculates the Digest AKA response and sends it to the BSF.

7. The BSF authenticates the UE by verifying the Digest AKA response.

8. The BSF generates Ks by concatenating CK and IK, and generating B-TID which is (base64-encode(RAND@BSF_servers_domain_name).

9. The BSF sends B-TID and the lifetime of Ks to UE using 200 OK message. The UE then generates Ks by concatenating CK and IK.

10. This Ks will be used to generate Ks_NAF, where Ks_NAF= KDF(Ks, "gba-me", RAND, IMPI, NAF_Id; KDF is the key derivation function, and "gba-me" is just a fixed text.

11. The UE sends the B-TID to the NAF.

12. The NAF request from the BSF the UE key material corresponding to B-TID which will be sent in the request along with the NAF's public hostname.

13. The BSF verifies the NAF hostname. If verification successful it will derive the keys required to protect the protocol used over the reference point Ua from the key Ks and the key derivation parameters, and supply the NAF with the requested Ks_NAF and the lifetime of the key.

14. The NAF now can communicate with the UE over Ua securely.

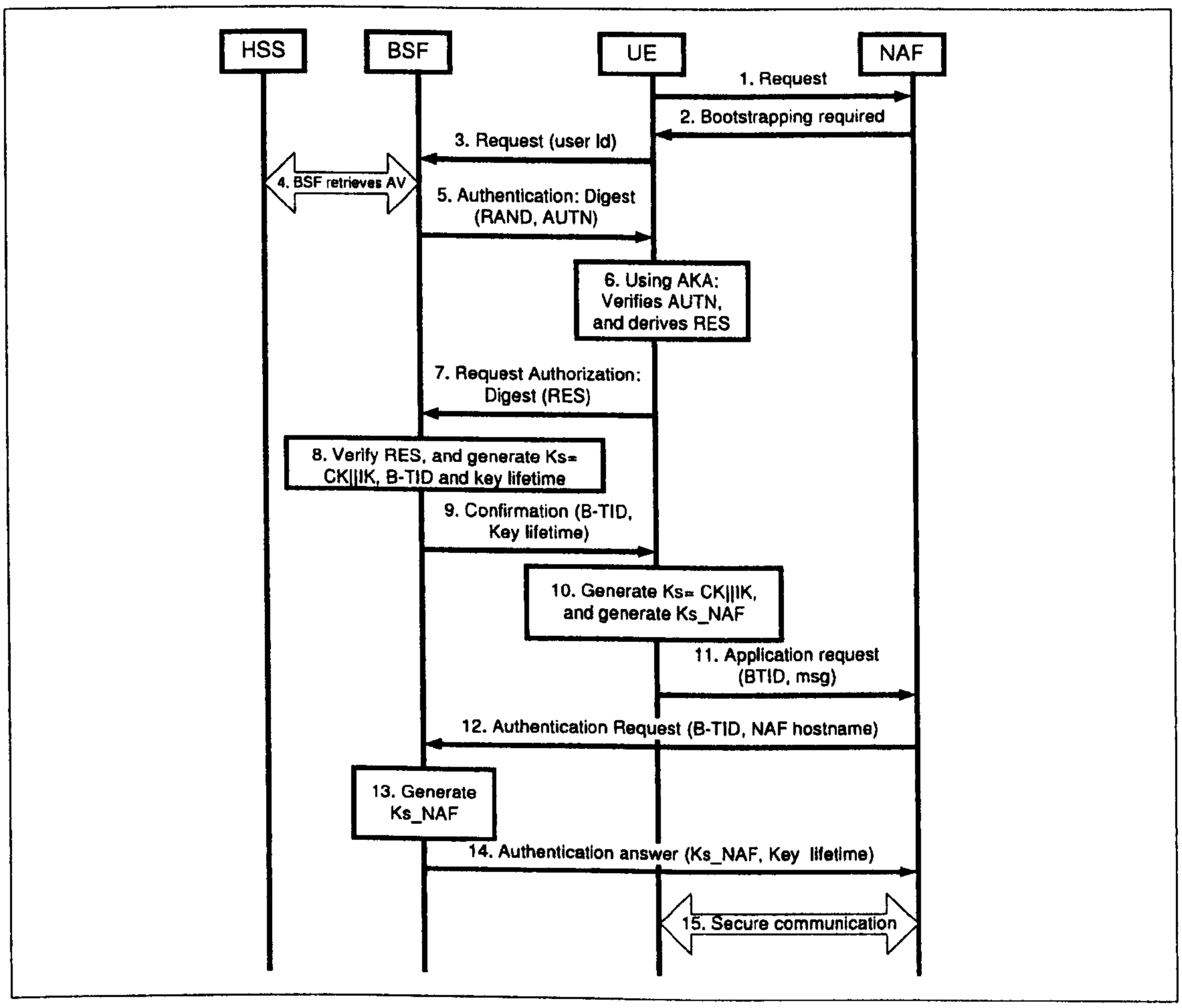

Figure 5.8: Bootstrapping procedure

5.5.3 Bootstrapping procedure with UICC

The TS 133.220 document [95] proposes an enhancements to the bootstrapping protocol by adding an interface between the ME and the UICC. The idea behind this addition is that only the UICC is trusted at the user side and therefore the 3G AKA keys CK and IK shall not leave the UICC.

The bootstrapping procedure only differs in the local handling of keys and Authentication Vectors in the UE and the BSF, all the messages exchanged over

the Ub reference point are identical [95].

5.5.4 Issues not covered by the TS 133.220

This section highlights some important security considerations that are not covered by the TS 133.220.

- The GBA does not guarantee the freshness of the key. It does not guarantee that the key was not used in the previous run of the Ua protocol. It is up to the UE and the NAF to ensure the key freshness in GBA.
- The BSF will not require the UE nor the NAF to refresh the key before the expiry time. This should be handled by the UE's and the NAF's local policy.
- Any updates to the subscriber's GUSS in HSS will not be propagated to BSF or NAF until the next time the BSF fetches these GUSS from HSS or the NAF from BSF.
- The current system allows the BSF to send the private user identity (IMPI) to the NAF. This could effect the user privacy?

5.6 The Generic Mobile Web Service Platform

In this section we propose a generic platform for authentication and payment between a consumer and a Web Service Provider that builds upon the Mobile Operator relationship with the mobile subscriber. The proposed scheme enables

the Mobile Operator to provide a trusted authentication service that allows a third party to implement an environment where Web Service Providers gain direct commercial access to the Mobile Operator's subscriber base for the consumption of digital and physical products.

The technical novelty of the protocol is that it builds on existing industry standards of GBA [95] from 3GPP and web services from Liberty to provide a scalable, intuitive and easy to use authentication solution capable of underpinning the delivery of interoperable web services direct to the mobile consumer. The non-technical novelty of the protocol concerns the potential business models arising from the delivery of interoperable and independent web services to mobile consumers in a scheme that is not operated by the mobile operator.

But why should the Mobile Operator wish to encourage such access? It has long been noted [113] that distribution structures, and specifically the consumer facing retailing function, evolve as industries mature. Many consider traditional Mobile Operators to be at the early stages of their development as retailers of digital content. The current distribution structures typified by Vodafone Live! from Vodafone, T Zones from T-Mobile, and e-mocion from Telefonica are examples of "one stop shops". Vertically integrated, they source, market and advertise a range of goods to consumers who are encouraged to repeat purchase. They may be considered as analogous to a Department Store on the high street. The typical High Street has evolved, however, and in many cases is complemented (if not replaced) by the Shopping Mall. Comprising both Department Stores and specialist retailers the operator of the Shopping Mall benefits from a large number of customers (i.e. traffic volume) whilst remaining independent from the cost and management of the retailed stock. As the commercial benefit from provision

of digital content to mobile consumers transitions from promotional to revenue generating, the "Shopping Mall" concept of digital content retailing may become an attractive model for the traditional Mobile Operator.

5.7 The Web service requirement

The proposal involves four main actors; the Consumer, the Mobile Operator, the IdP (e.g. Shopping Mall Operator) and the Service Provider.

The consumer is assumed to access the scheme via a bandwidth-constrained Mobile Station, comprising mobile device and service-enabling SIM card connected to a GPRS or UMTS mobile network. Service latency should be minimal without the need to purchase new equipment, and the "purchase experience" should be consistent across all services, irrespective of the actual service provider. Payment for services should be through the normal on-phone and off-phone payment mechanisms. Anonymity is an optional consumer requirement. Service consumption is ad hoc, irregular and transitory in duration.

The Shopping Mall Operator is assumed to require the maximum number of consumers for the available services, and the maximum number of available services for the participating consumers. Service Providers and Consumers should be capable of dynamically and asynchronously entering and leaving the system. The service should be available to consumers from various and disparate trust domains and the service must be terminal vendor independent and capable of being set-up using Over The Air (OTA) techniques.

Finally, Mobile Operator and Service Provider entities will not want to develop new business processes solely for specific Shopping Mall Operators. These

entities must interact with the system using standard, internationally agreed protocols.

We base our proposed scheme on the assumption that these requirements are met with a Web Services architecture as described in the functional diagram of Figure 5.9, which describes the interaction of the main scheme actors where:

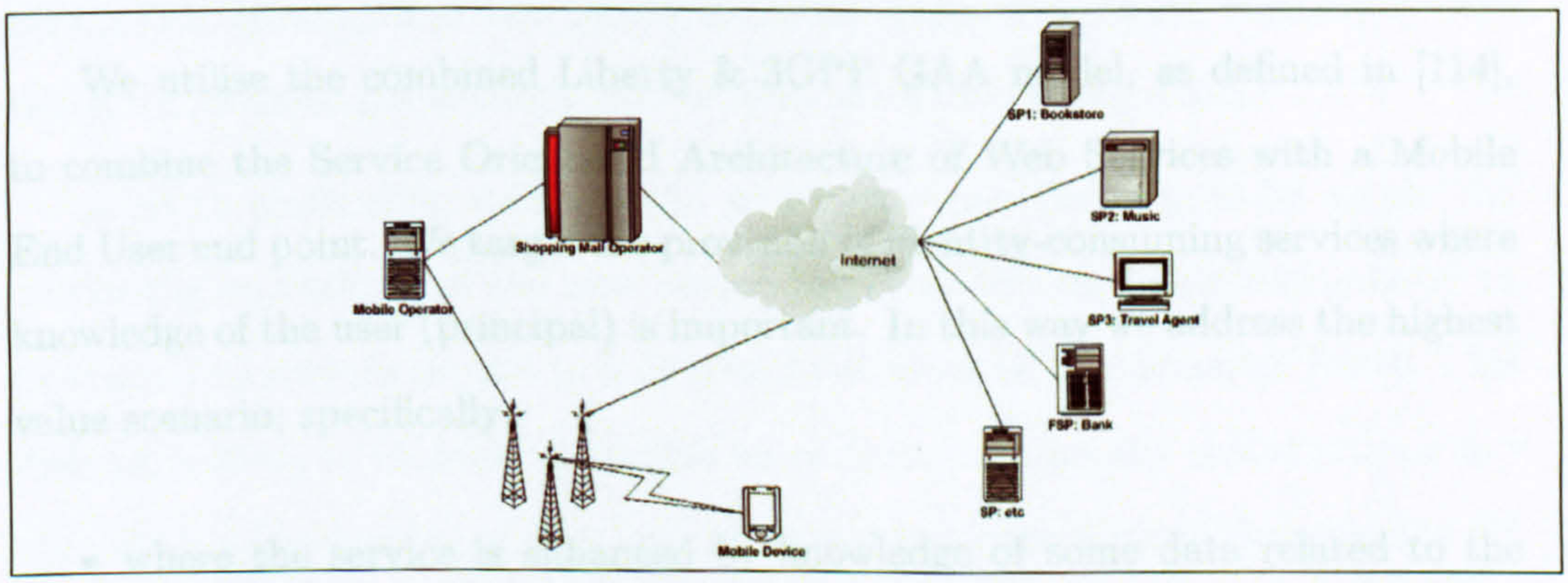

Figure 5.9: Interaction of Scheme Actors

- the consumer service endpoint is an OTA installed application running on a mobile device that uses the SIM card as its security element,
- the service content is provided by a Web Services Provider in accordance with internet standards,
- the Mobile Operator provides the authentication service, and
- the Shopping Mall Operator implements a co-located 3GPP Network Application Function and Liberty-enabled Identity Provider entity.

5.8 The proposed scheme

In our proposal the Shopping Mall Operator acts as an Identity Provider between Web Service Providers and each of the Web Service consumers (Mobile Stations). The Mobile Operator owning the SIM deployed in the Mobile Station, acts as an Authentication Authority to the Shopping Mall Operator.

We utilise the combined Liberty & 3GPP GAA model, as defined in [114], to combine the Service Orientated Architecture of Web Services with a Mobile End User end point. We target the provision of identity-consuming services where knowledge of the user (principal) is important. In this way we address the highest value scenario; specifically:

- where the service is enhanced by knowledge of some data related to the identity of the principal (e.g. payment).
- where privacy, trust and authentication are highly relevant.

We consider a federated environment where it is in the principal's interest to re-use such assertions/validations/vouches for access to unrelated services. Our platform implements a permission based access control where the permission level is a function of the "quality" of the initial assertion. Further, we consider the general case where, although the consumer identity is provided by the Mobile Operator customer owning entity, they are given the freedom to attest for the identity of a particular consumer up to "a certain level". Therefore not all assertions are necessarily considered to be of equal quality.

Securing identity is fundamental for Web services security, and as the identity of valid users must move around when information moves from one trust

domain to another, and the fact that Web services will be used to cross trust domains makes portable trust an important requirement for Web services security. Authentication credentials are defined in SAML vocabularies.

In the GSM/3GPP mobile architecture, security and trust reside in two locations. These are the network HLR (Home Location Register) of the HSS (Home Subscriber System) and the Operator issued tamper resistant SIM card. We therefore consider the network HSS as the customer owning entity.

A client application needs to run within the user device in order to use the processing capabilities of the user device. However this user device is unlikely to be trusted by scheme entities to hold a valuable network level identity. [Note: This distrust is likely to increase as devices move from traditionally closed proprietary operating systems to more open operating systems capable of performing the file manipulation required by advanced 2.5G and 3G services].

The customer owning entity — the network HLR — attests the identity of a particular consumer up to "a certain level". Application layer credentials are bootstrapped from the (3G) cellular network mutual authentication process and provided to both the End User device and the Service Provider. This allows the Service Provider and End User to communicate securely as they now share the same secret.

We use the GBA or Generic Bootstrapping architecture of GAA (Generic Authentication Architecture) as described in [114] to exploit the 3GPP Authentication and Key Agreement process to produce application credentials. The Mobile Station uses the Bootstrapping Server Function of the Mobile Operators Home Subscriber System to create these application layer credentials, i.e. GBA, over the Ub interface. These are then shared with the Identity Provider (IdP

or sometimes referred to as the Network Application Function) via the Zn interface. The Mobile Station client can then communicate directly with the Service Provider using these credentials.

The scheme for a registered user is summarised with reference to Figure 5.10 where the user posses user equipment (UE) comprising mobile equipment (ME) and a UICC SIM card ($UICC$) (User-dependent and user subscription-dependent). The Shopping Mall Operator implements the Network Application Function (NAF/IdP) with consumer web services provided by the Service Provider (SP). The Mobile Operator provides the Boostrapping Server Function and Home Subscriber System (BSF/HSS). The use case for a non-registered user begins when the User attempts to purchase a service from an (SP). The (SP) advises the (NAF/IdP) who determines if User is registered. If not registered, the (NAF/IdP) determines if User has the capability, i.e. the "User Agent", in the form of the MIDP2.0 and Javacard application code. If the User Agent is not present then this application code is OTA (Over The Air) downloaded to the User Equipment in accordance with [115]. The User Agent then registers with the (NAF/IdP) for Single Sign-On service.

1. Once registered, the User Agent of the(UE) performs GBA_U with (BSF) over Ub.

2. The User Agent applet within the UICC is provided with Ub parameters.

3. The UICC component of the User Agent calculates the K_s and provides the ME with the service layer credentials (K_s_(int/ext)_NAF). The K_s always remains in the UICC.

4. The User Agent makes contact with the (NAF/IdP) to obtain a "Shopping

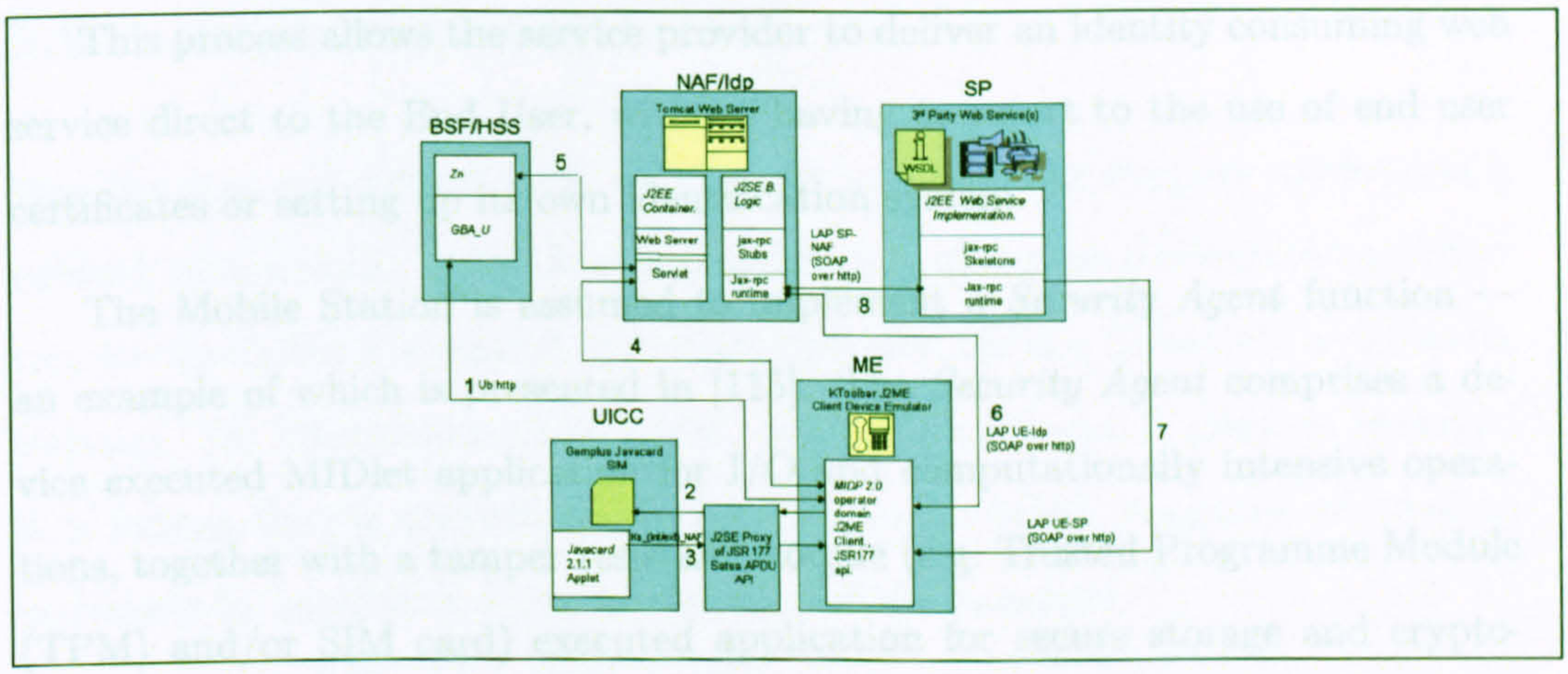

Figure 5.10: Scheme Description

Mall" identity.

5. Service credentials appropriate to the User Agent are communicated via Zn to the (NAF/IdP).

6. A SAML authentication token for the "Shopping Mall" is provided to the User Agent from the (NAF/IdP).

7. (UE) communicates with (SP) using service credentials and requests a service.

8. (SP) confirms validity of (UE)'s service credentials.

