Secure Mobile Service-Oriented Architecture

Feng Zhang

KTH Royal Institute of Technology
School of Information and Communication Technology
Communication Systems

Doctoral Dissertation in Communication Systems

Stockholm, Sweden 2012
TRITA-ICT/COS AVH 12:06
ISSN: 1653-6347
ISRN: KTH/ICT/COS/AVH-12/06-SE
ISBN: 978-91-7501-351-0


@ Feng Zhang, 2012
Abstract

Mobile transactions have been in development for around ten years. More and more initiatives and efforts are invested in this area resulting in dramatic and rapid development and deployment of mobile technologies and applications. However, there are still many issues that hinder wider deployment and acceptance of mobile systems, especially those handling serious and sensitive mobile transactions. One of the most important of them is security.

This dissertation is focused on security architecture for mobile environments. Research issues addressed in this dissertation are based on three currently important groups of problems: a) lack of an open, comprehensive, adaptable and secure infrastructure for mobile services and applications; b) lack of standardized solutions for secure mobile transactions, compliant with various regulatory and user requirements and applicable to different types of popular mobile devices and hardware/software mobile platforms; and c) resource limitations of mobile devices and mobile networks.

The main contribution of this dissertation is large-scale, secure service-oriented architecture for mobile environments. The architecture structures secure mobile transaction systems into seven layers, called trusted stack, which is equivalent to ISO/OSI layered networking model. These layers are, starting from the bottom: 1) secure element (chip) layer, 2) applets layer, 3) middleware layer, 4) mobile applications layer, 5) communication layer, 6) services broker layer, and 7) mobile service provider layer. These seven layers include all necessary components required for implementation and operations of secure mobile transaction systems and therefore provide a framework for designing and implementing such systems.

Besides the architecture, four types of security services necessary and critical for serious mobile transactions, have also been designed and described in the dissertation. These services are: (1) mobile registration and identity management; (2) mobile PKI; (3) mobile authentication and authorization; and (4) secure messaging. These services are lightweight, therefore suitable for mobile environments, technologies and applications, and also compliant with existing Internet security standards.

Finally, as the proof of correctness of the proposed concept and methodology, a prototype system was also developed based on the designed security architecture. The system provides comprehensive security services mentioned above to several types of mobile services providers: mobile banking, mobile commerce, mobile ticketing, and mobile parking. These types of providers have been selected only as currently the most popular and representative, since the architecture is applicable to any other type of mobile service providers.

Keywords: Mobile, Security, Service-Oriented Architecture
Performing Ph.D. research and writing of a Ph.D. dissertation can be a lonely and isolating experience, yet it is obviously not possible without the personal and practical support of numerous people.

I would like to express my deep and sincere gratitude to my supervisor, Professor Sead Muftic, who provided me the opportunity to perform this research with him. He has always been giving me tremendous intellectual support and patient guidance, without which I could have not achieved the research goals. Not only did I learn a lot from him, but also enjoyed the time being his student. His wide knowledge and logical way of thinking represent great value for me and will remain in the future.

I also wish to express my warm thanks to Professor Louise Yngström for giving me lots of important and very useful advices, especially during the initial stage of this research.

I am grateful to Dr. Gernot Schmöelzer, Dr. Matei Ciobanu Morogan, Dr. Abdul Ghafoor Abbasi, Dr. Muhammad Awais Shibli, Hao Zhao, Chenchen Yuan and many others who worked in the ICT/KTH Seclab, for their enormous helps with various aspects of my research. It was my great honor to work with them, what also brings so much fun to my research.

Many people on the faculty and staff of the ICT School of KTH and DSV of Stockholm University assisted and encouraged me in various ways during last few years. I am especially grateful to Birgitta Olsson, Fatima Ferreira, Irina Radulescu and Anna Barkered, who helped me with administrative issues, such as documents, office, traveling and visa, which saved me a lot of time.

The financial support of the School of ICT in KTH is gratefully acknowledged.