The (UE) can now purchase from (SP) using On-Phone billing (i.e. via HSS as the payment gateway) or Off-Phone billing (i.e. via a second Service Provider who performs a payment gateway service). The use case continues with (UE) accessing multiple Service Providers until the session is actively terminated either by the User or the (SP).

This process allows the service provider to deliver an identity consuming web service direct to the End User, without having to resort to the use of end user certificates or setting up its own identification system.

The Mobile Station is assumed to implement a *Security Agent* function — an example of which is presented in [115]. The *Security Agent* comprises a device executed MIDlet application for I/O and computationally intensive operations, together with a tamper-resistant module (e.g. Trusted Programme Module (TPM) and/or SIM card) executed application for secure storage and cryptographic processing. The Shopping Mall Operator is assumed to implement a Token Distribution Centre.

We adopt a push-based model [116] to exchange authentication and payment SAML authorisation tokens between the scheme entities. Tokens are pushed from the Shopping Mall Operator to the Mobile Station, for local storage. This allows a shopping basket of services to be assembled before the tokens are subsequently pushed from the Mobile Station to the Web Service Providers in exchange for their services.

By storing the tokens on the Mobile Station we simulate a familiar shopping behaviour. We allow the consumer to pause (i.e. service interuption) between the phases of entering the Shopping Mall (i.e. authentication), browsing and selecting the goods (web service selection) and proceeding to the checkout (i.e. payment). It is considered good practice [117] to design mobile applications so that they can be interrupted by the user.

5.9 Implementation

Web Services are defined [118] as software systems that support interoperable network interactions. They allow implementation of a service-orientated architecture incorporating the entities of Service Provider, Service Consumer and Service Registry. For information to be moved around the network it must be packaged in a format that is understood by these entities. SOAP supports information exchanges by specifying a way to structure XML messages.

As in any open network environment, these exchanges are exposed to security threats of message leakage, tampering and vandalism. We propose protocol and token implementation options that are designed to resist masquerading, message tampering, replay, and denial of service attacks. Further, as the characteristic of a Web Service is a response to a message, perceived service quality is also dependent on latency between message and response. We therefore also consider the implementation options that affect this.

We present both specific protocol exchanges and the structure and syntax of the authentication and payment tokens.

5.9.1 Prerequisites for protocol

Our protocol uses both symmetric and asymmetric cryptographic techniques to provide the authentication and integrity services required.

The following requirements must be met prior to the use of the protocol.

- All actors have agreed on a specific signature algorithm. The signature on data X using private key K is written $s_K(X)$.

- All actors have agreed on an asymmetric encryption algorithm, for which the encryption of data X using public key P is written $e_P(X)$.

- All actors except the consumer have encryption key pairs for encryption scheme, and all the actors possess a trusted copy of the public key of the other actors.

- All actors except the consumer have asymmetric key pair for a signature scheme, and all the actors possess a trusted copy of the public key of the other actors.

We use asymmetric cryptography to provide the security services between the Shopping Mall Operator and the Web Service Providers. Our scheme assumes that a Web Service Provider may have a transitory relationship with multiple Shopping Mall Operators. In this topology it is best to avoid the necessity of establishing a long term shared secret; we thus adopt an asymmetric cryptographic solution for provision of security services. The Shopping Mall Operator generates an asymmetric key pairs for the NAF/IdP Server and obtains certificates for the public keys from a Certification Authority. Likewise, the Web Service Providers generate key pairs for their Web Service Servers, and obtain certificates for the public keys. The private keys will be used for digitally signing messages. Our protocol is based on the assumption that the Web Service Providers have access to a trusted copy of the public key of the Certification Authority used to sign the Shopping Mall Operator's public key certificates, and vice versa. We also assume that the Shopping Mall Operator's Server certificate is in a format processable by the Web Service Providers, and that the Web Service Providers certificate are in a format processable by the Shopping Mall Operator. It is further assumed that

the Web Service has access to the Shopping Mall Operator's certificate, prior to commencement of the protocol. Adoption of asymmetric cryptography and PKI enables the scheme to easily scale to multiple Web Service Providers for each Shopping Mall Operator and beyond to a multiple Shopping Mall Operator topology.

5.9.2 Protocol

We describe the critical protocol exchanges to address the threat model by considering the authentication, service selection and payment phases of the protocol. Our description assumes that an authenticated key establishment process has taken place between the Mobile Operator and the *Security Agent* of a Mobile Station [115].

We adopt the following additional notation:

$$SP = \text{Service Provider}$$

$$IdP = \text{Identity Provider}$$

$$NAF = \text{Network Application Function}$$

$$BSF = \text{Bootstrapping Server Function}$$

$$User = \text{Mobile phone user}$$

$$WS = \text{Web Service}$$

$$IMPI = \text{IP Multimedia Private Identity}$$

We have divided the protocol into three sections; Authentication, Service

Selection, and Payment, the following subsections will describe each section. Figure 5.11 provides an overview of the protocol message flow.

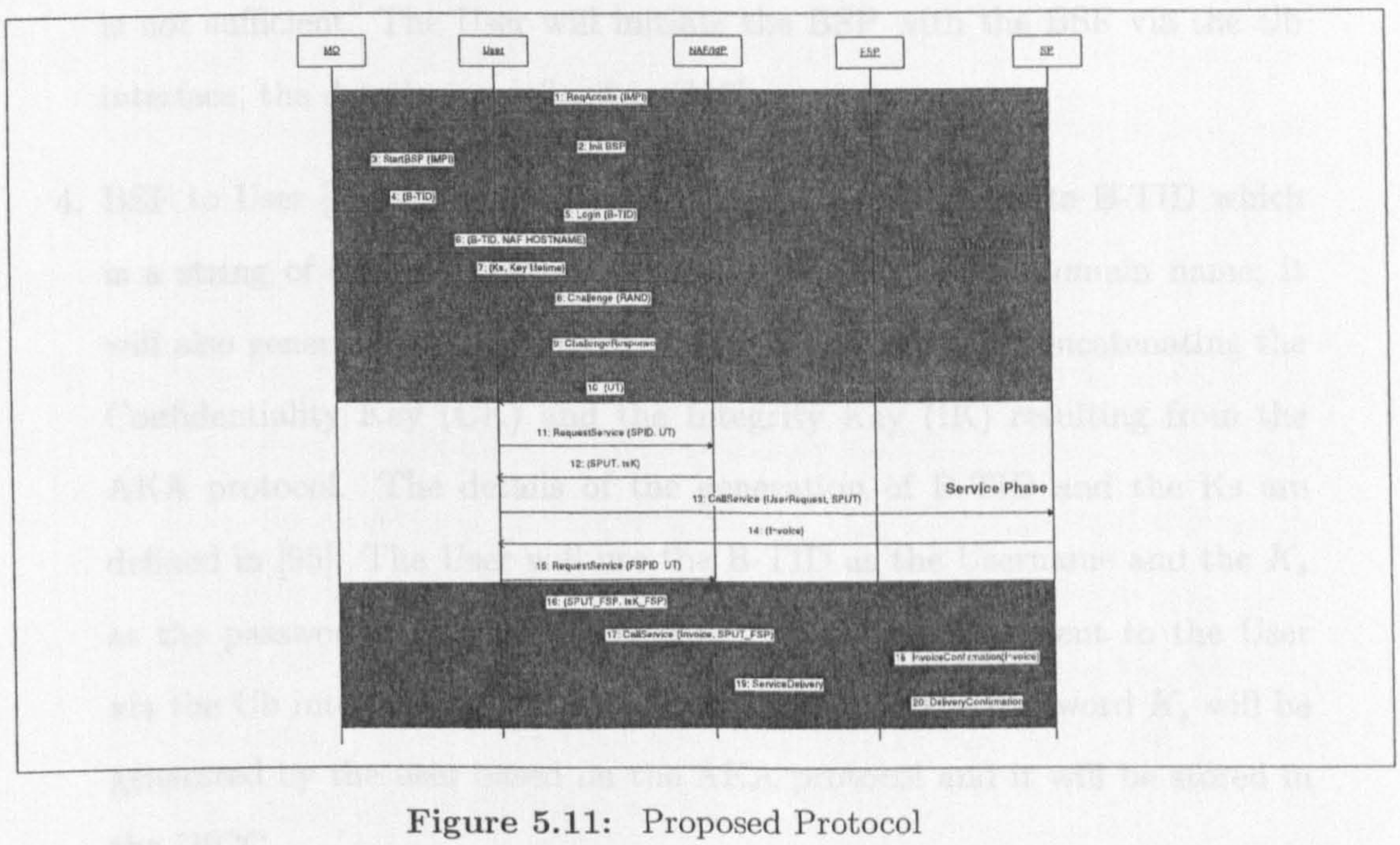

Figure 5.11: Proposed Protocol

Authentication:

The security requirement for this protocol phase is to authenticate the user without user entry of username and password combination or a preloadeed certificate in a manner that avoids the security threat of a man in the middle attack.

1. User to NAF/IdP [ReqAccess(IMPI)]. The User sends a request to access the "Shopping Mall" attaching with the request the User IMPI number.

2. NAF/IdP to User [Init BSP]. Assuming the User has not been authenticated at this stage, the NAF/IdP will send a request to the User to initiate a new Bootstrapping Procedure (BSP).

3. User to BSF [StartBSP(IMPI)]. We assume at this stage that the User does not have a valid bootstrapping session or the freshness of the key material is not sufficient. The User will initiate the BSP with the BSF via the Ub interface, the details are defined in [110].

4. BSF to User [B-TID, Key lifetime]. The BSF will generate B-TID which is a string of base 64 random data and the BSF server domain name; it will also generate key material K_s which is the result of concatenating the Confidentiality Key (CK) and the Integrity Key (IK) resulting from the AKA protocol. The details of the generation of B-TID and the Ks are defined in [95]. The User will use the B-TID as the Username and the K_s as the password to access the NAF/IdP. B-TID will be sent to the User via the Ub interface along with the Key Lifetime, the password K_s will be generated by the user based on the AKA protocol and it will be stored in the UICC.

5. User to NAF/IdP [Login (B-TID)]. The User starts the login procedure by forwarding its 'Username' i.e. the B-TID to the NAF/IdP.

6. NAF/IdP to BSF [B-TID, NAF hostname]. The NAF/IdP needs to obtain the User's password i.e. K_s that belongs to B-TID in order to be able to authenticate the User. This is done by the NAF/IdP sending the B-TID and its NAF hostname to the BSF via Zn interface, the details of this operation are defined in [95].

7. BSF to NAF/IdP [K_s, Key lifetime]. In response to step 6 the BSF will send to the NAF/IdP the User password i.e. K_s and the key lifetime (Note: other related data will be sent in this message, these data were omitted

here for simplicity); the details of this operation are defined in [95] and the security of this message are defined in [110].

8. NAF/IdP to User [Challenge (RAND)]. The NAF/IdP will challenge the User the possession of the password i.e. K_s. This step is required to protect against re-play attack. The NAF/IdP generates a random number RAND and sends it to the User.

9. User to NAF/IdP [ChallengeResponse]. After the User receives the RAND, the User will generate the ChallengeResponse and sends it to the NAF/IdP to prove the possession of the password i.e. K_s. The challengeResponse is a function of the RAND and K_s; ChallengeResponse = f(RAND,K_s); this operation will take place in UICC as K_s will never be reviled to the handset. It is assumed that both the User and the NAF/IdP uses the same function 'f' to generate the ChallengeResponse.

10. NAF/IdP to User [UserToken]. The NAF/IdP needs to verify the ChallengeResponse received in step 9, and if not successful it repeat step 8; if successful it will generate a UserToken (UT) and sends it to the User. The UT will be generated as follows: the IdP part of the NAF/IdP will generate a Temporary User ID (TUID), this will be used to access the SSO system in which the IdP acts as the Authentication Server. The TUID is derived by the IdP from the User ID (UID).

    Note: the NAF IdP mapping is done using a 'User map table', which maps the User's IMPI to the UID (or TUID).

    The UT will be built by concatenating the TUID to a date/time timestamp (TS), and signing the TUID||TS with the IdP digital signature private

key $IdP_{ds:sk}$, and encrypting the result with the IdP encryption public key $IdP_{e:pk}$, such that the UT=$e_{IdP_{e:PK}}(s_{IdP_{ds:SK}}[\text{TUID}||\text{TS}])$.

This UT will be sent to the User encrypted using the password K_s received in step 7.

This protocol phase uses standardised GBA techniques to establish a shared secret between the UE and the Shopping Mall IdP. This concludes the authentication phase, all steps can happen at an earlier time before requesting access to any particular third party service provider, providing the lifetime of the keys have not been exceeded.

Service Selection:

The security requirement for this protocol phase is for the user to select a service without exposing their identity to the Web Service Provider. The proposed protocol for the Service Phase of the Figure described below:

1. User to NAF/IdP [RequestService (SPID, UT)]. Once the User receives the UT he/she can now request access to any service provider (SP) in the Shopping Mall. However, to do that the User must first receive SP UserToken from the NAF/IdP. This is achieved by the RequestService message where the User sends the ID of the requested SP to the NAF/IdP concatenated with the UT.

   The RequestService message will be encrypted with K_s to protect the message confidentiality, the RequestService = e_{K_s}(SPID||UT).

2. NAF/IdP to User [SPUT, tsK]. The NAF/IdP now generates a specific UserToken for the User to be used only with the SP requested by the SPID

from the RequestService message; this UserToken will be referred to as the SPUT. The SPUT is built as follows: first a Temporary Session Key (tsK) is generated by the IdP, this will be concatenated with the SPID and the TUID and a new timestamp TS; these data then will be signed with the IdP digital signature private key $IdP_{ds:SK}$, and encrypted with the SP encryption public key $SP_{e:PK}$, such that the SPUT=$e_{SP_{e:PK}}(s_{IdP_{ds:SK}}[\text{TUID}||\text{SPID}||\text{tsK}||\text{TS}])$. The SPUT will be sent along with tsK to the User in a message encrypted using K_s.

It is the creation of the service provider specific SPUT, from the user token UT generated following successful authentication of the user by the NAF/IdP, that provides the user anonymity towards the SP. This is an important aspect of the proposed scheme.

3. User to SP [CallService(UserRequest, SPUT)]. The User can now talk directly with the SP requesting any services offered by this SP, the CallService message will contain the UserRequest and the SPUT. The UserRequest will be encrypted using the tsK to protect the User privacy.

   Note: it is assumed at this stage that when the User sends this message to the SP that the User is confirming his/her selection, which can be indicated in the UserRequest.

4. SP to User [Invoice]. Once the SP receives the CallService message it decrypts the SPUT using its encryption private key $SP_{e:SK}$, it then verifies the signature of the SPUT, this is done by validating the SPUT using the NAF/IdP signature public key $s_{IdP_{ds:PK}}$, to ensure the integrity of the content of SPUT; if the validation is successful the SP compares the TUID (or UID) to its registered Users database if it exist, this option allows the

SP to give customized services to its customers. Then the SP gets the tsK from SPUT and use it to decrypt the UserRequest; the SP will reply with an 'Invoice', this Invoice will contain a confirmation of the UserRequest, Price, and a method of payment (e.g. Credit Cards only). The Invoice will be signed by the SP digital signature private key and encrypted with tsK.

Invoice= $e_{SP_{tsK}}(s_{SP_{ds:SK}}$

([UserRequest||Price||MethodOfPayment||TS]))

This protocol phase creates a Temporary Session Key (tsK) to provide a confidentiality service that ensures the users anonymity and protects their privacy with respect to the Web Service Provider.

Payment:

The security requirement for this protocol phase is to perform a confidential payment without a risk of "replay" attacks to any participating financial services operation. The payment vehicle could be either on-phone or off-phone billing.

The proposed protocol for the Payment Phase of the Figure described below:

1. User to NAF/IdP [RequestService (FSPID, UT)]. The User verifies the invoice by decrypting it using tsK, and verifies the content of it. If the Invoice verification process is successful, the User now starts the payment phase. It is assumed that the User has an account with a Financial Service Provider (FSP), who will charge the User and pay the SP. However for the User to communicate with the FSP the User must obtain a SPUT for this FSP; this is done the same way as in steps 11, 12, and changing the SPID with the FSPID.

2. NAF/IdP to User [SPUT_FSP, tsK_FSP], SPUT_FSP=$e_{FSP_{e:PK}}$ ($s_{IdP_{ds:SK}}$[TUID||FSPID||tsK_FSP||TS]).

3. User to FSP (CallService[Invoice, SPUT_FSP]). The User forward the Invoice and the SPUT_FSP to the FSP in a CallService message, to indicate the User confirmation for the FSP to charge the User and pay the SP indicated in the Invoice.

4. FSP to SP [InvoiceConfirmation(Invoice)]. Similar to protocol message 14 of Figure 3, the FSP will decrypt and validate the signature of SPUT_FSP (received in message 17) to obtain the tsK_FSP which will be used to decrypt the Invoice, which if successful will indicate the User confirmation to process the Invoice.

   The FSP then charge the Users with the amount stated in the Invoice, and generate an InvoiceConfirmation, which is the Invoice concatenated with the FSPID and a status flag to indicate the statues of the charging, which can only be True (successful operation) or False (unsuccessful operation). The InvoiceConfirmation message will be signed with the FSP signature private key $s_{FSPd_{s:SK}}$ to protect the integrity of the message and to act as a proof of payment. InvoiceConfirmation= $s_{FSPd_{s:SK}}$(Invoice||FSPID||StatusFlag)

5. SP to User [Service Delivery], once the SP receives the InvoiceConfirmation message from the FSP, it validates the message signature and then checks the StatusFlag, which if set to True, the SP will deliver the service to the User; an optional message can be sent to the FSP to confirm service delivery.

This protocol phase uses an IdP signed timestamp TS and Web Service

Provider identifier SPID to protect against replay attacks. User choice of Financial Service Provider is ensured by the service provider specific user token SPUT, with the Temporary Session Key tsK providing the confidentiality service..

As mentioned above the scheme supports both On-Phone and Off-Phone payment mechanisms. The protocol depicted in Figure 3 and described in detail above is for the Off-Phone payment mechanism. The On-Phone payment mechanism refers to the case when the user uses the Mobile Operator as a FSP by charging the user's phone bills. The payment protocol will be exactly as in the Off-Phone case, with the main difference that the FSP will be the MO — the entity that contains the BSF.

5.9.3 Authentication & payment tokens

Our platform creates a collaborative commercial environment. Central to this is the notion of portable trust, i.e. identity credentials issued in one domain being accepted as proof of the subject's claimed identity (authentication) in another. There exists, therefore many parallel authentication processes. Section 3 and subsequent explanations describe the process adopted for one of them, namely GBA leveraging the 3GPP mobile cellular credentials. To cater for this generic requirement we implement the scheme tokens (e.g. UT) as SAML objects, whose quality rating is based on the value of the attestation that the authentication domain gave the subject. The SAML object, or token, UT is therefore an authentication assertion of the subject (single domain entity), that has been accepted by the NAF/IdP for use within the collaborative commercial environment of the Controlled Shopping Mall (CSM). By issuing a UT to the subject, the subject is now considered a principal (Liberty terminology) within the CSM. The UT is, in

essence, the portable identity authentication assertion of the subject. To provide the quality metric, the UT is signed by the NAF/IdP in a way that is appropriate for the attesting authority. This quality metric is a very important element of our platform as it allows many diverse SP's to decide how much to "trust" the principal.

The following is a list of the various tokens deployed in "The Proposed Scheme":

- UserToken

  (UT=$e_{IdP_{e:PK}}(s_{IdP_{ds:SK}}$[UID||TS])); used by the NAF/IdP only to identify the user in the "Shopping Mall".

- SP UserToken

  (SPUT=$e_{SP_{e:PK}}(s_{IdP_{ds:SK}}$[TUID||SPID||tsK||TS])); user identifier that is unique for every SP inside the "Shopping Mall".

- Invoice

  $e_{SP_{tsK}}(s_{SP_{ds:SK}}$

  ([UserRequest||Price||MethodOfPayment||TS])); is the payment token.

- InvoiceConfirmation

  $s_{FSPd_{s:SK}}$(Invoice||FSPID||StatusFlag); used as proof of payment.

These tokens are implemented as XML objects, as detailed below:

- UserToken

```
<UserToken>
    <UID>String</UID>
    <TimeStamp>Timestamp</TimeStamp>
</UserToken>
```

- SP UserToken

```
<SPUserToken>
    <TempUID>String</TempUID>
    <SPID>String</SPID>
    <TempSessionKey>Key</TempSessionKey>
    <TimeStamp>Timestamp</TimeStamp>
</SPUserToken>
```

- Invoice

```
<Invoice>
    <InvoiceNumber>String<InvoicecNumber>
    <UserRequest>
        <Item>String</Item>
        <Quantity>int</Quantity>
    </UserRequest>
    <Price>double</Price>
    <TimeStamp>Timestamp</TimeStamp>
</Invoice>
```

- InvoiceConfirmation

```
<InvoiceConfirmation>
    <Invoice>
        <InvoiceNumber>String<InvoicecNumber>
        <UserRequest>
            <Item>String</Item>
            <Quantity>int</Quantity>
        </UserRequest>
        <Price>double</Price>
        <TimeStamp>String</TimeStamp>
    </Invoice>
    <FinacialSP>String<FinacialSP>
    <Status>boolean<Status>
</InvoiceConfirmation>
```

These XML objects are incorporated in the SOAP messages that exchange information between scheme actors. Example SOAP messages arising from the scheme are presented in Appendix A.

5.9.4 Proof of concept prototype

To validate our proposal we have constructed a Proof of Concept model, based on the readily available open source tools:

- BSF, NAF_IdP, SP and FSP are deployed as Web Services in Axis (Apache Extensible Interaction System). Axis is a SOAP processor that has been developed as an Apache open source project. Apache Axis 1.3 is deployed on top of Jakarta Tomcat application server and the above Services are deployed in the Apache Axis 1.3. Those services are implemented in J2EE environment.

- A J2ME Client performs the Mobile End User function and is emulated by the Wireless KToolbar [119] from Sun Microsystems, running our *Security Agent* MIDP 2.0 MIDlet on the reference J2ME implementation. The SIM card *Security Agent* function is provided by the JCOPS suite of tools for Java Card applet development.

- Communication between Web Services as well as Web Services and Mobile client has been developed using SOAP messages over http. For authentication SAML tokens were used and are added to the SOAP messages. Axis client is also included in some of the Web Services to invoke services in another Web Service. WSDL document for each web service is created by the Axis Engine.

- According to the protocol, communication between all the entities are secured using java.security and javax.crypto libraries. SOAP messages are signed by XML signature to ensure message integrity. VeriSign's Trust Services Integration Kit is used generate XML signatures.

The demonstration environment of the proof of concept model is implemented in J2ME and J2EE. J2ME provides the necessary Mobile device simulation and J2EE provide the web service implementation and deployment. The model is designed so that each phase of a specific use case is initiated manually and monitored by visual feedback through the use of J2ME mobile simulator. The full code of the system is presented in Appendix 1.

From this Proof of Concept we have identified the following technical and implementation bottlenecks to the acceptance of the scheme:

1. The mobile operator must be considered as a trusted third party. The derived GAA credentials used to secure the application are always known to the mobile operator's BSF.

2. GBA requires a GBA Bootstrapping Client within the ME for operation. This client is a native software object within the ME; it is neither implemented by, nor under the control of, the Shopping Mall Operator. For added security against tampering the UICC SIM must also GBA aware. All participating UEs must be GBA compliant.

3. The BSF is a network operator service. The Mobile Operators of all participating users must implement a compliant BSF service.

5.10 Evaluation

In this section we evaluate the novelty and originality of the protocol and include a security analysis section which identifies some possible security threats.

The main novelty of the protocol can be summarized as follows:

- The way in which the SIM been utilized, from the way we install the user agent OTA, and the storage of the security tokens. This is an improvement over other schemes, e.g. [78, 79], where it is not clear how to get the SIM to behave as required and the ability to gain access to the SIM. In [66]the author proposes an end-to-end application-layer security solution for wireless enterprise applications using the J2ME. The proposed solution uses pure Java components to provide end-to-end client authentication and data confidentiality and integrity between wireless J2ME-based clients and J2EE-based servers. However, the proposal does not make use of the SIM and it places great trust on the mobile device which is an entrusted environment. Therefore we believe we add significant improvement by utilizing the SIM to handle the sensitive security credentials.

- The way the "smart" tokens allowed to establish a relation between the NAF environment (i.e. the mobile environment) and the IdP environment (i.e. the SSO environment).

- How the system allows the user to switch on and off from the anonymous mode. By linking (or breaking the link) the user account in the relation between the NAF and the IdP. In [77] it is only possible to use one mode of operation, and in [47] the author makes intensive use of certificates at the client side (i.e. user), which is an issue in mobile environment.

- Because of the architecture of the system, payments is just another service and financial service providers (including the mobile operator)can be treated as just another service provider in the system.

5.10.1 System Simulation

To further demonstrate and evaluate the solution, a proof-of-concept demonstration has been developed. This section will present the operation of the demonstration. In this scenario we have built a prototype for an M-Commerce case study, that simulate a virtual shopping mall. The user will enter the shopping mall using an on-device main page, from which he can choose his favorite store. Once inside the store the user can then browse for goods/services on offer. After that the user is presented with a payment option (e.g. paying using their phone bills or via bank transfer). Once the payment is confirmed the user will receive the good/service they requested. Figure 5.12 provides the welcoming screen of the Shopping Mall.

Figure 5.13 demonstrate the purchasing process where the buyer select a store within the shopping mall.

The conformation of the purchase is shown in Figure 5.14, where the user is presented with an invoice to confirm the purchase.

The selection of the payment method is presented in Figure 5.15. In this case the user is presented with two options: paying through the user's mobile operator; or using a bank account.

In Figure 5.16 the user receives the video clip he purchased.

Both Figure 5.17 and Figure 5.18 give examples of the security tokens of the

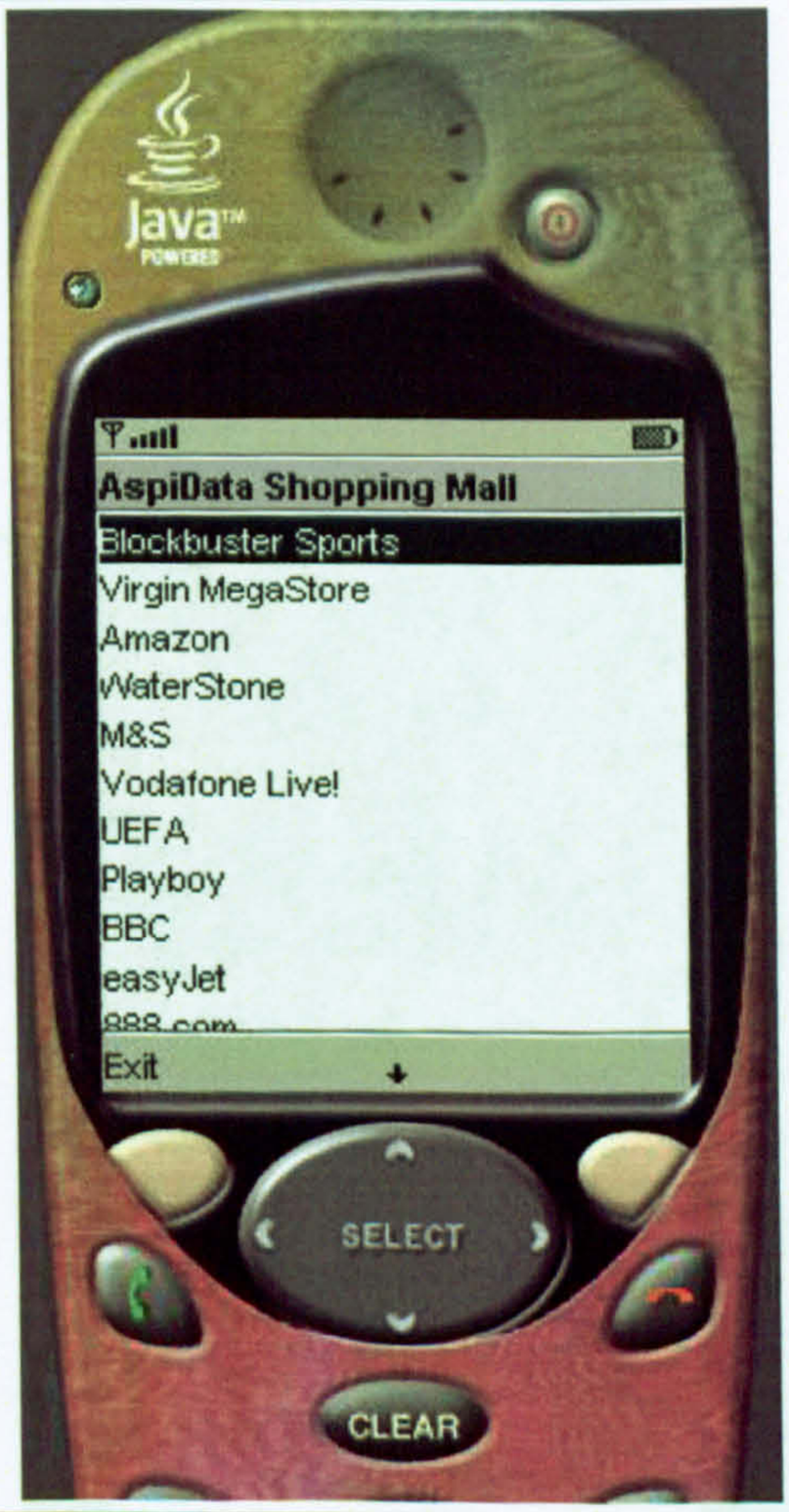

Figure 5.12: ShoppingMall overview

system both in the mobile and the shopping mall environment.