Finally, and most importantly, I own my loving thanks to my wife Yidi Zhao for her great support and understanding during my research. Without her encouragement and understanding, it would be impossible for me to complete this work.

Feng Zhang
Stockholm, 2012
Dedication

To my parents:
Weiping Zhang &
Guixiang Wu
# Table of Contents

Abstract .......................................................................................................................5  
Acknowledgement ......................................................................................................7  
Dedication ....................................................................................................................9  
Table of Contents ......................................................................................................11  
List of Figures ...........................................................................................................15  
List of Tables ............................................................................................................17  

## PART I: INTRODUCTION

### Chapter 1: Introduction .........................................................................................21  
1.1 Background ..........................................................................................................21  
1.2 Research Problems, Methodology and Approach ..............................................23  
  1.2.1 Research Problems .........................................................................................23  
  1.2.2 Research Methodology and Approach ............................................................28  
  1.2.3 Research Objectives .......................................................................................29  
  1.2.4 Research Foci ..................................................................................................29  
  1.2.5 Design and Development Results ..................................................................30  
1.4 Delimitations ........................................................................................................31  
1.3 Outline of the Thesis ............................................................................................31  

### Chapter 2: Related Work ......................................................................................33  
2.1 Overview of Secure SOA for Mobile Environments ...........................................33  
2.2 Overviews of Secure Mobile Identity Management ............................................37  
2.3 Overviews of Mobile PKI .....................................................................................39  
2.4 Overviews of Mobile Authentication, Access Control and Authorization ..........42  
2.5 Descriptions of Some existing Mobile Payment Systems ..................................44  
  2.5.1 MobiPay .........................................................................................................44  
  2.5.2 SEMOPS ........................................................................................................45  
  2.5.3 SmartMoney ..................................................................................................45  
  2.5.4 PayPal ..........................................................................................................45  

## PART II: SECURE MOBILE SERVICE-ORIENTED ARCHITECTURE

### Chapter 3: A Model of Secure Mobile Service-Oriented Architecture ............49  
3.1 Introduction and Approach ...............................................................................49  
3.2 Trusted Stack Model ..........................................................................................49  
3.3 Secure Mobile SOA ............................................................................................52  
3.4 Service Providers and Services ..........................................................................53  
  3.4.1 Mobile Service Consumer – Secure Mobile Client (SMC) .........................54  
  3.4.2 Communication Services Providers ...............................................................54  
  3.4.3 Security Service Providers ............................................................................56  

### Chapter 4: Registration and Identity Management Mobile Services ............59  
4.1 Overview of Registration and Identity Federation Mobile Services ................59  
4.2 Layered Architecture for Identity Management Mobile Services ....................61  
  4.2.1 Services Consumers Layer ............................................................................62  
  4.2.2 Domain Layer Service Providers .................................................................62
4.2.3 Infrastructure Layer Service Providers .............................................. 62
4.3 Mobile Registration Services .............................................................. 62
  4.3.1 Consumer Registration ................................................................. 62
  4.3.2 Service Provider Registration ......................................................... 64
  4.3.3 Domain Registration Service ......................................................... 65
4.4 Federated Identity Management Service .............................................. 66
4.5 Identity Verification Protocol .............................................................. 67
4.6 Evaluation ............................................................................................ 69
4.7 Summary ............................................................................................. 70

Chapter 5: Mobile PKI (m-PKI) Service ...................................................... 71
  5.1 Overview of m-PKI Services ................................................................. 71
  5.2 m-PKI Architecture ............................................................................ 72
  5.3 m-PKI Client’s Functionalities ............................................................. 73
  5.4 m-PKI SPs’ Functionalities ................................................................. 73
  5.5 Lightweight Certificate Management Services .................................... 74
     5.5.1 Certificate Request and Certificate Distribution Service ............... 74
     5.5.2 Certificate Revocation Service ...................................................... 76
  5.6 Evaluation ............................................................................................ 76
     5.6.1 Evaluations of Usability ................................................................. 76
     5.6.2 Evaluation of Security ................................................................. 77
  5.7 Summary ............................................................................................. 77