Figure 5.13: purchasing process

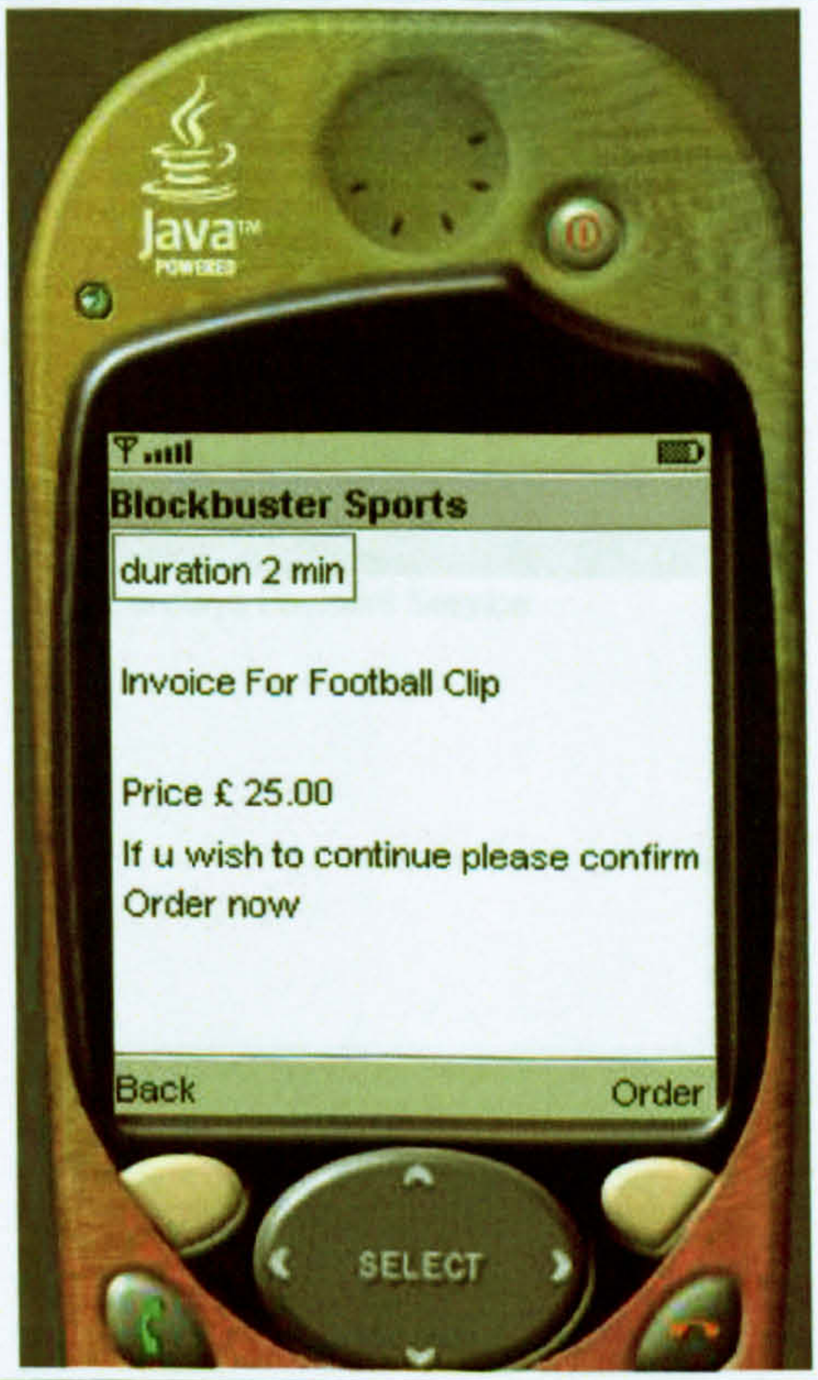

Figure 5.14: Conformation and Invoice

Figure 5.15: Selecting a payment method

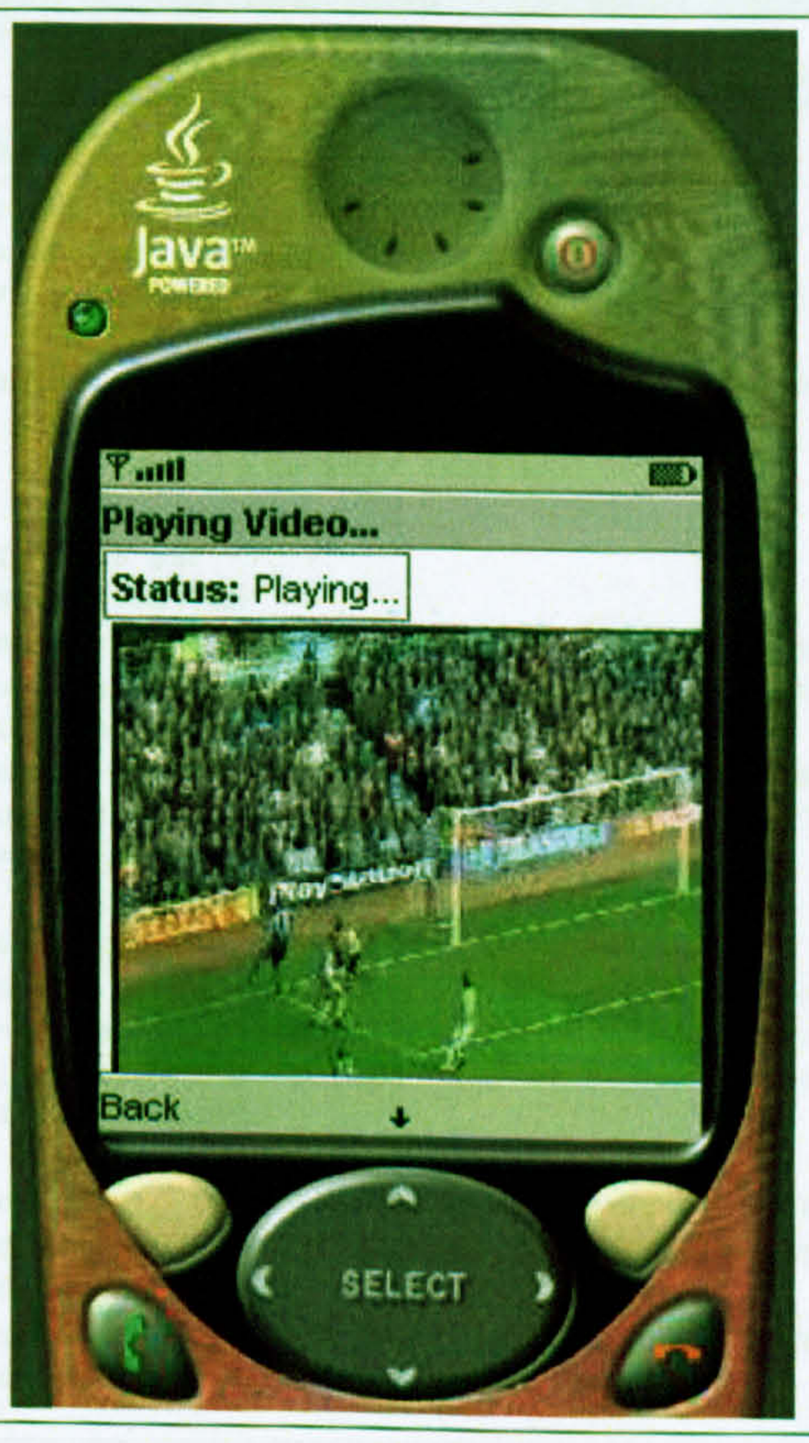

Figure 5.16: Receiving the order

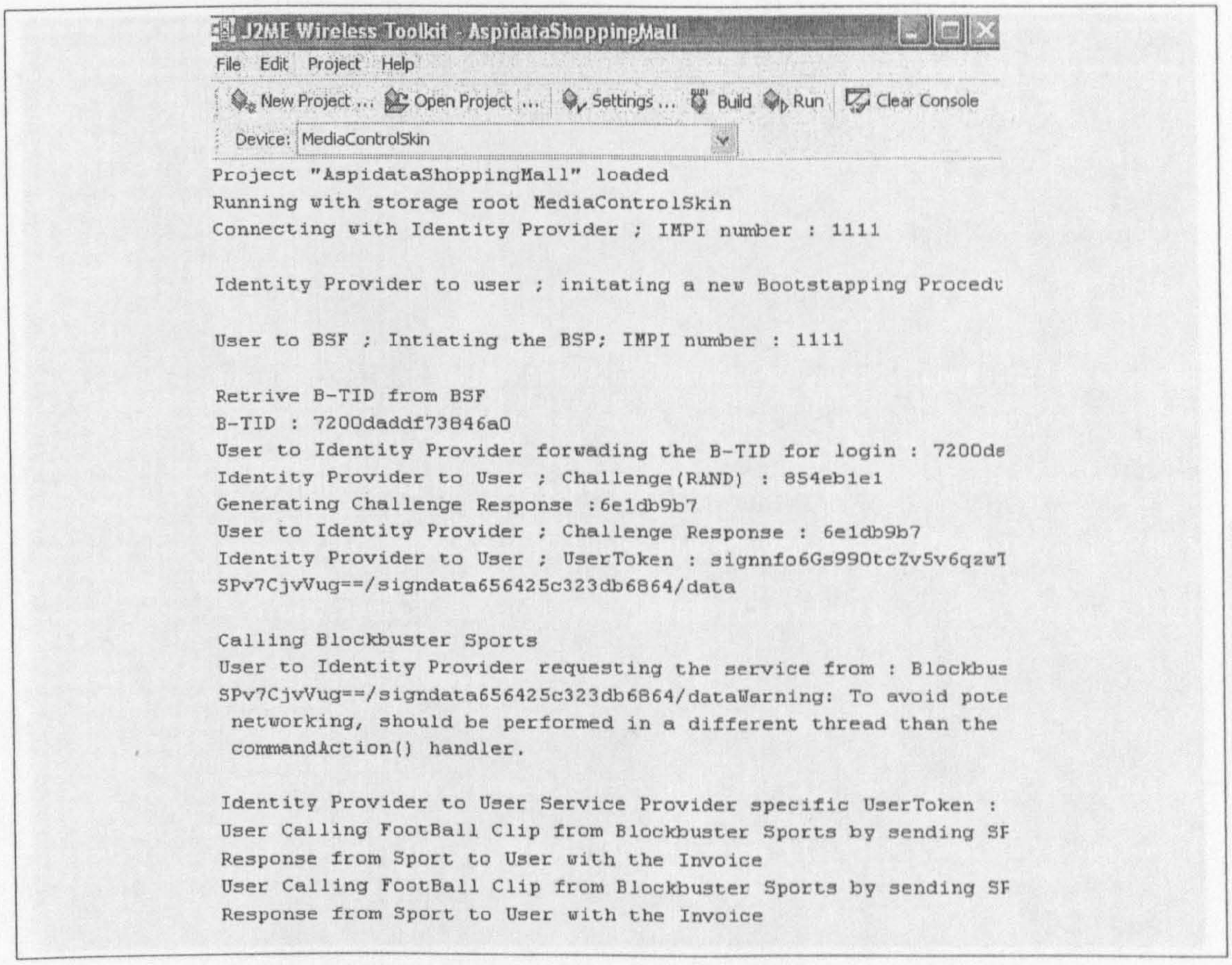

Figure 5.17: Mobile environment security tokens

```
**********1**********
DB connection ssuccessful
**********2**********
insert into naf.user_session (session_rad, tid, sk, impi, logintime) values ('85
4eb1e1','7200daddf73846a0','YYYYYYYYYYYYYY','1111','29-Apr-2003 00:07:55')
DB connection ssuccessful
SELECT sk, tid, ski FROM mo.user_session, mo.user where mo.user_session.impi = '
1111' and mo.user.impi = mo.user_session.impi and mo.user_session.status = 1
DB connection ssuccessful
select session_rad, sk, impi from naf.user_session where tid = '7200daddf73846a0
' and status = 1
DB connection ssuccessful
select session_rad, sk, impi from naf.user_session where tid = '7200daddf73846a0
' and status = 1
6e1db9b7
DB connection ssuccessful
update naf.user_session set tuid= '656425c323db6864' ,ut ='signnfo6Gs990tcZv5v6q
zwTVVPusWbYzPRz9TJjEXdgDTyjwc7UdkAforlfce4Ucbd9FNKCFtjX2FsF
SPv7CjvVug==/signdata656425c323db6864/data', status =2 where session_rad = '854e
b1e1' and tid = '7200daddf73846a0' and impi = '1111'
**** Calling ReqService
*** user Token signnfo6Gs990tcZv5v6qzwTVVPusWbYzPRz9TJjEXdgDTyjwc7UdkAforlfce4Uc
bd9FNKCFtjX2FsF
SPv7CjvVug==/signdata656425c323db6864/data
***Plan Text :signnfo6Gs990tcZv5v6qzwTVVPusWbYzPRz9TJjEXdgDTyjwc7UdkAforlfce4Ucb
d9FNKCFtjX2FsF
SPv7CjvVug==/signdata656425c323db6864/data
***** Sign---->nfo6Gs990tcZv5v6qzwTVVPusWbYzPRz9TJjEXdgDTyjwc7UdkAforlfce4Ucbd9F
NKCFtjX2FsF
SPv7CjvVug==
******* Text---->656425c323db6864
DB connection ssuccessful
select session_rad, sk, impi from naf.user_session where tuid = '656425c323db686
4' and status = 2
DB connection ssuccessful
SELECT sp_u_id FROM naf.user_account where impi ='1111' and sp_id = 'Blockbuster
 Sports'
DB connection ssuccessful
insert into naf.user_session_sp (impi, sp_id, u_sp_id, timestamp, tuid) values (
'1111','Blockbuster Sports','-1','29-Apr-2003 00:09:56','656425c323db6864')
```

Figure 5.18: Shopping mall security tokens

5.10.2 Security Analysis

Security analysis of communication protocols is not a straightforward process and it very much depends of the objectives required. The threat analysis presented in this section is based on the security analysis principles presented in [10, 94].

The centralization issue is a major security issue with NAF/IdP (i.e the shopping mall server). The IdP section is based on an SSO system, which could be a single point of failure. If the IdP fails, the whole system will fail. However it is assumed that the IdP will be held by a trusted third party which will ensure the overall protection of the system.

Timing attack: the aim of this attack is to reveal the user identity or to understand user behaviours by monitoring the network traffic. This attack can be serious if not many users are using the system or the IdP doesn't change the "User Identity" regularly. An eavesdropper who is monitoring the network traffic can conclude with some probability that this User is trying to access a particular SP. More precise solution to this is to increase the time intervals between receiving and sending another message at the User and/or IdP side.

Attack on the communication links: a network eavesdropper who captures any of the communication messages in the system will not be able to gain access to the information about the user as all the user related data are encrypted.

Attack on the MS: all the security properties of the MS are stored in the SIM. If these properties can be extracted from the SIM, either by stealing or cloning the SIM, the attacker can claim to be the subscriber to the BSF. The user will be at risk until the SIM will be blocked.

Attack on the Service Provider: if an attacker manages to get access to the

SP system or even the SP itself tries to gain information about the user it should not be able to do so, because the user is known to the SP by a random ID with no link to user true identity.

Attack on communication links: an attacker may not be able to read the messages sent in the system, but it is still possible that he/she can modify the network traffic. Therefore integrity protection is fundamental to the operation of such a system. This problem can be overcome with the use of some established techniques such as SSL/TLS with client/server certificates [17].

5.11 Summary

SAML is used in Web Service to address the issue of portable identity, and very likely to be adopted it in Mobile Web service. Care must be taken when designing a SAML protocol for Mobile Web service, to balance the requirements between protocol security and efficiency.

When designing an authentication protocol with SAML for Mobile Web service, the network architecture must be taken into account to achieve the required balance between security and efficiency.

This chapter introduced a generic platform for the direct consumption of Web services by a Mobile Station; we described how the Liberty Alliance ID-FF model can make use of the extended authentication services offered by 3GPP GAA to provide a secure infrastructure environment to mobile phone users and Service Providers.

The main contributions of system architecture, protocols, and enabling data structures could form the basis to provides:

1. the user with a high level service discovery interface plus anonymity from Web Service Providers;

2. the Mobile Operator with a pivotal role and a revenue generating opportunity in the provision of a web services security and payment platform;

3. the Service Provider with a secure, scalable distribution channel.

The proposed architecture allows developers and researchers to rethink current distribution structures and business models for the sourcing and delivery of digital services to mobile subscribers.

More an more services are adopting Web service for their solution to benefit from the flexibility and other advantages that Web services offer. And with the service now available on mobile devices such as mobile phons and PDAs where users keep personal information; in addition to the huge number of service interactions, user privacy is of critical important to be protected and managed efficiently.

Chapter 6

Privacy in Mobile Web Services

6.1 Introduction

Web services are increasingly being adopted as a viable means to access Web-based applications. This has been enabled by the tremendous standardization effort to describe, advertise, discover, and invoke web services. On the other hand,government agencies and other industrial entities collect, store, process, and share information about millions of people who have different preferences regarding their privacy. This naturally raises a number of legal and technical issues that must be addressed to preserve people privacy. The issue of privacy becomes even more important when addressing Web services in mobile environment. People nowadays keep all sort of personal and work related data in their mobile phones and PDAs, in addition to the growing number of services that use location based services which could effect the privacy of the user if not managed properly. This section first looks at the issue of privacy then it looks at its effect on web services.

6.1.1 Privacy overview

Before looking at privacy from a technical point of view it helps to define privacy in more general terms. People often think of privacy as some kind of right. The concept of a 'right' is not correct, because a right seems to be an absolute standard. In addition, there is a difference between legal rights, on the one hand, and natural or moral rights, on the other.

Privacy is the interest that individuals have in sustaining a 'personal space', free from interference by other people and organizations [120]. Also privacy is not a single interest, but rather has several dimensions:

- Privacy of the person: This is concerned with the integrity of the individual's body.
- Privacy of personal behaviour: This relates to all aspects of behaviour, but especially to sensitive matters, such as habits and political activities, both in private and in public places.
- Privacy of personal communications: Individuals claim an interest in being able to communicate among themselves, using various media, without routine monitoring of their communications by other persons or organizations.
- Privacy of personal data: Individuals claim that data about themselves should not be automatically available to other individuals and organizations, and that, even where data is possessed by another party, the individual must be able to exercise a substantial degree of control over that data and its use. This is sometimes referred to as 'data privacy' and 'information privacy'.

6.1 Introduction

The problem of privacy protection is that it has to be balanced against many other, often competing, interests.

The privacy interests of one person or category of people may conflict with some other interest of their own, and the two may have to be traded off (e.g. privacy against access to credit, or quality of health care); and the privacy interest of one person or category of people may conflict with other interests of another person, category of people, organization, or society as a whole (e.g. creditors, an insurer, and protection of the public against serious diseases).

Finding the balance between computer security and privacy is not always easy. In order to protect user's privacy, it is best not to give any information, however, in many cases computer users often asked to give some personal/private data such as in system authentication to gain access to a system or application.

A common misuse of the term 'privacy', particularly by security specialists and computer scientists, is to refer to the security of data against various risks, such as the risks of data being accessed or modified by unauthorized persons. In some cases, it is used even more restrictively, to refer only to the security of data during transmission. These aspects are only a small fraction of the considerations within the field of 'information privacy'. More appropriate terms to use for those concepts are 'data security' and 'data transmission security'.

The term 'confidentiality' is also sometimes used by computer scientists to refer to 'data transmission security', risking confusion with obligations under the law of confidence.

The following are common privacy and security problems exist in a typical computer network:

- Lack of privacy: Transmissions that can be sent over a network can be intercepted.

- No proof of sender: It is possible for people to transmit data, and pretend to send it as someone else.

- Data integrity: In a default setup, there is no way to check to make sure that data was not altered in transmission.

- Non-repudiation: In a default setup, there is no way to prove that a user really did send a given transmission (eg, a user can easily claim "I did not send that email - it must have been someone else!")

Because of the nature of WS where one entity will interact with many other entities at different level of security and trust; it is important to 'manage' or 'protect' privacy of the various entities in the system. This section looks at current standards/architectures that address privacy and its possible use in web services.

6.1.2 The Platform for Privacy Preferences

The Platform for Privacy Preferences Project (P3P) is a W3C framework for web privacy management (www.w3.org/P3P/). It enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents [121]. P3P uses XML policy files to describe a web sites privacy practices. These policies describe who collects what data and for what purpose.

The idea behind the system is that sites create P3P files describing their privacy policies, which can easily be read by browser software. Visitors set their own privacy preferences on their browsers, and if those preferences and the policies of visited sites match up, everybody is happy and the system stays in the background. If they don't match, the browser can point this out and ask the user what to do. It can also prevent actions that would otherwise happen automatically, for example it can block cookies if the user's preferences don't match the site's cookie policy.

P3P 1.0 has mainly five goals as defined by the standard [121]:

- A standard schema for data a Web site may wish to collect, known as the "P3P base data schema"
- A standard set of uses, recipients, data categories, and other privacy disclosures
- An XML format for expressing a privacy policy
- A means of associating privacy policies with Web pages or sites, and cookies
- A mechanism for transporting P3P policies over HTTP

Though P3P is a very good tool for web privacy it has some serious issues. The main problem with P3P is that it cannot enforce the rules! If a site wants to say one thing but actually executes another policy, it can do that. Another problem is that P3P requires users to make privacy decisions in advance, without regard to specific circumstances in a particular site context. Though P3P is useful to address user privacy challenges, it has some issues that could effect its adoption by the industry such as [122]:

- The languages available to describe user privacy preferences are not sufficiently expressive

- P3P policies published by web sites are not trusted by users

- P3P framework does not provide a coherent view of available privacy protection mechanisms to the user

According to the Platform for Privacy Preference 1.1 Specification "Significant sections were removed from earlier drafts of the P3P 1.0 specification in order to facilitate rapid implementation and deployment of a P3P first step. A future version of the P3P specification might incorporate those features after P3P 1.0 is deployed. Such specification would likely include four major components that were part of the original P3P vision but not included in P3P 1.0:

- a mechanism to allow sites to offer a choice of P3P policies to visitors

- mechanisms to allow for non-repudiation of agreements between visitors and Web sites

- a mechanism to allow visitors (through their user agents) to explicitly agree to a P3P policy

- a mechanism to allow user agents to transfer user data to services

"

6.1.3 Privacy in The Web services architecture

The Web services architecture (WSA) [123] provides a conceptual model and a context for understanding Web services and the relationships between the components of this model; in addition the WSA describes a set of requirements [124] for a standard reference architecture for Web services; the one of interest here is the set or requirements which enables privacy protection for the consumer of a Web Service across multiple domains and services, which are(based on [123]):

- the WSA must enable privacy policy statements to be expressed about Web services.
- advertised Web Service privacy policies must be expressed in P3P [121].
- the WSA must enable a consumer to access à Web Service's advertised privacy policy statement.
- the WSA must enable delegation and propagation of privacy policy.
- Web Services must not be precluded from supporting interactions where one or more parties of the interaction are anonymous.

The issue with privacy policies is that they are typically much more of the obligatory form than access control policies. A policy that requires a provider agent to properly propagate P3P tags, for example, represents an obligation on the provider entity. However, it is not possible to prevent a rogue provider agent from leaking private information, for that the WSA also highlight some key security considerations, here are the list of which may have direct impact with privacy:

- **Distributed Policies:** Security Policies can be used to define the access privileges of request and responses between parties. These polices can be validated at run time in the context of interaction. Each party in an interaction validates its own policies.

- **Trust Policies:** Distributed policies that apply to the environment of the other sides party in an interaction. Trust policies may be recursive. They may be defined against trust policies of involved parties and even whole domains. Distributed Identities, Policies and Trust can be described and processed by a machine. Trust mechanisms can be used to form Delegation and Federation relationships. These mechanisms can be used to facilitate secure interactions between web services across trust boundaries in a distributed fashion.

- **Secure Discovery Mechanism:** Secure Discovery Mechanism enforces policies that govern publication and discovery of a Service. When publishing a Service, an identity is usually necessary to assert Service publication policies, except for some cases of peer-to-peer discovery. When a requester entity discovers a Service, it may or may not provide an Identity; discovery may well be anonymous.

- **Trust and Discovery:** The decision whether or not to trust a particular Service arises when a requester entity chooses a Web Service from a previously unknown provider entity. This leads to an important difference between manual discovery and autonomous discovery. When manual discovery is used, a human makes the judgement of whether to trust and engage a previously unknown Service that is discovered. Whereas with autonomous discovery, a machine makes this decision. Since people may not

trust a machine to make significant judgement decisions, agents performing autonomous discovery are often limited to using private discovery services that list only those services that have been pre-screened and known to be trustworthy by the requester entity.

6.1.4 WS-Privacy

WS-Privacy enables users and Web services to state privacy preferences and Web services to state and implement privacy practices. WS-Privacy is not yet published but it has been proposed within the context of Web Services Security. It will be using a combination of WS-Policy, WS-Security, and WS-Trust, organizations can state and indicate conformance to stated privacy policies. This specification will describe a model for how a privacy language may be embedded into WS-Policy descriptions and how WS-Security may be used to associate privacy claims with a message. Finally, this specification will describe how WS-Trust mechanisms can be used to evaluate these privacy claims for both user preferences and organizational practice claims [22].

WS-Privacy will allows web Service providers and requestors to stat their privacy preferences and organizational privacy practice statements. It is very likely that WS-Privacy will be similar to P3P to allow for privacy policy exchange and agreement for web services.

6.1.5 Related Work on Privacy

The huge development of the number of Web services application over the past years meant that the use of personal information by these application has increased greatly. This has lead to the concerns of misuse of these personal data whether by malicious or accidental means. Which in turn lead various legal and technical bodies to come up with ways to address these privacy issues. However, a flexible, practical system for ensuring that these rights are respected in web services is still missing [125, 126]. Existing privacy policy approaches for privacy protection, such as making the service providers privacy policy known to the user, or the use of P3P [121] privacy policies, are inadequate. In the former case, the user cannot know for sure whether or not the provider will honor its policy; in the latter case, there is no flexibility for the user to specify her own policy for governing her own personal information the providers policy is the only one offered. The work in [126] tries to address the above limitation by developing a flexible user-oriented privacy controller based system that preserves legislated user privacy rights expressed in the form of user privacy policies. The processing of the policies, as well as the enforcement of compliance to the policies, are the responsibility of privacy controllers.

An important issue regarding privacy with Web services architecture was discussed in [127]. The WSA does not cover all the privacy issues that might arise in a real scenario; for example there are no consideration regarding the discovery agencies in web services. Discovery agencies provide a searchable set of web services descriptions in centralized our distributed UDDI registries [24]. Some of the problems comes from the fact that any entity in WSA can act in one or multiple roles. This can become an issue for privacy if for example a discovery

agencies also act as Service providers; or when discovery agencies can delegate tasks to other services, in such cases its not clear how the discovery agencies will protect the requestors sensitive information.

One of the main topics associated with privacy is that of trust. However, the meaning of trust can be sometime ill-defined, the work in [128] describe six characteristics of trust: Implicitness, Asymmetry, Transitivity, Antonymy, Asynchrony, and Gravity. Each of these characteristics has different effects on trust depending in the environment. The work concludes that due to the above characteristics trust is difficult to measure.

Another important issue related to privacy, in particular for Web Services is the ability to enforce and verify their compliance with privacy policies. Structured policy languages can play a major role by supporting automated enforcement of privacy policies and auditing of access decisions. In [129] the authors list the requirements for any standard structured language for supporting expression and enforcement of privacy policies. This includes: it should be able to describe the constraints and the purposes for which the data was collected, the language must enable the expression of directly-enforceable policies, it should be platform independent, and the language used for privacy policies must be the same as or integrated with the language used for access control policies. Also in [129] the author gives a brief overview of the Enterprise Privacy Authorization Language (EPAL) from IBM [130] and OASIS Standard eXtensible Access Control Markup Language (XACML) [131, 132], and then compares the two languages to show where they differ. Concluding that functionality of XACML 2.0 is a superset of EPAL 1.2 and provides more features.

P3P is usually implemented as an agent working with the users web browsers

to manage/supervise users' privacy policy. In [133] the authors propose a P3P privacy enhancing agent to address the privacy concerns from the client side by helping users analyzing web site P3P policies. This agent will retrieve P3P policy, enforce user privacy preference, and handel the decision making and negotiation processes based on user preferences, transaction history with the particular web site and related privacy practice knowledge from other agents.

6.1.6 Enabling Privacy with P3P in Federated Environment

The aim here is to utilize the benefit of P3P in assisting to protect the user privacy at the same time overcoming P3P enforcement limitation by deploy it in federated system such as the Liberty Alliance. The novelty of the system lies in the integration of the Web Services technologies and P3P [121], with the Generic Authentication Architecture (GAA)from the Third Generation Partnership Project (3GPP) [95].

The rest of this chapter describes the architecture of the system and shows how this integration takes place.

Generic Mobile Web Services Platform for Healthcare

The architecture used is based on the Generic Mobile Web Service Platform described in 5.6, with some modifications. The platform consists of four main actors:

- The Mobile Operator(MO): This is the Mobile Phone Network Operator , containing the Bootstrapping Function (BSF) that is part of the GAA.

- The Authentication Server: (AS) contains two main entities; the first is the Network Application Function (NAF) which is used to communicate with BSF. The other entity is the Iden-All P3P privacy policy files must comply with identity Provider (IdP), that is used as the identity the minimum standard set by the AS, which is provider for the SSO system. checked during the registration.

- Service Providers: This refers to any service provider participating in the system that provides a service to the User.

- The User: the user can be anyone with 3GPP UMTS Mobile Device (MD) who wants to access the system.

The entities above interact with each other as described in Figure 6.1. When the User request access to the AS, the AS will request the user's GAA security credentials. The User will start the bootstrapping procedure with the BSF of the MO as described in [134]. Using the Ub interface the BSF will generate for the User a random identifier (B-TID), and using the AKA protocol both (the BSF and the USER) will generate a secret session key K_s. The user will then use B-TID and K_s as user name and password to login to the NAF located at the AS. If successful the User will be authenticated to the IdP SSO system such as [52] to access the various services available that are registered with the AS.

The privacy of the User is managed as follows:

- All SPs to be part of the AS must have a P3P privacy policy file. These policies describe who collects what data and for what purpose.

- All P3P privacy policy files must comply with the minimum standard set by the AS, which is checked during the registration.

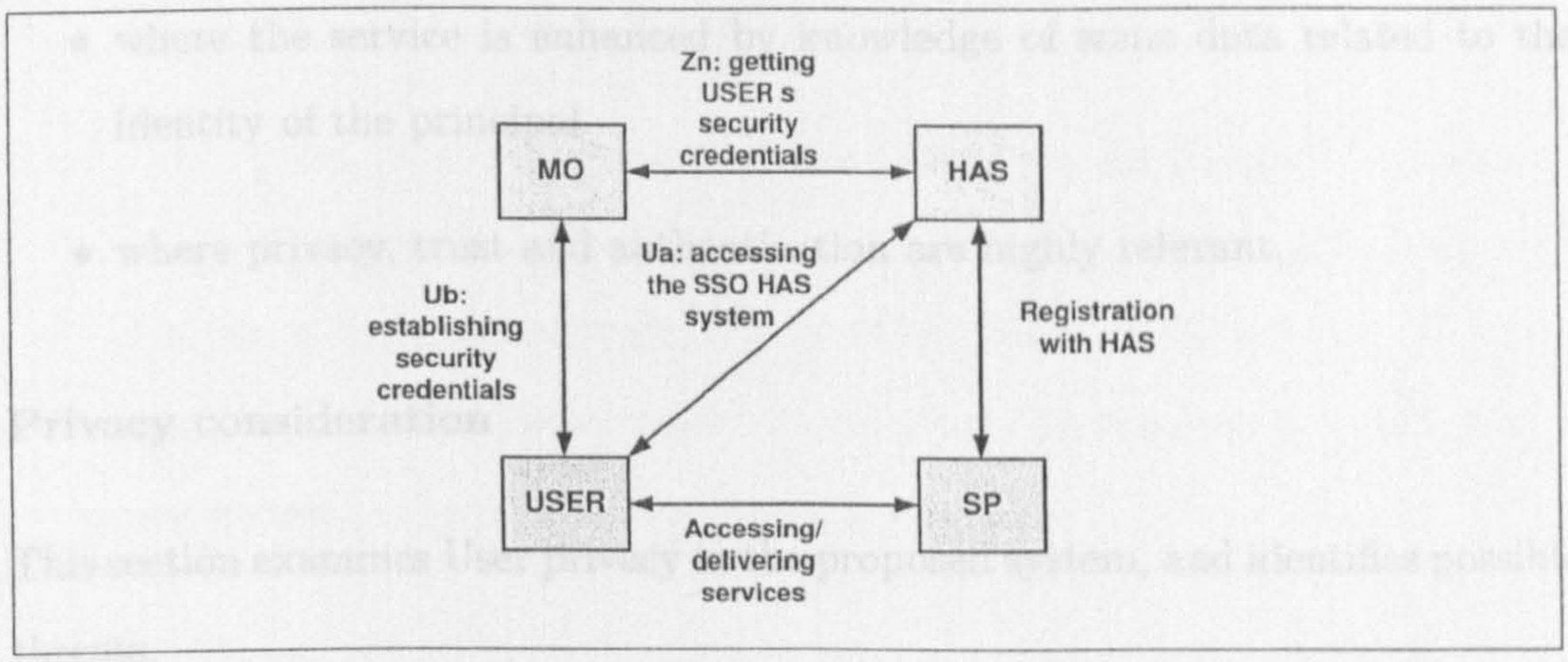

Figure 6.1: System Architecture

- The User must set their own P3P privacy file. When accessing a SP the two P3P files are compared (User and SP) if they match then the system will just sit in the background; if not the User will be alerted and he/she can make decision to accept the service or not.

In addition to this, the privacy of the user will be protected by encrypting all the messages sent by the user as well as changing the identity of the user each time the user accesses different SP. The details are given in the next section.

In our proposal the AS acts as an Identity Provider between Web Service Providers and each of the Web Service Users (Mobile Stations). The Mobile Operator owning the SIM deployed in the Mobile Station, acts as an Authentication Authority to the AS.

We utilize the combined Liberty & 3GPP GAA model, as defined in [114], to combine the Service Orientated Architecture of Web Services with a Mobile End User end point. We target the provision of identity-consuming services where knowledge of the user (principal) is important. In this way we address the highest value scenario:

- where the service is enhanced by knowledge of some data related to the identity of the principal.

- where privacy, trust and authentication are highly relevant.

Privacy consideration

This section examines User privacy in the proposed system, and identifies possible threats.

It is assumed that both the AS and the MO are trusted entities and their system is very hard to compromise or break.

- Privacy requirements and policies: This is achieved first by the AS setting a general privacy policy guidelines, which all the SP's must comply with in their P3P privacy policy files. The main issue here is that it is not easy to technically force the SP to comply with its P3P policy, however protection can be put in place in a form of legal contract during the registration with the AS. This will take place when the SP register with the AS. Another alternative is to deploy XACML [129, 131] to assist with the policies enforcement.

- User Identity: the User will access various SPs with different IDs only known to the AS (trusted entity), therefore for any one spoofing on the network will not be able to follow the User's activities even if more than one 'bad' SPs joins and share their knowledge about the system users. This is achieved because of the SSO system used by the AS.

- Communication privacy: all communication messages are encrypted and

can only be read by the intended receiver.

- Single point of failure: the IdP server part of the AS is arguably the single point of failure in the system as it manages the Users identities, approve the policy files, and generate the security token for the system. If this is compromised then the system will collapse, however; we assumed that it was secure.

6.2 Summary

After an overview of the issue of privacy in this chapter, we reviewed key technologies and related work that can be used to assist in protecting user privacy in mobile Web services. P3P was identified as a technology with great potential in this issue; we finally proposed a mechanism to enable the use of P3P in federated environment and limiting some of its drawbacks.

Chapter 7

Authentication with Timestamps in Federated System

7.1 Introduction

One of the key factors in the process of determine the meaning of a message in authentication protocols are the "Time variant parameters". There is a need to have a method to ensure that the message is fresh before action upon it; and that the message is not a reply of an old instance.

Time variant parameters may be used in authentication protocols to prevent replay and interleaving attacks, to provide uniqueness or timeliness guarantees, and to prevent certain chosen-text attacks. Time-variant parameters which serve to distinguish one protocol instance from another are also called nonces, unique numbers, or non-repeating values. In authentication protocols the commonly used time variant parameters are: timestamps, sequence numbers and nonce/random numbers [2].

Authentication protocols based on timestamps and sequence numbers usually require one less message than protocols based on random numbers. In addition, the timestamp based protocol has the advantage that it is not required to maintain state information [2, 135].

The understanding of how the freshness component works in the message is vital to the overall design of the authentication protocol. One of the essential requirements is that the freshness component should be bound together with the rest of the message in a way that it can not be later attached to a replayed message.

7.2 Motivation for using Timestamps in Authentication Protocols

The notion of time is fundamental for describing and verifying security properties related to the expiration of keys and the freshness of messages [136]. Timestamps are necessary in authentication protocols that support multiple authentication without multiple request to an authentication server [137]. The motivation for using Timestamps in Authentication protocols can be summarized as follows:

- Timestamps may be used to provide timeliness and uniqueness guarantees, which guarantees the freshness of a message.
- As a result of the above point this helps to detect message replay; and to detect forced delays.
- Timestamps may also be used to implement time-limited access privileges.

- Timestamps in authentication protocols offer the advantage of fewer messages (typically by one) than other challenge-response protocols. This is of special importance in mobile networks such as GSM/UMTS where there is usually a cost associated with each message sent.

7.3 Issues with using Timestamps in Mobile Web Services

The topic of timestamps is not new in the security literatures. [135, 138, 139] highlight various issues when using timestamps in communication protocols. This section reviews the issues of concern to the Mobile Web services environment.

7.3.1 Clock Synchronization

Timestamp based protocols require that time clocks be both synchronized and secured. The possibility of adversarial modification of local time clocks is difficult to guarantee in many distributed environments such as in Web services. While technical solutions exist for synchronizing distributed clocks, if synchronization is accomplished via network protocols, such protocols themselves must be secure, which typically requires authentication; this leads to a circular security argument if such authentication is itself timestamp based [2].

As described in chapter one in section 1.3.3; if timestamps are used as freshness guarantees by reference to absolute time, then the difference between local clocks at various machines must be much less than the allowable age of a message deemed to be valid.

7.3.2 Trusted Clocks

As mentioned before, the timestamps based protocol can help to ensure freshness of messages. The freshness is usually achieved as follows. The party originating a message obtains a timestamp from its local clock, and cryptographically binds it to a message. Upon receiving a timestamped message, the second party obtains the current time from its own clock, and subtracts the timestamp received. The received message is valid provided the timestamp difference is within the acceptance window.

Therefore, there is a need for a level of trust that the other party clock is functioning correctly. In a server/client communication it is generally assumed that the server clock is trusted to function as expected, however the same can't be easily assumed at the client side. This is an issue when mutual authentication is required and if the authentication protocol is timestamp based. The clocks in most clients handsets in GSM/UMTS systems can be easily modified by the users (for good reasons such as setting the time of their mobile phones). Therefore if such devices (e.g. mobile phone) are used in such protocol, there will be a need to establish the trust ensuring that all clocks are behaving as expected.

7.4 Timestamp Authentication Protocols in Federated System

Federated systems such as Liberty [52] are suitable to adapt timestamp based authentication protocols for the following reasons: there exist a trusted entity (e.g. IdP) which can be used to manage clock synchronization, it is common for

the different entities in the system to interact with each other and in the case of GSM/UMTS devices the potential efficiency gains are considerable, since less messages are needed for authentication.

However, these devices' clocks may not be trusted by all devices in the system as they are handled by users who have easy access to the clock in these devices. Again the trusted entities in the federated system can play important role to increase the level of trust.

The protocol proposed in this work uses digital cryptography to protect the integrity and the confidentiality of the messages in the system, in a similar way as in [2]. It is possible to set the following assumptions in the federated system:

- The system consists of three types of actors: Identity Provider (IdP), which acts a trusted third party where its main role is to act as an authentication server in the federated system; the User or the Mobile Device (MD) and is reference to GSM/UMTS mobile devices. the Service Provider (SP) who provides a service to MDs.

- All actors have agreed on a specific signature algorithm. The signature on data X using private key K is written $s_K(X)$.

- All actors posses a trusted copy of the IdP public key used for verification in the signature scheme.

In the proposed environment the following conditions hold (some are based on the work in [135]):

- It is important for the clocks used in the system to be synchronized but not

necessary to be correct. The important thing is that all devices agree on the current time for the purposes of authentication.

- The clocks used by the communicating devices are not precisely synchronized. The differences in clock values are less than some threshold value T.

- Messages sent from one device to another are subject to a maximum transit delay of D.

- Giving the current time t_c, and the timestamp in the message is t_m, then the message is considered 'fresh' if t_c-T-$D \leq t_m \leq t_c + T$.

- From the above the time of acceptance interval or the 'window of acceptance' is $[t_c$-T-$D, t_c + T]$.

- A timestamp based authentication protocol is used between pairs of devices in the system to guarantee the freshness of the messages, and that the protocol is designed in a way that the recipient of a timestamped protocol message can guarantee its origin and integrity by cryptographic means.

7.4.1 The Proposed Scheme

The concept behind the scheme is to increase the level of trust in the user's clock, by getting a trusted third party to approve the time from the user's clock. When the user (i.e MD) authenticates to the IdP to obtain a 'user token' (UT) to access an SP. The IdP will attach a signed value of the MD current time $t_{c_{MD}}$ and the IdP current time $t_{c_{IdP}}$ to the UT. So when the user accesses an SP, the SP can compare the MD timestamps, the one attached with the UT and the timestamp

sent with the message t_m. This way the SP will be able to detect if the users clock has changed significantly since the session started (i.e. since the user authenticated to the IdP) to take an appropriate action. The proposed scheme takes the following steps in a federated system, also presented in Figure 7.1:

Message 1 User → IdP: t_m, Access Info

Message 2 IdP → User: $s_{SK_{IdP}}(t_{User}, t_{IdP})$, UT

Message 3 User → SP: $s_{SK_{IdP}}(t_{User}, t_{IdP})$, UT, t_m

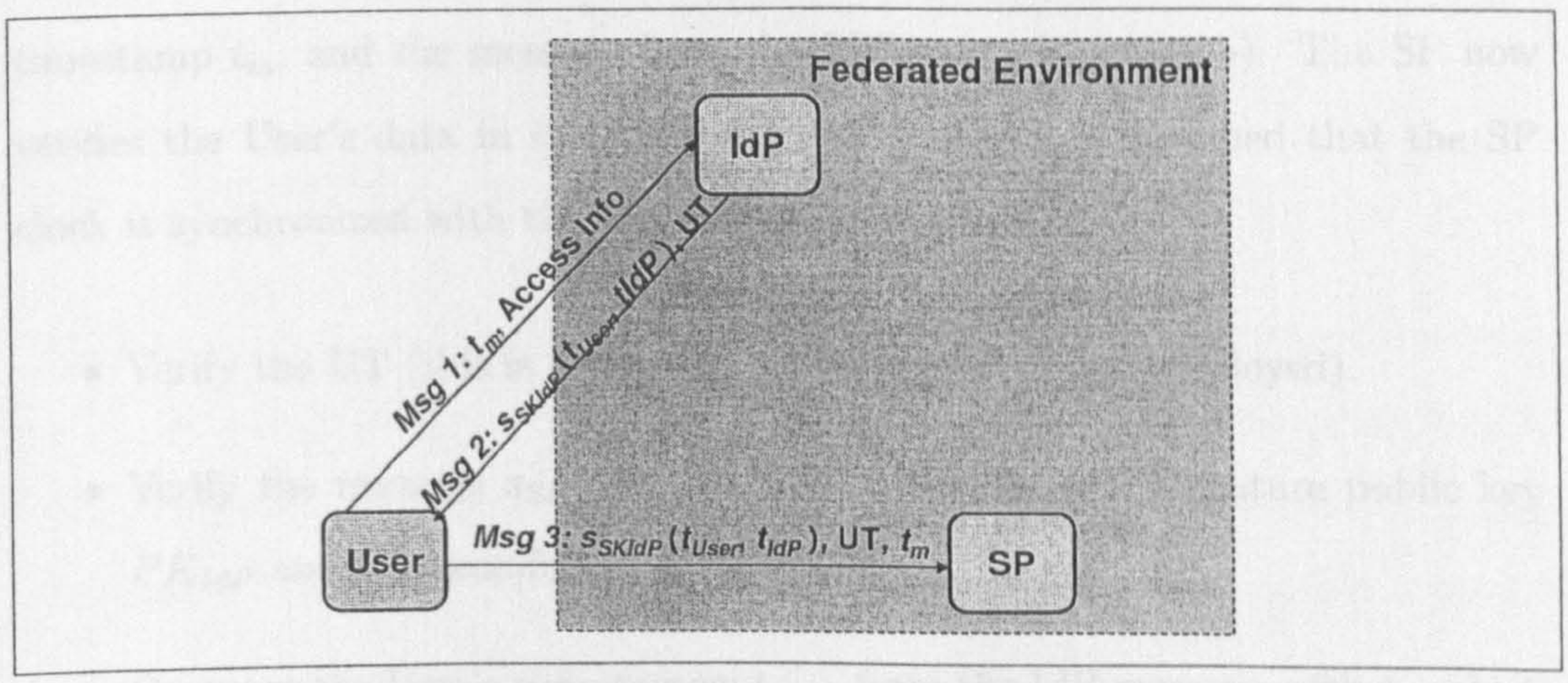

Figure 7.1: Timestamp Authentication scheme in Federated System

In message 1 the user will attach its current timestamp t_m with its security credentials (i.e. Access Info) to access the federated environment through the IdP. The IdP will compare the user's t_m with its local clock t_{IdP}. If the different in times (both directions) is bigger than a threshold value T, the IdP will ask the user to adjust its clock by sending a reply message with the its current timestamp t_{IdP}. On the other hand if the difference in time is less than T and the verification

of the Access info (e.g. Username/Passowrd) is successful, then the IdP will do two things:

- Generate a User Token UT.
- Attach its timestamp t_{IdP} to the user timestamp t_{user} (note that at this stage $t_m = t_{user}$) and sign them with the its signature private key SK_{IdP}.

In message 2 the IdP will send to the user the information generated above $s_{SK_{IdP}}(t_{User}, t_{IdP})$, UT.

In message 3 the User will try to access the SP by providing its UT, its current timestamp t_m, and the message from the IdP $s_{SK_{IdP}}(t_{User}, t_{IdP})$. The SP now verifies the User's data in the following way (note: it is assumed that the SP clock is synchronized with the IdP clock):

- Verify the UT (this is subject to the federated system deployed).
- Verify the message $s_{SK_{IdP}}(t_{User}, t_{IdP})$ using the IdP signature public key PK_{IdP} and if successful
- Compare the User's timestamps; t_{User} from the IdP message with t_m which is the User current time. Message will be accepted if the difference between t_m and t_{User} is less than T. This will ensure to the SP that the User's clock is correct (i.e. synchronized with the IdP clock)

If the above conditions are met then a timestamp based authentication protocol can be used after this stage to authenticate the user to the SP. Note that no extra message were introduced, since these are the normal message exchange in the federated environment scenario [52].

Some of the major issues with timestamp authentication protocols are the clock synchronization presented in section 7.3.1 and the trusted clock issues which were presented in section 7.3.2. The proposed scheme solves the User's trusted clock issue, where the SP can now trust the user's clock as its value has been tied to the IdP trusted clock at the beginning of the session. That enables the SP to detect any changes in value to the user's clock. The clock synchronization issue was also addressed, which showed that its important for the clocks used in the system to be synchronized but not necessary to be correct. The important thing is that all devices agree on the current time for the purposes of authentication.

7.5 Security Analysis and Evaluation

Experience based validation/verification approach together with the protocol design principles presented in chapter one was the chosen methodology for evaluating the security in our proposed protocols. The aim of the security analysis presented in this section is to evaluate the security threats that could raises directly from our protocol (i.e. the use of timestamp for authentication). The evaluation is considered over different security objectives as follows:

7.5.1 Integrity of the clock values

This is concern with the integrity of the messages in the system, mainly the values of tm, t_{User}, and t_{IdP}. The aim of this attack is to change the value of tm, t_{User}, or t_{IdP}. In message 2 the attacker will not be able to change the value of t_{User} or t_{IdP} without detection as these values are signed by the IdP. Therefore, any

changes will be detected either by the User or the SP. This is also true in part of message 3. However, both in message 1 and part of message 3 an attacker could change the value of tm without detection. Possible solution is for the User to sign tm. This was not introduced in the design to reduce the work load on the user, however depending on the application on which this could be deployed it may be necessary for the User to sign tm.