Chapter 6: Authentication and Authorization Mobile Services .................... 79
  6.1 Overview of Authentication and Authorization Services for Mobile
     Transactions ....................................................................................... 79
  6.2 Authentication and Authorization Mobile Service ................................. 80
     6.2.1 Certificate-based Authentication .................................................. 80
     6.2.3 Role-based Authorization ........................................................... 82
     6.2.4 Transaction Authorization ........................................................... 83
     6.2.5 Location-based Authentication and Authorization ....................... 83
  6.3 Evaluations .......................................................................................... 91
     6.3.1 Certificate-based Authentication and Role-based Authorization ...... 91
     6.3.2 Location-based Authentication and Authorization ....................... 92

Chapter 7: Security of Mobile Messages ..................................................... 95
  7.1 Overview of Mobile Messages Security ............................................... 95
  7.2 Message Security Options ................................................................... 95
  7.3 Message Format .................................................................................. 96

PART III: CONCLUSIONS

Chapter 8: Validation and Verification of Research Results ....................... 103
  8.1 Descriptions of the Prototype System ............................................... 103
     8.1.1 System Components .................................................................... 104
     8.1.2 System Functions and Services ................................................... 106
     8.1.3 System Activation and Administration ....................................... 108
     8.1.4 Secure Mobile Client ................................................................. 116
  8.2 Functional Verification with Finite State Machine ............................... 123
  8.3 Regulatory Framework for Mobile Payment System ............................. 125
# List of Figures

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Design Science Research Methodology (DSRM) Process Model</td>
<td>29</td>
</tr>
<tr>
<td>2</td>
<td>A Simplified View of a SOA with Support for Mobile Services</td>
<td>34</td>
</tr>
<tr>
<td>3</td>
<td>Dynamical m-Services Architecture</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>Core Architecture of the Mobile Host</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>Public Key Platform Implemented in Finland</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>Enrolment Scenario</td>
<td>41</td>
</tr>
<tr>
<td>7</td>
<td>Wireless PKI Model</td>
<td>41</td>
</tr>
<tr>
<td>8</td>
<td>Abstract Model for Origin Authentication Using Joint Signature Scheme</td>
<td>43</td>
</tr>
<tr>
<td>9</td>
<td>Trusted Stack Model</td>
<td>51</td>
</tr>
<tr>
<td>10</td>
<td>Layered Service-Oriented Architecture</td>
<td>53</td>
</tr>
<tr>
<td>11</td>
<td>Layered Architecture for Registration and Federated Services</td>
<td>61</td>
</tr>
<tr>
<td>12</td>
<td>Strong Registration Service</td>
<td>64</td>
</tr>
<tr>
<td>13</td>
<td>Unilateral Registration Service</td>
<td>65</td>
</tr>
<tr>
<td>14</td>
<td>Mutual Registration Service</td>
<td>66</td>
</tr>
<tr>
<td>15</td>
<td>Inter-Domain Identity Verification</td>
<td>67</td>
</tr>
<tr>
<td>16</td>
<td>International Identity Verification Protocol</td>
<td>68</td>
</tr>
<tr>
<td>17</td>
<td>m-PKI Architecture</td>
<td>72</td>
</tr>
<tr>
<td>18</td>
<td>Certificate Request Service</td>
<td>75</td>
</tr>
<tr>
<td>19</td>
<td>Certificate Revocation Service</td>
<td>76</td>
</tr>
<tr>
<td>20</td>
<td>Certificate-based Authentication Service</td>
<td>82</td>
</tr>
<tr>
<td>21</td>
<td>Role-based Authorization Service</td>
<td>83</td>
</tr>
<tr>
<td>22</td>
<td>Location-based Authentication Architecture</td>
<td>85</td>
</tr>
<tr>
<td>23</td>
<td>Location Registration</td>
<td>86</td>
</tr>
<tr>
<td>24</td>
<td>Authentication and Authorization Mobile Services</td>
<td>87</td>
</tr>
<tr>
<td>25</td>
<td>Location Information</td>
<td>89</td>
</tr>
<tr>
<td>26</td>
<td>Step 1 of Location Verification</td>
<td>90</td>
</tr>
<tr>
<td>27</td>
<td>Step 2 of Location Verification</td>
<td>91</td>
</tr>
<tr>
<td>28</td>
<td>Message Security Options</td>
<td>96</td>
</tr>
<tr>
<td>29</td>
<td>Format of Secure Messages</td>
<td>97</td>
</tr>
<tr>
<td>30</td>
<td>Message Confidentiality Protection</td>
<td>97</td>
</tr>
<tr>
<td>31</td>
<td>Message Integrity Protection</td>
<td>98</td>
</tr>
<tr>
<td>32</td>
<td>Message Confidentiality and Integrity Protections</td>
<td>99</td>
</tr>
<tr>
<td>33</td>
<td>Administration Station Login Panel</td>
<td>108</td>
</tr>
<tr>
<td>34</td>
<td>System Username/Password Login Panel</td>
<td>109</td>
</tr>
<tr>
<td>35</td>
<td>The Form to Specify System Configuration</td>
<td>109</td>
</tr>
<tr>
<td>36</td>
<td>The Form to Register Security Domain</td>
<td>110</td>
</tr>
<tr>
<td>37</td>
<td>The Form to Register the Security Administrator of System</td>
<td>111</td>
</tr>
<tr>
<td>38</td>
<td>Initial Panel and Drop-down Selection of Subsystems</td>
<td>111</td>
</tr>
<tr>
<td>39</td>
<td>The Main Panel of the Identity Management System</td>
<td>112</td>
</tr>
<tr>
<td>40</td>
<td>“Persons” Drop-down Menu</td>
<td>112</td>
</tr>
<tr>
<td>41</td>
<td>Registration of Mobile Payment Server</td>
<td>113</td>
</tr>
<tr>
<td>42</td>
<td>“Login to server” Pop-up Option</td>
<td>113</td>
</tr>
<tr>
<td>43</td>
<td>Registering Message Channels and Message Rules</td>
<td>114</td>
</tr>
<tr>
<td>44</td>
<td>List Messages</td>
<td>114</td>
</tr>
<tr>
<td>45</td>
<td>Registering of Message Channel</td>
<td>114</td>
</tr>
<tr>
<td>46</td>
<td>Registration of Message Rules</td>
<td>115</td>
</tr>
<tr>
<td>47</td>
<td>Registration of System Setting</td>
<td>115</td>
</tr>
</tbody>
</table>
Figure 48. Example of the SMS Client ................................................................. 116
Figure 49. Example of the Mobile Thin Client .................................................. 117
Figure 50. Example of the Mobile USSD Client .................................................. 118
Figure 51. Example of the Mobile Thick Client .................................................. 119
Figure 52. Example of the Mobile SIM/UICC Client ........................................ 120
Figure 53. Internal Structure and Components of the Secure Mobile Client........ 122
Figure 54. Example of the Process for Storing Data on the UICC ....................... 122
Figure 55. Finite State Model of our System ....................................................... 125
Figure 56. Cash Dispensing Transaction Messages: Mobile and Static ATM .......... 137
Figure 57. Message Flows in Mobile-ATM .......................................................... 138
Figure 58. Account-to-Account Transfer Transaction ........................................ 139
Figure 59. Loans: Application and Administration ............................................. 140
Figure 60. Components of m-Commerce System .............................................. 141
Figure 61. Cash Dispensing and Micropayments ............................................... 142
Figure 62. Debit and Credit Card Payment ......................................................... 143
Figure 63. Mobile Parking System Architecture ............................................... 143
Figure 64. System Architecture for Mobile Ticketing ........................................ 144
List of Tables

Table 1. Attacker Capabilities to GSM Networks .........................................................25
Table 2. Evaluation of Security of FIM Service ..........................................................70
Table 3. Security Evaluation Results .........................................................................77
Table 4. Security Evaluation Results .........................................................................92
Table 5. Security Evaluation Results .........................................................................93
PART I: INTRODUCTION

Part one contains the introduction and background of this research. Chapter 1 describes research area, research problems, research methodologies, and research goals. Chapter 2 contains the overview of activities and reported results in these research areas.
Chapter 1: Introduction

This Chapter represents the overall introduction of this research: background and overview of existing problems in the research areas, research methodology used in order to achieve the research goals, research focus and objectives, and finally, the outlines of the thesis.