7.5.2 Clock failures

Weather it's caused by an attacker or not a clock failure in any of the system entities will cause the system to fail. By failure we mean the value of the clock has shifted by more than T. A failure in the User clock will denial the IdP from generating the user token, that the User needs to access the SP. Failure at the SP clock again will cause the system to fail, as the different between the tsp and t_{User} could be bigger than T, that will cause the SP to ask the User to try again. This will be the case until the SP resynchronizes its clock with the IdP. Note: we have assumed during the design of this protocol that the SP clock is synchronized with the IdP. It's more interesting case when the IdP clock fails. What will happen is this; the User will send its t_{User} in message 1. Because of the failure in the IdP clock, the different between t_{User} and t_{IdP} will be bigger than T, the IdP will reply asking the User to adjust its clock value to t_{IdP}. The User will then adjust its value and resend message 1. After receiving message 2, the user will add tm to message 2 to create message 3, and send it to the SP. Now if the SP clock is synchronized with the IdP the authentication will succeed as the different in value between t_{SP} and t_{IdP} will be less then T. However, if the SP clock were not synchronized by when the SP receives message 3 the authentication will fail.

In our proposed scheme the IdP clock is assumed to be the master clock that all SPs in the system synchronizes their clock with.

It's interesting to note that in this scheme a failure in the IdP clock will not fail the system, it will only cause some delay as the User will have to adjust its clock value and resend message 1. One possible solution to solve this problem is this: every time an SP synchronizes its clock with the IdP, if the different in value is bigger than T, then the SP notify the IdP. If the IdP receives many notifications for such error from many (many here will depend of the size of the network) different SPs the IdP could assume the there is an error with its clock.

7.5.3 Denial of Service attack

Unfortunately the system is subject to a Denial of Service (DoS) attack. In essence clock failures are an accidental DoS attack. An attacker could change the value of tm in message 1. I the attacker manages to change the value of tm so that the different in value between tm and t_{IdP} is bigger than the threshold value T, then the IdP will respond by asking the user to adjust its value of tm to make it equal to t_{IdP}. The attacker could repeat this attack every time the user requests the IdP to create its user token. The result the IdP will never be able to create the user token. Possible solution to this attack is for the user to sign message 1, however this will lead the user to manage its integrity keys which may or may not be an issue depending on the application its deployed in. another alternative is for the user to encrypt message 1, though an attacker will not be able to read the message it is still possible to change the network traffic in the message without been detected.

The analysis in this section was mainly to consider possible security threats

to the system due to the use of timestamp for authentication, it did not consider other potential threats for two reasons; first the aim of this protocol is to see the viability of using timestamp for authentication in federated system, and the investigation suggest that yes it is possible but with some care as some potential security threats do exist. The second reason is that more detail security analysis which address security issues in federated system were presented in earlier chapters, which are still applicable here such as the failure if the IdP or SIM cloning (incase the user is mobile device).

7.6 Summary

Time variant parameters are important in all authentication protocols, which are used to prevent replay and interleaving attacks, to provide uniqueness or timeliness guarantees, and to prevent certain chosen-text attacks. One of the key advantages of timestamp based authentication protocol is that they usually requires one less message than protocols based on random numbers, in addition timestamp based protocol has the advantage of that its not required to maintain state information. This efficiency gain that timestamp based authentication protocol can offer over other authentication protocols can be of great value, especially when mobile phones are used in the system, since there is usually an associated cost related to the number of messages sent.

A list of motivations for using timestamp based protocols was given in section 7.2. Key challenges for timestamp based authentication protocols such as clock synchronization and trusted clocks were discussed in Section 7.3. We have demonstrated how to address these concerns and propose a mechanism to use

timestamp authentication protocol even if mobile devices with untrusted clocks are used.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Web services are becoming a model on which to build distributed applications that uses the Internet. These services predicate a set of standards that provide a simple and consistent way to access the functionality of diverse systems via the World Wide Web. As Web services are not just being used to integrate internal systems, but they are also integrating data sources from outside the organization. That was one of the main reasons that mobile commerce developers and researchers looked to benefit from Web service. However; one of the key issues with such environment is the security of the system, and this was the focus of this thesis.

Web services security standards/solutions were developed with the fixed network in mind, and because of that current Web services security solutions are not practical nor secure enough to adopt in mobile environment. Furthermore, Mobile networks can offer various platforms/architectures in which Web service

security systems can use to secure mobile Web services applications. In this thesis we have examined both Web services security mechanisms and the mobile networks security architectures/technologies.

The basic specification of Web services completely ignored the need for secure services that are based on a authentication and authorization infrastructure [77]. Federated systems have been identified as good mechanisms to deploy in mobile Web services application. As it simplify the authentication process and can assist to enforce privacy policy.

The existing cryptographic mechanisms in GSM and UMTS networks do provide some level of protection to the information security. However, some potential attacks as in DDoS attacks, or large-scale worm infection has not yet been quantified, this is particulary true in UMTS network. And with the introduction of new services, new types of mobile devices, and further evolution of the network architecture (e.g. IMS and 3G/WLAN interworking), will likely introduce new threats of the security of the network. The understanding of such security threats is essential when designing new services for these networks. One of the main issues we faced in this research is the lack of available public data of experimental investigation on such security threats that are carried on within the industry laboratories.

Single Sing On is one of the mechanism that can assist with user authentication in mobile environment. This is very important in mobile applications as it is inconvenient for the user to keep inputting username/password each time they access a service. However, registration for such systems have some security concerns in various Single Sign-On schemes. We have proposed a solution where a security token is used to address this issue, with a protocol that uses the GSM

system to protect user privacy in Single Sign-On scheme. The proposed protocol allow for the user to have full anonymity as far as the service providers are concerned; however it is possible for a trusted authority to reveal the identity of the user if he or she is suspected of illegal activities.

SAML is used in Web Service to address the issue of portable identity, and is adopted it in Mobile Web service. Care must be taken when designing a SAML protocol for Mobile Web service, to balance the requirements between protocol security and efficiency. When designing an authentication protocol with SAML for Mobile Web service, the network architecture must be taken into account to achieve the required balance between security and efficiency.

We have identified the 3GPP Generic Authentication Architecture (GAA) as a useful tool in assisting in mobile Web service authentication. We have used that in the development of the *generic mobile Web service platform*. This generic platform is built on top of the 3GPP GAA and makes use of Liberty Alliance federated environment model. The system can be used to provide a base platform for mobile Web services applications and solutions. Initial results showed that the system is robust and flexible to allow for more investigations to be carried out to try to solve key privacy and security problems in mobile Web services.

The proposed architecture allows developers and researchers to rethink current distribution structures and business models for the sourcing and delivery of digital services to mobile subscribers. The main contributions of system architecture, protocols, and enabling data structures could form the basis to provides:

1. the user with a high level service discovery interface plus anonymity from Web Service Providers;

2. the Mobile Operator with a pivotal role and a revenue generating opportunity in the provision of a web services security and payment platform;

3. the Service Provider with a secure, scalable distribution channel.

User privacy has always been important when dealing with users transactions over digital networks. In mobile Web service user privacy becomes critical, as people keeps all sort of personal and work related data in their mobile phones and PDAs, in addition to the growing number of services that uses location based services which could effect the privacy of the user if not managed properly.

Chapter five also gives an overview of the issue of privacy, a review of key technologies and related work that can be used to assist in protecting user privacy in mobile Web services. P3P was identified as a technology with great potential in this issue. we finally proposed a mechanism to enable the use of P3P in federated environment and limiting some of its drawbacks. Though P3P is useful to address user privacy challenges, it has some issues such as the lack of trust by the sites who publish it and the ability to enforce its policies.

We argue that the above issues can be addressed in federated environment where some degree of trust is available (e.g. circle of trust in the Liberty Alliance model). In the proposed system this is achieved by the AS (Authentication Server)setting a general privacy policy guidelines, which all the SP's must comply with in their P3P privacy policy files. The main issue here is that it is not easy to technically force the SP to comply with its P3P policy, however protection can be put in place in a form of legal contract during the registration with the AS. This will take place when the SP register with the AS. Another alternative is to deploy XACML to assist with the policies enforcement.

Reducing the number of exchange messages in mobile Web service protocol design is desirable for many reasons including reducing security risks (as less data will be flying around), cost saving both in terms of resources and financial cost (many mobile operator charge for most data traffic over its network). Timestamp based authentication protocols usually requires one less message than protocols based on random numbers, which makes it suitable for mobile Web services environment. This efficiency gain that timestamp based authentication protocol can offer over other authentication protocols can be of great value, especially when mobile phones are used in the system, since there is usually an associated cost related to the number of messages sent. In addition time variant parameters are important in all authentication protocols, which are used to prevent replay and interleaving attacks, to provide uniqueness or timeliness guarantees, and to prevent certain chosen-text attacks. However timestamp based protocols have their own challenges such as clock synchronization and trusted clocks. Chapter seven demonstrated how to address these concerns and propose a mechanism to use timestamp authentication protocol even if mobile devices with untrusted clocks are used through the use of trusted third party in federated environment.

8.2 Achievement

This section lists the main achievement of this research, which are driven from the motivation and the aims and objectives sections earlier. These are outlined as follows:

1. Identify the limitations/weakness of current Web services security technologies/techniques when applied in GSM/UMTS environment, which can be

summarized in:

(a) Web service does not have clear entities to play the role of Authentication Server, this is an issue as the GSM/UMTS environment is a centralized architecture. Mobile operators are suitable to play this role.

(b) Some of protocol structure and protocol vocabulary such as in SAML, are inefficient in GSM/UMTS environment, as they consume many messages and a large space within the message. There is a cost in most GSM/UMTS messages exchange.

(c) The security registration process in SSO schemes can be improved.

2. Identify the limitations/weakness of current security system in GSM/UMTS technology, to be addressed when developing Mobile Web services system.

3. Develop novel protocols to address the issues of authentications and privacy in mobile web services environment.

4. Design and build prototype to test the various techniques and protocols developed by this research.

8.3 Limitation

Though the thesis attempts to cover the topic of Authentication and Privacy in Mobile Web Service in some details, due to the nature of the PhD research work and the time constrains. The following is a list of the main limitation of the work:

- Proposed protocols could not been verified with the industrial, the mobile operator would not disclose the details of their security system (for security reasons).
- Important related topics were not covered due to time constrains; these include trusted computing techniques, and XACML
- Could not run formal security verification on a limited scale of the system, due to initial assumptions and research objectives.

8.4 Future Work

In the course of this research, several prospects for future work have become evident and some issues may be the subject for further study. These are summarized below.

- Vulnerability analysis: there is no formal analytical tool to assess the vulnerability and strength of a system security proposal. In particular, the analysis on the interdependency among various system components and security operations.
- Trusted third party: the most trusted element in the mobile ecosystem is the mobile operator, this is due to its role in the system. We have utilized this fact and used the mobile operator as a trusted IdP/AS in federated environment that assist SPs to authenticate mobile devies/users. It is interesting to see if it is possible to address the issue of mobile users authentication to SPs in a secure and practical means without mobile operator involvement.

- SAML: the terminology for SAML is not suitable to be used in mobile networks due to the length of its vocabulary, that makes it use extra resources. More work is needed to make SAML efficient to be used in mobile Web services systems.

- GAA from 3GPP: As proposed in chapter 5, the GAA can be a great authentication tool to integrate mobile users with Web services applications. However; more work is needed to evaluate and improve efficiency of the system. Also its important to be specific on what it means that the user has been authenticated by the operator GAA system?

- Trusted computing: TC is gaining considerable industrial support, though TC was looked at during the work of this thesis, not much time was spent on it due to the focuss of this thesis on other related issues and time constraint.

- XACML: Chapter six identified P3P as technology that could assist in protecting user privacy in mobile Web service environment, due its wide use among web application. Another alternative is to deploy XACML to assist with the policies enforcement. In particular how it can be deployed within a federated system.

Appendix 1

Bibliography

[1] B. Schneier, *Applied Cryptography: Protocols, Algorithms and Source Code in C.* John Wiley and Sons Inc, 1996.

[2] A. Menezes, P. van Oorschot, and S. A. Vanston, *Handbook of Applied Cryptography.* CRC Press Inc, 1996.

[3] "DATA ENCRYPTION STANDARD (DES)," tech. rep., Federal Information, Processing Standards Publication 46-2, December 1993. Supersedes FIPS PUB 46-1, 1988 January 22.

[4] A. Nash, W. Duane, C. Joseph, and D. Brink, *PKI: Implementing and Managing E-Security.* RSA Press, 2001.

[5] "Security aspects," tech. rep., ETSI European Telecommunications Standards Institution, June 2001. GSM 02.09 version 8.0.1 Release 1999, Digital cellular telecommunications system (Phase 2+).

[6] "ADVANCED ENCRYPTION STANDARD (AES)," tech. rep., Federal Information, Processing Standards Publication 197, November 2001.

[7] "Specification of the 3gpp confidentiality and integrity algorithms; document 2: Kasumi algorithm specification," tech. rep., ETSI European Telecommunications Standards Institution, September 2005. (Universal Mobile Telecommunications System (UMTS); 3GPP TS 35.201 version 6.1.0 Release 6).

[8] "Specification of the A5/3 Encryption Algorithms for GSM and ECSD, and the GEA3 Encryption Algorithm for GPRS," tech. rep., 3GPP 3rd Generation Partnership Project, September 2003. TS 55.216, A5/3 and GEA3 Specifications, Release 6.

[9] C. Mitchell, ed., *Security for Mobility.* Telecommunications Series 51, IEE, 2004.

[10] M. Abadi and R. Needham, "Prudent engineering practice for cryptographic protocols," *IEEE*, pp. 122–136, May 1994. Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy.

[11] M. Burrows, M. Abadi, and R. Needham, "A logic of authenticaion," *ACM Transaction on Computer Systems*, vol. 8, pp. 18–36, February 1990.

[12] S. McClure, J. Scambray, and G. Kurtz, *Hacking Exposed: Network Security Secrets and Solutions.* Osborne/McGraw Hill, third edition ed., 2001.

[13] D. Babb, C. Bishop, and T. Dodgson, "Security issues for downloaded code in mobile phones," *IEEE: Electronics and Communication Engineering Journal*, vol. 14, pp. 219 – 227, October 2002.

[14] C. J. Mitchell, "The security of the GSM air interface protocol," tech. rep., Royal Holloway, University of London, August 2001.

[15] Vijay K. Garg and Joseph E. Wilkes, ed., *Wirless and Personal Communications Systems.* Prentice Hall PTR, 1996.

[16] S. Graham, D. Davis, and S. Simeonov, *Building Web Services with Java.* Sams Publishing, June 2005.

[17] B. Galbraith and W. Hankison, *Professional Web Service Security.* Wrox Press Ltd, December 2002.

[18] H. M. Deitel, P. J. Deitel, B. DuWaldt, and L. K. Trees, *Web Services a Technical Introduction.* Prentice hall, April 2003.

[19] E. Newcomer, *Understanding Web Services.* Addison-Wesley Professional, May 2002.

[20] "Extensible Markup Language (XML) 1.0," tech. rep., W3C, February 1998. W3C Recommendation.

[21] "Simple Object Access Protocol (SOAP)1.2," tech. rep., W3C, June 2003. W3C Recommendation.

[22] J. Rosenberg and D. Remy, *Securing Web Service with WS-Security.* Sams Publishing, April 2004.

[23] "Web Services Description Language (WSDL) 1.1," tech. rep., W3C, March 2001. W3C Note.

[24] "Universal Description, Discovery and Integration (UDDI) V3," tech. rep., The Advancement of Structured Information Standards (OASIS), February 2005. OASIS Standard.

[25] "XML-Signature Syntax and Processing," tech. rep., W3C, February 2002. W3C Recommendation.

[26] "XML Encryption Syntax and Processing," tech. rep., W3C, December 2002. W3C Recommendation.

[27] "SAML V2.0 Executive Overview," tech. rep., The Advancement of Structured Information Standards (OASIS), April 2005. OASIS Standard.

[28] I. B. M. Corporation, "Security in a web services world: A proposed architecture and roadmap." http://msdn2.microsoft.com/en-us/library/ms977312.aspx, April 2002.

[29] "WS-Security Core Specification 1.1," tech. rep., The Advancement of Structured Information Standards (OASIS), April 2006. OASIS Standard.

[30] Donal O'Mahony and Michael Peirce and Hitesh Tewari, ed., *Electronic Payment Systems for E-Commerce.* Artech House, 2001. Second edition.

[31] Heikki Kaaranen and Ari Ahtiainen and Lauri Laitinen and Siama k Naghian and Valtteri Niemi, ed., *UMTS Networks Architecture, Mobility and Services.* John Wiley and Sons Ltd, 2005. Second edition.

[32] "General system for mobile communications." www.gsm.org.

[33] K. Boman, G. Horn, P. Howard, and V. Niemi, "UMTS security," *Electronics and Communication Engineering Journal*, vol. 14, no. 5, pp. 191 – 204, 2002.

[34] J. R. Rao, P. Rohatgi, H. Scherzer, and S. Tinguely, "Partitioning attacks: Or how to rapidly clone some GSM cards," *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 31 – 41, May 2002.

[35] H. Yang, F. Ricciato, S. LU, and L. Zhang, ""securing a wireless world"," in *Proceedings of the IEEE*, vol. 94, pp. 442 – 454, IEEE, February 2006.

[36] O. Whitehouse and G. Murphy, "Attacks and counter measures in 2.5G and 3G cellular IP networks," tech. rep., Atstake Inc, http://www.atstake.com/reports/, May 2004.

[37] F-Secure, "F-secure malware information pages: Cabir." http://www.f-secure.com/v-descs/cabir.shtml, January 2006.

[38] F-Secure, "F-secure virus descriptions : Fax free." http://www.f-secure.com/v-descs/mosquito.shtml.

[39] "3G security; security architecture," tech. rep., ETSI European Telecommunications Standards Institution, Septemper 2005.

[40] P. Howard, "3g security overview," March 2000. IIR Fraud and Security Conference.

[41] "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)," tech. rep., IETF Internet Engineering Task Force, September 2002. RFC 3310.

[42] G. M. KIEN, "An introduction to access security in umts," *IEEE Wireless Communications*, February 2004.

[43] "The MILENAGE Algorithm Set: An Example Algorithm for the 3GPP; Authentication and Key Generation Functions," tech. rep., ETSI SAGE Task Force for 3GPP, December 2000. Security Algorithms Group of Experts (SAGE) (SP-010144 version 1.0).

[44] "Specification of the 3gpp confidentiality and integrity algorithms; document 1: f8 and f9 specifications," tech. rep., ETSI European Telecommunications Standards Institution, September 2005. (Universal Mobile Telecommunications System (UMTS); 3GPP TS 35.201 version 6.1.0 Release 6).

[45] "Cryptographic Algorithm Requirements," tech. rep., 3GPP 3rd Generation Partnership Project, June 2001. 3G Security, Release 4.

[46] Fort George G.and Meade, "Department of Defense Password Management Guidline," tech. rep., Computer Security Center, Department of Defense, April 1985.

[47] C. Farkas and G. Ziegler, "Anonymity and accountability in self-organising electronic communities," Proceedings of the ACM workshop on privacy in the Electronic society, pp. 81–90, November 2002.

[48] Shona Hunter, "A Critical Analysis of Approaches to the Concept of Social Identity in Social Policy," tech. rep., SAGE Publications, 2003.

[49] J. Camp, "Identity in digital government," tech. rep., Kennedy School of Government, Harvard University, 2003. A report of the 2003 Civic Scenario Workshop.

[50] T. Nabeth, "Understanding the identity concept in the context of digital social environments," tech. rep., The Centre for Advanced Learning Technologies, January 2005. Network of Excellence FIDIS (Future of the Identity in the Information Society), a project of the 6th Framework programme of the European commission.

[51] "Liberty ID-WSF a Web Service Framework," tech. rep., Liberty Alliance, 2004.

[52] "Liberty ID-FF Architecture Overview," tech. rep., Liberty Alliance, April 2003.

[53] "Liberty ID-FF Protocols and Schema Specification," tech. rep., Liberty Alliance, April 2003.

[54] "Liberty Alliance Project Whitepaper: Personal Identity," tech. rep., Liberty Alliance, June 2006.

[55] M. Hillenbrand, J. Gotze, J. Muller, and P. Muller, "A single sign-on framework for web-services-based distributed applications," *Proceedings of the 8th International Conference on Telecommunications, ConTEL 2005*, vol. 1, pp. 273 – 279, 2005.