1.1 Background

At the very beginning of human society, people exchanged goods with equal values. Later, the concept of money was introduced, so people started to use money for transactions. Because of industrial revolution and globalization, commercial transactions have rapidly increased. However, exchanging a large amount of money is a risky task. As a solution, people started using banking facilities for financial transactions. At present, banks provide attractive services for effective money transactions. However, many problems related to bank transactions still remain.

In developing countries, these problems have become even worse. In many rural areas there are no bank branches or bank facilities, such as ATMs. If people need to use bank facilities, they have to travel far to the nearest town or city, which is inconvenient and influences the development of local economies. However, requirements for using bank facilities in rural areas are rapidly increasing. Every year migrants send billions of dollars back to their families, most of them living in rural areas. This is a large industry, but only large companies are capable to deal with such transactions. Even worse is that most of these companies do not have branches in rural areas. In addition, implementing and using IT-based solutions in developing countries is also a big challenge due to poor communication and IT infrastructure, which is a bottleneck for expansion of bank facilities.

On the other hand, wireless and mobile technologies have been experiencing dramatic growth. Functional capabilities of mobile telephony have been rapidly expanding and have extended their usage well beyond the classical communication applications – telephone calls and short messaging. There is mounting evidence of positive financial, economic and social impacts of those technologies all over the world. Although the IT infrastructure is usually undeveloped in rural areas, remarkably in most of the developing countries mobile telecommunication sector achieved a rapid expansion in recent years (Center Bank of Sri Lanka 2007). Hence, mobile communication infrastructure could be used as an effective deployment platform for electronic banking and financial systems.

During the last two decades, mobile technology has achieved great progress. At one hand, mobile networks improved from cellular networks to 3G, now 4G networks. On the other hand, mobile devices become much more powerful than twenty years ago.

Today, mobile phones already became one of the most important accessories in people’s everyday lives all over the World. According to the GSMA, an industry
group, total number of mobile subscribers in the World will reach 6 billion by 2013 (Mobile marvels 2009). With the development of mobile networks, mobile industry is changing from being voice-centric to data-centric, when consumers spending 90% of their time talking, to being engaged on mobile data services 80% of the time (Chetan Sharma Consulting 2011). Therefore, in addition to standard communication services—phone calls and exchange of Short Messaging Service (SMS) messages, mobile applications, such as mobile banking and mobile commerce, became more and more popular. Mobile technology is considered an innovative approach to complement current, mainly web-based applications and transactions and to transform existing, still usually paper-based financial systems into cashless systems. Financial institutions and mobile network operators (MNO) are investing a lot of efforts to convert their payment transactions, such as air-time top up, and money transfers or bill payments, into mobile electronic form. Customers are also demanding such kinds of mobile applications to access resources, database, and services, provided by various service providers, anytime and anywhere.

Due to significant decrease of the cost of mobile phones and mobile services, currently there are many initiatives to perform transactions using mobile technology (Cardinal Bank n.d.), (SunTrust n.d.), (Mendes, et al. 2007), (infoDev 2006). However, all of those initiatives today are just “point-solutions”. Most of the current solutions and implementations of mobile financial systems have only two components: a simple mobile phone application and a corresponding mobile transactions server. The server understands limited set of messages coming from phones, uses background financial data to perform transactions, and returns the result. Such concept and systems cannot be easily extended with new functions, since both client application and server must be modified. Those solutions do not scale, meaning that phone application, designed to access and use one server, cannot access and use any other mobile financial server. Finally, these systems cannot be interconnected, so that users registered in one system, cannot transfer funds to and use functions of other systems.

Another fact, which is more important, is that the security of these solutions is not adequate. In most of them, security even does not exist. There are two main reasons for this: external reason and internal reason. The external reason is that the environments for mobile transactions and applications are not secure. Compared with wired networks, wireless networks are much more vulnerable to attacks. Wireless technology has achieved great expansion during the last several years and it is extremely complex, compared with wired networks. Under the circumstances mentioned above, there are many security issues with wireless networks which hackers consider as the weakest point in the communication chain of mobile financial transactions. During this thesis, possible threats and security issues in several networks, which are used for mobile transactions, were analyzed. The results are described in Section 1.2.1. The internal reason is that wireless engineers usually are not security experts and they all assume that security will be added later, if it is needed. This is not a good approach, since security must be integrated with wireless technologies. This leads to the fact that existing solutions need a dynamic, flexible and adaptable security architecture. Under this architecture, various security services should be provided for mobile environments, where the mobile clients using different mobile devices with different mobile platforms are involved in the same and relatively fixed mobile services.
This thesis was originally motivated by an idea of a solution that provides secure mobile financial system supporting transactions, such as mobile banking and mobile commerce for developing countries (SPIDER 2007). During the first phase of this research, intensive software engineering work was performed in order to develop and the necessary components of the secure mobile architecture. All those components cooperate with each other to provide functionalities for mobile transactions. After these initial results, a secure mobile service-oriented architecture was designed based on the established components and functionalities. Based on this architecture, a comprehensive set of security services and protocols have been designed and implemented. In this process many standards were used for the design and adjustments to the architecture, components and protocols were conducted to make the solution proposed in this thesis more flexible, adaptable and interoperable. Finally, the evaluations and verifications were performed for the designed system in order to check whether it could address the research issues.

1.2 Research Problems, Methodology and Approach

1.2.1 Research Problems
During the initial phase of this research, I did a comprehensive study for collecting information in order to identify research problems targeted in this research. It included studying existing mobile transaction systems and solutions, observing the real usage scenarios of mobile transactions, and getting the response from customers about usage experiences, suggestions and requirements. After that, I summarized and analyzed the collected information from the security point of view in order to specify precisely targeted research problems.

One of the main research problems for mobile transaction environments is the gap between the complexity of business requirements and difficulties to achieve practical business objectives. Technical solutions often must be adjusted in order to achieve business objectives. Many ecosystem players must be involved in mobile transactions, such as banks, MNOs and software vendors. These ecosystem players must cooperate with each other in order to provide mobile application transactions, which unfortunately is difficult to achieve. There are many reasons for this, based on both business and technical aspects. One of the reasons is that most of the practical implementations are targeting solutions for specific environments or situations.

There are several existing systems providing mobile transactions, but all these systems deployed at the time when this research started were just “point solutions”, meaning that each system can provide a specific service to a specific application in a specific environment. Most of the current solutions and implementations of mobile transaction systems comprise mainly two components: a simple mobile phone application and a corresponding mobile transactions server. The server understands limited set of messages coming from phones, uses background financial data to perform transactions, and returns results. Such concept and systems cannot be easily extended with new functions, since in that case both client application and server must be modified. Those systems do not scale, meaning that phone application, designed to access and use one server, cannot access and use any other mobile financial server. Finally, these systems cannot be interconnected, so that users...
registered in one system can transfer funds to and use functions of other systems. Another fact, which is more important, is that security of these solutions is not adequate. In most of them, it even does not exist. This leads to the conclusion, that existing mobile solutions do not have a dynamic, flexible and adaptable security architecture, which can provide security services for mobile environments, where lots of mobile clients using different mobile devices with different mobile platforms are using various mobile services. As suggested by Ramsden, “The main challenges facing telecom operators are a lack of infrastructure, the need for cooperation between operators, retailers and banks, and the need to create clearly defined revenue models for all investing parties” (Ramsden 2010).