[56] "Microsoft .NET Passport Review Guide," tech. rep., Microsoft, January 2004.

[57] "The Kerberos Network Authentication Service (V5)," tech. rep., IETF Internet Engineering Task Force, September 1993. RFC 1510.

[58] B. Neuman and T. Ts'o, "Kerberos. an authentication service for computer networks," *IEEE Communications Magazine*, pp. 33 – 38, 1994.

[59] J. T. Kohl, "The evolution of the kerberos authentication service," *In Proceeding of the Spring 1991 EurOpen Conference*, 1991.

[60] S. M. Bellovin and M. Merritt, "Limitations of the kerberos authenication system," *Computer Communication Review*, pp. 119–132, October 1990.

[61] "Liberty Authentication Context Specification," tech. rep., Liberty Alliance, April 2003.

[62] "Transport Layer Security Protocol (TLS) ," tech. rep., IETF Internet Engineering Task Force, January 1999. RFC 2246.

[63] *The Wireless Application Protocol Architecture Specification.* http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html, 1998.

[64] I. M. Kalden, R. and M. Meyer, "Wireless internet access based on gprs," *IEEE Personal Communications*, vol. 7, no. 2, pp. 8 – 18, 2000.

[65] J. NTT DoCoMo, "Here comes the wallet phone," *IEEE Spectrum*, November 2005.

[66] W. Itani and A. Kayssi, "J2me application-layer end-to-end security for m-commerce," *Journal of Network and Computer Applications*, vol. 27, pp. 13 – 32, January 2004.

[67] B. Pfitzmann, "Privacy in enterprise identity federation - policies for liberty 2 single signon," LNCS, 2003. 3rd Workshop on Privacy Enhancing Technologies (PET 2003).

[68] B. Pfitzmann, "Privacy in enterprise identity federation - policies for liberty 2 single signon," tech. rep., Elsevier Information Security Technical Report (ISTR), 2004.

[69] T. Fleury, J. Basney, and V. Welch, "Single sign-on for java web start applications using myproxy," *Proceedings of the 13th ACM Conference on Computer and Communications Security Co-Located Workshops*, November 2006.

[70] H. R., P. W., W. Ford, and S. D., "Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile," tech. rep., IETF Internet Engineering Task Force, 2002. Request For Comments 3280.

[71] M. Papazoglou and D. Georgakopoulos, "Service oriented computing," *Communications of the ACM*, vol. 46, pp. 8 – 25, October 2003.

[72] Z. Stojanovic and A. Dahanayake, "Service oriented software system engineering: Challenges and practices," *Idea Group Publishing*, 2005.

[73] J. Li and A. H. Karp, "Access control for the services oriented architecture," *Proceedings of the 13th ACM Conference on Computer and Communications Security Co-Located Workshops*, November 2007.

[74] "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0," tech. rep., The Advancement of Structured Information Standards (OASIS), March 2005. OASIS Standard.

[75] P. Felix and C. Ribeiro, "A scalable and flexible web services authentication model," *Proceedings of the 13th ACM Conference on Computer and Communications Security Co-Located Workshops*, November 2007.

[76] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030 – 1044, 1985.

[77] M. Hillenbrand, J. Gotze, J. Muller, and P. Muller, "A Single Sign-On Framework for Web-Services-based Distributed Applications," University of Kaiserslautern.

[78] A. Pashalidis and C. J. Mitchell, "Using GSM/UMTS for single sign-on," in *Proceedings of SympoTIC '03*, (Bratislava, Slovakia), pp. 138–145, IEEE Press, October 2003. Joint IST Workshop on Mobile Future and Symposium on Trends in Communications.

[79] V. Khu-smith and C. J. Mitchell, ""using gsm to enhance e-commerce security"," in *WMC '02: Proceedings of the 2nd international workshop on Mobile commerce*, ACM Press, 2002.

[80] N. Asokan and L. Tarkkala, "Issues in initializing security," in *IEEE International Symposium on Signal Processing and Information Technology*, IEEE, 2005.

[81] A. Balyan, K. Loganathan, and S. Sripathi, "Security architecture for ip-based multi-service networks," *Bell Labs - Technical Journal*, vol. 11, no. 1, pp. 59 – 78, 2006.

[82] M. Sher and T. Magedanz, "Security associations management (sam) model for ip multimedia system (ims)," *Network Control and Engineering for QoS, Security and Mobility*, vol. 229, pp. 311 – 325, 2007.

[83] I. Sommerville, *Software Engineering*. Addison Wesley, 2004.

[84] C. A. Meadows, "Formal verification of cryptographic protocols: A survey," *Springer-Verlag*, vol. ASIACRYPT'94, pp. 135–150, 1995.

[85] C. A. Meadows, "The nrl protocol analyzer: An overview," *Journal of Logic Programming*, vol. 26, pp. 113–131, 1996.

[86] D. Longley and S. Rigby, "An automatic search for security flaws in key management schemes," *Computers and Security*, vol. 11, no. 1, pp. 75–89, 1992.

[87] R. Anderson, "Why information security is hard an economic perspective," 2001.

[88] D. Gollmann, *Computer Security*. John Wiley and Sons, 1999.

[89] L. C. Aiello and F. Massacci, "Verifying security protocols as planning in logic programming," *Transactions on Computational Logic*, 2001.

[90] "Liberty ID-WSF Security Mechanisms," tech. rep., Liberty Alliance, 2003.

[91] "Liberty ID-WSF Authentication Service and Single Sign-On Service," tech. rep., Liberty Alliance, May 2003.

[92] "Digital cellular telecommunications system (Phase 2+)," tech. rep., ETSI European Telecommunications Standards Institution, June 2001. Security aspects (GSM 02.09 version 8.0.1).

[93] "Digital cellular telecommunications system (phase 2+)," tech. rep., ETSI European Telecommunications Standards Institution, July 2001. Security related network functions (GSM 03.220 version 8.1.0).

[94] V. L. Voydock and S. T. Kent, "Security mechanisms in high-level network protocols," *Computing Surveys*, vol. 15, pp. 135–171, June 1983.

[95] "Generic bootstrapping architecture," tech. rep., ETSI European Telecommunications Standards Institution, June 2005. UMTS, Generic Authentication Architecture.

[96] "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0," tech. rep., The Advancement of Structured Information Standards (OASIS), March 2005. OASIS Standard.

[97] "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0," tech. rep., The Advancement of Structured Information Standards (OASIS), March 2005. OASIS Standard.

[98] "WAP Architecture," tech. rep., WAP Forum, July 2001. Wireless Application Protocol, Architecture Specification, WAP-210-WAPArch-20010712.

[99] "SAML Implementation Guidelines," tech. rep., The Advancement of Structured Information Standards (OASIS), August 2004. OASIS Standard.

[100] D. Welch, "Wireless security threat taxonomy," *IEEE Proceedings: Workshop on Information Assurance*, pp. 76 – 83, June 2003.

[101] C. Gutirrez, E. Fernndez-Medina, and M. Piattini, "Web services enterprise security architecture: A case study," *Proceedings of the 2005 ACM Workshop on Secure Web Services*, November 2005.

[102] B. Garbinato and I. Rickebusch, "Orchestrating fair exchanges between mutually distrustful web services," *Proceedings of the 13th ACM Conference on Computer and Communications Security Co-Located Workshops*, November 2006.

[103] T. S. Masashi Morioka, Yoshifumi Yonemoto and M. Etoh, "Scalable security description framework for mobile web services," *IEEE*, pp. 804 – 808, 2003.

[104] T. GroB, "Security analysis of the saml single sign-on browser/artifact profile," *IEEE Computer Society*, December 2003. 19th Annual Computer Security Applications Conference.

[105] "SSTC Response to Security Analysis of the SAML Single Sign-on Browser/Artifact Profile," tech. rep., The Advancement of Structured Information Standards (OASIS), July 2005. OASIS Standard.

[106] J. Wang and D. D. Vecchio, "Extending the security assertion markup language to support delegation for web services and grid services," *IEEE Computer Society*, 2005. Proceedings of the IEEE International Conference on Web Services (ICWS05).

[107] Y.-Y. JChan, "Weakest link attack on single sign-on and its case in saml v2.0 web sso," *Springer Verlag, Heidelberg*, 2006. ICCSA 2006: International Conference on Computational Science and Its Applications, 3982 LNCS.

[108] S.-T. Cheng, J.-P. Liu, J.-L. Kao, and C.-M. Chen, "A new framework for mobile web services," *IEEE Proceedings: Symposium on Application and the Internet (SAINT'02w)*, 2002.

[109] W. Zahreddine and Q. H. Mahmoud, "An agent-based approach to composite mobile web services," *IEEE Proceedings: The 19th International Conference on Advanced Information Networking and Application (AINA'05)*, 2005.

[110] "Access to network application functions using hypertext transfer protocol over transport layer security," tech. rep., ETSI European Telecommunications Standards Institution, June 2005. UMTS, Generic Authentication Architecture.

[111] "Network domain security (nds); ip network layer security," tech. rep., ETSI European Telecommunications Standards Institution, December 2005. 3GPP TS 33.210, Release 7.

[112] "Diameter Base Protocol," tech. rep., IETF Internet Engineering Task Force, September 2003. RFC 3588.

[113] R. Ford, "Managing retail service businesses for the 1990s: Marketing aspects," *European Management Journal*, vol. 8, pp. 58–66, 1990.

[114] "Interworking of Liberty Alliance ID-FF, ID-WSF and Generic Authentication Architecture," tech. rep., 3GPP 3rd Generation Partnership Project, July 2005. 3GPP TR 33.980; Technical Specification Group Services and System Aspect, Release 4.

[115] J. A. MacDonald, W. G. Sirett, and C. J. Mitchell, "Overcoming channel bandwidth constraints in secure SIM applications," in *Security and Privacy in the Age of Ubiquitous Computing*, Springer Science and Business Media, 2005.

[116] S. Krishna, *Web Services Framework and Assertion exchange using SAML.* W3C, http://www.w3.org, 2001.

[117] C. Block and A. C. Wagner, *MIDP 2.0 Style Guide.* London: Addison-Wesley, 2003.

[118] J. Snell, D. Tidwell, and P. Kulchenko, *Programming Web Services with SOAP.* O'Reilly, 2002.

[119] Sun Microsystems, http://java.sun.com/products, *Wireless Toolkit, Version 2.1,*, 2003.

[120] R. Clarke, "Introduction to dataveillance and information privacy, and definitions of terms," August 1997.

[121] "The Platform for Privacy Preferences 1.0 Specification," tech. rep., W3C, February 2002. W3C Recommendation.

[122] P. Kolari, L. Ding, and L. Kagal, "Enhancing web privacy protection through declarative policies," 2005. Proceeding of the Sixth IEEE International Workshop on Policies for Distributed System and Networks.

[123] "Web Services Architecture," tech. rep., W3C, February 2004. W3C Working Group Note.

[124] "Web Services Architecture Requirements," tech. rep., W3C, February 2004. W3C Working Group Note.

[125] C. Adams and K. Barbieri, "Privacy enforcement in eservices environments," *In Privacy Protection for E-Services*, 2006.

[126] G. O. M. Yee, "A privacy controller approach for privacy protection in web services," *Proceedings of the 13th ACM Conference on Computer and Communications Security Co-Located Workshops*, November 2007.

[127] B. Carminati, E. Ferrari, and P. C. Hung, "The value of privacy, exploring privacy issues in web services discovery agencies," *IEEE Security and Privacy*, September 2005.

[128] E. J. Chang, F. K. Hussain, and T. S. Dillion, "Fuzzy nature of trust and dynamic trust modeling in service oriented environments," *Proceedings of the 2005 ACM Workshop on Secure Web Services*, November 2005.

[129] A. H. Anderson, "A comparison of two privacy policy languages: Epal and xacml," *Proceedings of the 13th ACM Conference on Computer and Communications Security Co-Located Workshops*, November 2006.

[130] IBM, "Enterprise privacy authorization language (epal)," tech. rep., http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/, 2003. Version 1.2.

[131] T. Moses, "extensible access control markup language (xacml)," tech. rep., OASIS, February 2005. Version 2.0.

[132] T. Moses, "Privacy policy profile of xacml v2.0," tech. rep., OASIS, February 2005.

[133] H.-H. Lee and M. Stamp, "P3p privacy enhancing agent," *Proceedings of the 13th ACM Conference on Computer and Communications Security Co-Located Workshops*, November 2006.

[134] "Support for subsriber certificates," tech. rep., ETSI European Telecommunications Standards Institution, June 2005. UMTS, Generic Authentication Architecture.

[135] C. J. Mitchell, "Timestamps and authentication protocols," tech. rep., Royal Holloway, University of London, February 2005.

[136] N. Heintze and J. D. Tygar, "Timed models for protocol security," Tech. Rep. CMU-CS-92-100, 1992.

[137] B. C. Neuman and S. G. Stubblebine, "A note on the use of timestamps as nonces," *Operating System Review*, vol. 27, pp. 10–14, April 1993.

[138] E.-J. Yoon and K.-Y. Yoo, "Efficient mutual authenticaion scheme with smart card," *Lecture Notes in Artificial Intelligence*, vol. 4088, pp. 813–818, 2006.

[139] D. E. Denning and G. M. Sacco, "Timestamps in key distripution protocols," *Communications of the ACM*, vol. 24, pp. 533–536, August 1981.

Appendix

```
package sso;

import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
import javax.microedition.io.*;
import javax.microedition.media.*;
import javax.microedition.media.control.*;

import org.ksoap.ClassMap;
import org.ksoap.Soap;
import org.ksoap.SoapEnvelope;
import org.ksoap.transport.HttpTransport;
import org.kxml.parser.XmlParser;
import org.ksoap.*;
import org.ksoap.transport.HttpTransport;
import org.kxml.parser.XmlParser;
import stuGenSP.ServiceProviderService_Stub;
import stuGeneretedNAFService.NAF_IdPService_Stub;
import stubGeneretedstuMobileOperatorService.MobileOperatorService_Stub;

import java.io.*;

public class MIDlet1
  extends MIDlet
  implements CommandListener,PlayerListener {
 private static MIDlet1 instance;

 private Displayable1 displayable = new Displayable1();

 private List previousMenu;
 private Player player;
 private List musicMenu;
 private List oneBeforePrevious;
 private List blockbuster;

 Display display = null;

 // a menu with items
 List mainMenu = null; // main menu

 // textbox
 TextBox input = null;

 // command
 static final Command backCommand = new Command("Back", Command.BACK, 0);
```

```
 static final Command mainMenuCommand = new Command("Main",
Command.SCREEN,
    1);

 static final Command exitCommand = new Command("Exit", Command.STOP, 2);

 static final Command orderCommand = new Command("Order", Command.OK, 3);

 static final Command cancelCommand = new Command("Cancel",
Command.CANCEL,
    4);
static final Command playCommand = new Command("Play Now", Command.OK, 5);
 String currentMenu = null;

 Object ServiceProviderService;
 int ServiceProviderServiceInt;

 String t1;

 String t2;

 String t3;
 Image im = null;
 String url = "http://www.corej2me.com/midpbook_v1e1/ch14/duke.png";
 String moRespone;

 String tid;
 String ks;
 Player audioplayer;
 String sk;

 String rad;

 String ut;
 String sput;
 String service = "";

 int i = 0;
 String respone;
 String soapReqpMesg2 =
    "<soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\""
    + "xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\""
    + "xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\">"
```

```
    + "<soapenv:Body><ns1:CallService
soapenv:encodingStyle=\"http://schemas.xmlsoap.org/soap/encoding/\""
    + "xmlns:ns1=\"http://soapinterop.org/\">"
    + "<ns1:arg0 xsi:type=\"soapenc:string\""
    +
    "xmlns:soapenc=\"http://schemas.xmlsoap.org/soap/encoding/\">222</ns1:arg0>"
    + "<ns1:arg1 xsi:type=\"soapenc:string\"
xmlns:soapenc=\"http://schemas.xmlsoap.org/soap/encoding/\">"
    + "test</ns1:arg1><ns1:arg2 xsi:type=\"soapenc:string\"
xmlns:soapenc=\"http://schemas.xmlsoap.org/soap/encoding/\">"
    + "test</ns1:arg2><ns1:arg3 xsi:type=\"soapenc:string\"
xmlns:soapenc=\"http://schemas.xmlsoap.org/soap/encoding/\">"
    + "test</ns1:arg3></ns1:CallService></soapenv:Body></soapenv:Envelope>";

 String soapReqpMesg3 =
    "<soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\""
    + "xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\""
    + "xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\">"
    + "<soapenv:Body><ns1:CallService
soapenv:encodingStyle=\"http://schemas.xmlsoap.org/soap/encoding/\""
    + "xmlns:ns1=\"http://soapinterop.org/\">"
    + "<ns1:arg0 xsi:type=\"soapenc:string\""
    +
"xmlns:soapenc=\"http://schemas.xmlsoap.org/soap/encoding/\">Football</ns1:arg0>"
    + "<ns1:arg1 xsi:type=\"soapenc:string\"
xmlns:soapenc=\"http://schemas.xmlsoap.org/soap/encoding/\">"
    + "test</ns1:arg1><ns1:arg2 xsi:type=\"soapenc:string\"
xmlns:soapenc=\"http://schemas.xmlsoap.org/soap/encoding/\">"
    + "test</ns1:arg2><ns1:arg3 xsi:type=\"soapenc:string\"
xmlns:soapenc=\"http://schemas.xmlsoap.org/soap/encoding/\">"
    + "test</ns1:arg3></ns1:CallService></soapenv:Body></soapenv:Envelope>";

 // constructor.
 public MIDlet1() {
 }

 /**
 * Start the MIDlet by creating a list of items and associating the exit
 * command with the list.
 */
public void startApp() throws MIDletStateChangeException {
 display = Display.getDisplay(this);

 try {
   mainMenu = new List("AspiData Shopping Mall", Choice.IMPLICIT);
```

```
mainMenu.append("Connecting to the Shopping Mall", null);
mainMenu();
NAF_IdPService_Stub naf = new NAF_IdPService_Stub();
MobileOperatorService_Stub mo = new MobileOperatorService_Stub();
ServiceProviderService_Stub sp = new ServiceProviderService_Stub();
System.out.println(
   "Connecting with Identity Provider ; IMPI number : 1111\n");
naf.reqAccess("1111");
System.out.println(
   "Identity Provider to user ; initating a new Bootstapping Procedure\n");
System.out.println(
   "User to BSF ; Intiating the BSP; IMPI number : 1111\n");
moRespone = mo.reqAccess("1111");
tid = moRespone.substring(29);


System.out.println("Retrive B-TID from BSF");
System.out.println("B-TID : "+tid+" ");

System.out.println("User to Identity Provider forwading the B-TID for login : "+ tid);
rad = naf.reqLogin(tid);

System.out.println("Identity Provider to User ; Challenge(RAND) : "+ rad);
sk = moRespone.substring(5, 19);
    int num1 = sk.hashCode();
    int num2 = rad.hashCode();
    String LoginChallengeGenerated = Integer.toHexString(num1 - num2);
    System.out.println("Generating Challenge Response :" + LoginChallengeGenerated);
    System.out.println("User to Identity Provider ; Challenge Response : "+
LoginChallengeGenerated);


    ut = naf.login(LoginChallengeGenerated, tid);

    System.out.println("Identity Provider to User ; UserToken : "+ ut);
    mainMenu.delete(0);
    mainMenu.append("Blockbuster Sports", null);
    mainMenu.append("Virgin MegaStore", null);
    mainMenu.append("Amazon", null);
    mainMenu.append("WaterStone", null);
    mainMenu.append("M&S", null);
    mainMenu.append("Vodafone Live!", null);
    mainMenu.append("UEFA", null);
    mainMenu.append("Playboy", null);
    mainMenu.append("BBC", null);
    mainMenu.append("easyJet", null);
```

```
mainMenu.append("888.com", null);
mainMenu.addCommand(exitCommand);
mainMenu.setCommandListener(this);

mainMenu();
  // System.out.println(sk + "sk");



  //System.out.println("ddddddddddd" + naf.reqService("222", ut));

  // System.out.println("ee"+t3+"ww");
  // mMessageItem.setText("oki");*/
 }
 catch (Exception ioe) {
  System.out.println("-->"+ioe.toString());
 }

}
public void pauseApp() {
 display = null;
 mainMenu = null;
 input = null;
}

public void destroyApp(boolean unconditional) {
 notifyDestroyed();
}

// main menu
void mainMenu() {
 display.setCurrent(mainMenu);
 currentMenu = "Main";
}
void intermediateMenu(List menu,String name){
 display.setCurrent(menu);
 currentMenu = name;
}

/**
 * a generic method that is called when any of the items on the list are
 * selected.
```

```
 */
public void prepare() {
  input = new TextBox("Enter some text: ", "", 5, TextField.ANY);
  input.addCommand(backCommand);
  input.setCommandListener(this);
  input.setString("");
  currentMenu = "text";
  this.oneBeforePrevious = this.previousMenu;
  this.previousMenu = (List) display.getCurrent();
  display.setCurrent(input);
 }

 public void createChannel4Menu() {
  try{
    System.out.println("\nCalling Blockbuster Sports");
    NAF_IdPService_Stub naf = new NAF_IdPService_Stub();
    System.out.println("User to Identity Provider requesting the service from :
Blockbuster Sports SPID : Blockbuster 4 , UT : "+ut);
    String sptut = naf.reqService("Blockbuster Sports", ut);

    sput = sptut.substring(sptut.indexOf("@"),sptut.length());
    System.out.println("Identity Provider to User Service Provider specific UserToken :
"+sput);
  List channel4Menu = new List("Blockbuster Sports",
                    Choice.IMPLICIT);

  channel4Menu.append("Football £25", null);
  channel4Menu.append("Cricket £35", null);

  channel4Menu.addCommand(backCommand);

  channel4Menu.setCommandListener(this);
  this.oneBeforePrevious = this.previousMenu;
  this.previousMenu = (List) display.getCurrent();
  display.setCurrent(channel4Menu);
  blockbuster = channel4Menu;
}catch(Exception ex){
  System.out.println(ex.toString());
}
}

public void createMusicMenu() {
  musicMenu = new List("Virgin MegaStore",
                    Choice.IMPLICIT);
```

```
musicMenu.append("Classic", null);
musicMenu.append("Rap", null);

musicMenu.addCommand(backCommand);

musicMenu.setCommandListener(this);
this.oneBeforePrevious = this.previousMenu;
this.previousMenu = (List) display.getCurrent();
 display.setCurrent(musicMenu);

}

/**
 * Test item1.
 */
public void channel4() {
 this.createChannel4Menu();
 currentMenu = "Sport";
}

/**
 * Test item2.
 */
/*public void createMenu(int h)
 {
 if(h==222)
 {
 this.createChannel4Menu();
 currentMenu = "channel4";
 }
 else
 {
 this.createGreenwich();
 currentMenu = "greenwich";
 }
 }*/

public void music() {
 // prepare();
 this.createMusicMenu();
 currentMenu = "Music";
}

public void chosenSP(String ServiceProviderServiceStr) {

 System.out.println("www");
```

```
ByteArrayInputStream bis = new ByteArrayInputStream(
   ServiceProviderServiceStr
   .getBytes());
InputStreamReader reader = new InputStreamReader(bis);
XmlParser xp;
SoapEnvelope envelope = new SoapEnvelope(new ClassMap(Soap.VER11));

try {
 xp = new XmlParser(reader);
 envelope.parse(xp);
 ServiceProviderService = envelope.getBody();
 HttpTransport k = new HttpTransport(
    "http://localhost:8080/axis/services/ServiceProviderService",
    "uurn:ServiceProviderService#CallService");
 //System.out.println(k.call(ServiceProviderService).toString());

 k.call(ServiceProviderService);

}
catch (Exception e1) {
 System.out.println("-->"+e1.toString());
 // TODO Auto-generated catch block
 e1.printStackTrace();
  }

 }

 public void CallPaymentService(){
  try{
   NAF_IdPService_Stub naf = new NAF_IdPService_Stub();
   System.out.println("User to Identity Provider requesting the service for Finacial SPID
: Finacial , UT : " +
                ut);
   String sptut = naf.reqService("Finacial", ut);

   sput = sptut.substring(sptut.indexOf("@"), sptut.length());
   System.out.println(
     "Identity Provider to User Finacial Service Provider specific UserToken : " +
     sput);

   System.out.println("User to Finacial Service Provider : Invoice");
  }catch(Exception ex){
   System.out.println("-->"+ex.toString());
  }

}
```

```
 public void ShowPaymentMethods(){

   List paymentMenu;

   if(currentMenu.endsWith("FootballInvoice") ||
currentMenu.endsWith("CricketInvoice")){
     paymentMenu = new List("Blockbuster Sports",
                            Choice.IMPLICIT);
   }else if(currentMenu.endsWith("RapInvoice") ||
currentMenu.endsWith("ClassicInvoice")){
     paymentMenu = new List("Virgin MegaStore",
                            Choice.IMPLICIT);
   }else{
     paymentMenu = new List("Payment Methods",
                            Choice.IMPLICIT);
   }


     paymentMenu.append("Pay from your Mobile Bill", null);
     paymentMenu.append("Barclays Payment Service", null);

     paymentMenu.addCommand(backCommand);

     paymentMenu.setCommandListener(this);
     this.oneBeforePrevious = this.previousMenu;

     display.setCurrent(paymentMenu);

 }
 /**
  * Test item3.
  */
 public void vh1() {
   prepare();
   currentMenu = "vh1";
 }
 public void DisplayFootBall(){
     currentMenu = "Football";
     this.playVideo("http://localhost/site/Football.mpg");
 //    Form footballclip = new Form("");
 //
 //        footballclip.append("");
 //
 //        footballclip.addCommand(backCommand);
```

```
//        footballclip.setCommandListener(this);
//
//        System.out.println("Service Provicer to User : Service Delivery");
//        Image im =
this.getImage("http://upload.wikimedia.org/wikipedia/commons/thumb/7/7f/Generic_foot
ball.png/180px-Generic_football.png");
//        ImageItem ii = new ImageItem(null, im, ImageItem.LAYOUT_DEFAULT, null);
//
//        if (footballclip.size() != 0)
//          footballclip.set(0, ii);
//        else // Append the image to the empty form
//          footballclip.append(ii);
//
//        this.oneBeforePrevious = this.previousMenu;
//        //this.previousMenu = (Form)display.getCurrent();
//        currentMenu = "FootBall";
//
//        display.setCurrent(footballclip);

    }
    private void playVideo(String URL){
      try {
          VideoControl vc;
          defplayer();
          // create a player instance
          player = Manager.createPlayer(URL);
          player.addPlayerListener(this);
          // realize the player
          player.realize();
          vc = (VideoControl)player.getControl(
                  "VideoControl");
          if(vc != null) {
            Item video = (Item)vc.initDisplayMode(
                        vc.USE_GUI_PRIMITIVE, null);
            Form v = new Form("Playing Video...");
            v.addCommand(backCommand);
            v.setCommandListener(this);
            StringItem si = new StringItem("Status: ",
                        "Playing...");
            v.append(si);
            v.append(video);
            this.oneBeforePrevious = this.previousMenu;


            display.setCurrent(v);
          }
```

```
      player.prefetch();
      player.start();
    }
    catch(Throwable t) {
      reset();
    }

}
```

```
 public void DisplayCricket(){
   currentMenu = "Cricket";
  this.playVideo("http://localhost/site/shoaib1.mpg");
//   Form cricketClip = new Form("");
//
//       cricketClip.append("");
//
//       cricketClip.addCommand(backCommand);
//       cricketClip.setCommandListener(this);
//
//       System.out.println("Service Provicer to User : Service Delivery");
//       Image im =
this.getImage("http://www.gametronik.com/site/rubriques/megadrive/Jeux/Shane%20Wa
rne%20Cricket/Shane%20Warne%20Cricket.png");
//       ImageItem ii = new ImageItem(null, im, ImageItem.LAYOUT_DEFAULT, null);
//
//       if (cricketClip.size() != 0)
//         cricketClip.set(0, ii);
//       else // Append the image to the empty form
//         cricketClip.append(ii);
//
//       this.oneBeforePrevious = this.previousMenu;
//       //this.previousMenu = (Form)display.getCurrent();
//       currentMenu = "Cricket";
//
//       display.setCurrent(cricketClip);

 }
```

```
private Image getImage(String url) {

   Image im = null;
   try {
    InputStream iStrm = (InputStream) Connector.openInputStream(url);
    ByteArrayOutputStream bStrm = new ByteArrayOutputStream();
    int ch;
    while ( (ch = iStrm.read()) != -1)
```

```
    bStrm.write(ch);

    // Place into image array
   byte imageData[] = bStrm.toByteArray();
   // Create the image from the byte array
   im = Image.createImage(imageData, 0, imageData.length);

}
   catch (Exception ex) {
    ex.printStackTrace();
   }
   finally {
    // Clean up
    //   if (iStrm != null)
    //     iStrm.close();
   }

   return (im == null ? null : im);

  }

public void getFootBall() {
  try{
     System.out.println("User Calling FootBall Clip from Blockbuster Sports by sending
SPUT :"+sput );
     //chosenSP(soapReqpMesg3);
     System.out.println("Response from Sport to User with the Invoice" );

     Form footballclip = new Form("Blockbuster Sports");

     footballclip.append("duration 2 min");
     footballclip.append("\nInvoice For Football Clip");
     footballclip.append("\nPrice £ 25.00");
     footballclip.append("If u wish to continue please confirm Order now");
     footballclip.addCommand(backCommand);
     footballclip.addCommand(orderCommand);
     footballclip.setCommandListener(this);

    /*  Image im = this.getImage();
ImageItem ii = new ImageItem(null, im, ImageItem.LAYOUT_DEFAULT, null);

if (footballclip.size() != 0)
  footballclip.set(0, ii);
else

  // Append the image to the empty form
```

```
        footballclip.append(ii);
*/
    this.oneBeforePrevious = this.previousMenu;
    this.previousMenu = (List) display.getCurrent();
    currentMenu = "FootballInvoice";

    display.setCurrent(footballclip);
   }catch(Exception ex){
    System.out.println(ex.toString());
   }
  }

  public void GetCricket(){
   try{
    System.out.println("User Calling Cricket Clip from Blockbuster Sports by sending
SPUT :"+sput );
    //chosenSP(soapReqpMesg3);
    System.out.println("Response from Blockbuster Sports to User with the Invoice" );

    Form cricketclip = new Form("Blockbuster Sports");
    cricketclip.append("duration 2 min");
    cricketclip.append("\nInvoice for Cricket Clip");
    cricketclip.append("\nPrice £ 35.00");
    cricketclip.append("If u wish to continue please confirm Order now");
    cricketclip.addCommand(backCommand);
    cricketclip.addCommand(orderCommand);
    cricketclip.setCommandListener(this);


    this.oneBeforePrevious = this.previousMenu;
    this.previousMenu = (List) display.getCurrent();
    currentMenu = "CricketInvoice";

    display.setCurrent(cricketclip);
   }catch(Exception ex){
    System.out.println(ex.toString());
   }
  }

  public void GetClassic(){
   try{
    System.out.println("User Calling Classic Music from Virgin MegaStore by sending
SPUT : @26334cf8" );
    //chosenSP(soapReqpMesg3);
    System.out.println("Response from Music to User with the Invoice" );
```

```
Form classicclip = new Form("Virgin MegaStore");
classicclip.append("duration 3 min");
classicclip.append("\nInvoice for Music");
classicclip.append("\nPrice £ 15.00");
classicclip.append("If u wish to continue please confirm Play now");
classicclip.addCommand(backCommand);
classicclip.addCommand(playCommand);
  classicclip.setCommandListener(this);


  this.oneBeforePrevious = this.previousMenu;
  this.previousMenu = (List) display.getCurrent();
  currentMenu = "ClassicInvoice";

  display.setCurrent(classicclip);
 }catch(Exception ex){
  System.out.println(ex.toString());
 }

}

private void reset() {
    player = null;
  }

private void stopPlayer() {

    try {

     defplayer();

    }
    catch(MediaException me) {

    }
    reset();
  }

/**
 * Handle events.
 */
public void commandAction(Command c, Displayable d) {
 String label = c.getLabel();
 if (label.equals("Exit")) {
  destroyApp(true);
```

```
}
else if (label.equals("Back")) {

  this.previousMenu = this.oneBeforePrevious;

  if (currentMenu.equals("Sport")
      || currentMenu.equals("Music")
      || currentMenu.equals("vh1")) {
      // go back to menu
      mainMenu();
      System.out.println("Goto main menu");
     }
     else if(currentMenu.equals("Cricket") || currentMenu.equals("Football")){

      this.stopPlayer();
      intermediateMenu(blockbuster,"Sports");
     }
     else if(currentMenu.equals("Music - Classic") || currentMenu.equals("Music -
Rap")){
      try{
        this.audioplayer.stop();
      }catch(Exception ex){
        System.out.println(ex.toString());
       }
      intermediateMenu(musicMenu,"Music");
     }

     else if(currentMenu.equals("CricketInvoice") ||
currentMenu.equals("FootballInvoice")){

      intermediateMenu(blockbuster,"Sports");
    }
    else if(currentMenu.equals("ClassicInvoice") || currentMenu.equals("RapInvoice")){

      intermediateMenu(musicMenu,"Music");
    }


     else {
       mainMenu();
       display.setCurrent(this.previousMenu);
     }
     // }
}
else if(label.equals("Order")){
    this.ShowPaymentMethods();
```

```
}
else if(label.equals("Play Now")){
  this.ShowPaymentMethods();
}
else {
 List down = (List) display.getCurrent();

 switch (down.getSelectedIndex()) {
case 0:

 if (down.getString(0).compareTo("Blockbuster Sports") == 0) {
  channel4();
  //chosenSP(soapReqpMesg3);
 }
 else if (down.getString(0).compareTo("Football £25") == 0) {
  service = "Football";
  this.getFootBall();

  //invokeWebService();
 }
 else if (down.getString(0).compareTo("Classic") == 0) {
  service = "Classic";
  System.out.println("Classic");
   this.GetClassic();

 }
else if(down.getString(0).compareTo("Pay from your Mobile Bill") == 0) {

  if(service.compareTo("Football") == 0){
    this.DisplayFootBall();
   }else if(service.compareTo("Cricket") == 0){
    this.DisplayCricket();
   }else if(service.compareTo("Classic") == 0){
    this.playMusic();
 }else if(service.compareTo("Rap") == 0){
  this.playMusic();
}
  }


  break;
case 1:

  if (down.getString(0).compareTo("Blockbuster Sports") == 0) {
```

```
  music();
  //chosenSP(soapReqpMesg2);
 }
 else if (down.getString(0).compareTo("Classic") == 0) {
  service = "Rap";
  System.out.println("Rap");
  this.GetClassic();
        }
        else if (down.getString(0).compareTo("Football £25") == 0) {
         service = "Cricket";
         this.GetCricket();
//          i = 1;
//          invokeWebService();

        }
        else if(down.getString(0).compareTo("Pay from your Mobile Bill") == 0) {

         this.CallPaymentService();
         if(service.compareTo("Football") == 0) {
           this.DisplayFootBall();
          }else if(service.compareTo("Cricket") == 0){
           this.DisplayCricket();
           }else if(service.compareTo("Classic") == 0){
            this.playMusic();

          }else if(service.compareTo("Rap") == 0){
           this.playMusic();
        }



       }


        break;


 }
  }
 }

public void playMusic(){
  try {

    Form musicClipForm = new Form("");
```

```
musicClipForm.addCommand(backCommand);
musicClipForm.setCommandListener(this);

System.out.println("Service Provicer to User : Service Delivery");

    this.oneBeforePrevious = this.previousMenu;



if(service.compareTo("Classic")==0){
 currentMenu = "Music - Classic";
 musicClipForm.append("Now You are listening to Classic");
  this.audioplayer = Manager.createPlayer("http://localhost/site/barebear.wav");
  this.audioplayer.start();
}else{
 currentMenu = "Music - Rap";
  musicClipForm.append("Now You are listening to Rap");
  this.audioplayer = Manager.createPlayer("http://localhost/site/scooby.wav");
   this.audioplayer.start();

  }
  display.setCurrent(musicClipForm);
 }
 catch (Exception e) {
  System.out.println("-->"+e.toString());

 }


}

private void defplayer() throws MediaException {
   if (player != null) {
     if(player.getState() == Player.STARTED) {
       player.stop();
     }
     if(player.getState() == Player.PREFETCHED) {
```

```
      player.deallocate();
    }
    if(player.getState() == Player.REALIZED ||
            player.getState() == Player.UNREALIZED) {
      player.close();
    }
  }
  player = null;
}
```

```
private Image getImage() {

 Thread t = new Thread() {
  public void run() {
   try {

     InputStream iStrm = (InputStream) Connector.openInputStream(url);
     ByteArrayOutputStream bStrm = new ByteArrayOutputStream();
     int ch;
     while ( (ch = iStrm.read()) != -1)
       bStrm.write(ch);

      // Place into image array
     byte imageData[] = bStrm.toByteArray();
     // Create the image from the byte array
     im = Image.createImage(imageData, 0, imageData.length);

    }
    catch (Exception e) {
     System.out.print(e.getMessage());
     e.printStackTrace();
    }
  }
};
```

```
  t.start();
  return (im == null ? null : im);

} //end getImag

public String invokeWebService() {
```

```
System.out.println("ddddddddddd" + i);
Thread t = new Thread() {
 public void run() {
  try {

    if (i == 1) {

      List channel4Menu = new List("You ",
                    Choice.IMPLICIT);

 SoapObject rpc = new SoapObject
    ("urn:Channel4Crickete", "getCricket");
 respone = ("" + new HttpTransport
          ("http://localhost:8080/axis/services/Channel4Cricket",
           "uurn:Channel4Crickete#getCricket").call(rpc));
 currentMenu = "channel4";
 channel4Menu.append("you can download your image from", null);
 channel4Menu.append(respone, null);

 channel4Menu.addCommand(backCommand);

 display.setCurrent(channel4Menu);
}
if (i == 2) {
 SoapObject rpc = new SoapObject
    ("urn:GreenwichMiliSeconds", "getMiliseconds");
 respone = ("" + new HttpTransport
         (
    "http://localhost:8080/axis/services/GreenwichMiliSeconds",
    "uurn:GreenwichMiliSeconds#getMiliseconds").call(rpc));

 List channel2Menu = new List("GreenwichMiliSeconds ",
                    Choice.IMPLICIT);
 currentMenu = "channel4";
 channel2Menu.append("time in miliseconds is", null);
 channel2Menu.append(respone, null);
 //channel2Menu.addCommand(backCommand);
 display.setCurrent(channel2Menu);
}
if (i == 3) {
 SoapObject rpc = new SoapObject
    ("urn:GreenwichDate", "getDate ");
 respone = ("" + new HttpTransport
          ("http://localhost:8080/axis/services/GreenwichDate",
           "uurn:GreenwichDate#getDate").call(rpc));
```

```
 currentMenu = "channel4";
 List channel4Menu = new List("GreenwichDate ",
                              Choice.IMPLICIT);
 channel4Menu.append("date iss", null);
 channel4Menu.append(respone, null);
 channel4Menu.addCommand(backCommand);
 display.setCurrent(channel4Menu);
}
    if (i == 4) {

    }

   }
   catch (Exception e) {
    System.out.print(e.getMessage());
    e.printStackTrace();
   }
  }
 };

 t.start();

 return respone;
} //end getImage
 public void playerUpdate(Player player,
   String event, Object data) {
    if(event == PlayerListener.END_OF_MEDIA) {
      try {
        defplayer();
      }
      catch(MediaException me) {
      }
      reset();
    }
  }

}
```