Another research problem is based on weaknesses of mobile networks and resource limitations of mobile devices. The reasons are based on the following aspects:

First of all, mobile networks are different from wired networks. All security threats existing in wired networks exist also in mobile networks. The use of mobile networks introduces more threats than fixed networks. The security analyses of mobile environments are described as follows.

1.2.1.1 GSM Network

GSM network is the most popular environment for mobile transactions. Threats and security issues in those networks were analyzed first. Five acknowledged attacker capabilities influence the security in GSM networks, shown in Table 1 (Gadaix 2001). The first capability is the easiest to conduct. What is new, in December 2009, a German computer engineer announced that he had deciphered and published a secret code used to encrypt most of the world's digital mobile phone calls, saying it was his attempt to expose weaknesses in the security of global wireless systems (O’Brien 2009). Therefore, we may say that the security provided by existing GSM networks is not adequate.

1.2.1.2 Short Messaging Service (SMS)

Mobile messaging market is rapidly growing and it is becoming large profitable business for telecom operators, especially in developing countries. And most of these mobile transaction systems utilize SMS service as their communication system. However, there are several issues with SMS messages:

- **SMS Spam**: Spamming is an action where the subscriber receives an unsolicited message, i.e. the one the subscriber did not want to receive. Spam SMS may take different forms including: commercial information, advertisement, bogus content and other messages generally intended to invite a response from the receiver.

- **Flooding**: Flooding is the case when a large number of messages are sent to one or more destinations.

- **SMS Fraud**: Nowadays there are many fraudsters trying to organize crime or terrorist cells – and they are using SMS fraud to fund their operations. For example, premium rate service (PRS) fraud is the case where the subscriber pays a higher than normal rate for a call in exchange for information (such as adult chat lines). Fraudsters send to subscribers an unsolicited SMS telling the subscribers that they have won a prize and need to call the specified number to collect their prize.
**Table 1. Attacker Capabilities to GSM Networks**

<table>
<thead>
<tr>
<th>Difficulty to counter</th>
<th><strong>Eavesdropping</strong></th>
<th>The capability of an intruder to intercept traffic and signaling information transferred to other users. The required equipment is a modified mobile phone.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Impersonation of a User</strong></td>
<td>This is the capability to send rogue data and/or signaling messages to the network with the intent of making them appear as originating from another user. This also requires only a modified mobile phone.</td>
</tr>
<tr>
<td></td>
<td><strong>Impersonation of the Network</strong></td>
<td>This is the capability to send rogue data and/or signaling messages to another user with the intent of making them appear as coming from a genuine network. This requires a modified BTS.</td>
</tr>
<tr>
<td></td>
<td><strong>MITM – Man-In-The-Middle Attack</strong></td>
<td>This is the capability of an attacker to insert itself between the network and the legitimate user in order to eavesdrop, modify, delete, re-order, re-play and spoof signaling data between the two parties. This requires a modified BTS in conjunction with a modified mobile phone.</td>
</tr>
<tr>
<td></td>
<td><strong>Network Authentication Compromise</strong></td>
<td>The intruder possesses a compromised authentication vector (challenge-response pairs, cipher keys, integrity keys, etc.)</td>
</tr>
</tbody>
</table>

**Impersonation of a user:** There are some websites, such as https://www.hoaxmail.co.uk/ providing the service that a person, who has paid the fees for their service, can send SMS message from a specific originator to a specific destination. In other words, this person is able to impersonate other users as originators of short messages.

**1.2.1.3 Bluetooth**

Bluetooth, a standard for local wireless communications of devices, such as mobile phones, wireless headsets, printers and cars, is developing rapidly, so that more and more mobile users transfer or share data with each other. However, there are some security issues:

- **Eavesdropping** – Bluetooth session starts with agreeing about a key by two communication entities and this key is vulnerable to attackers in some circumstances. Attackers can either get the key by exhaustively searching all possible PINs (without interacting with the victim devices) or by Man-in-the-Middle attack (RSA Laboratories 2000).
- **Location Attack** – Attackers are able to determine geographic location of victims, what can be an advantage by some attackers for many illegal purposes.
- **Vulnerable Cipher** – As Jakobsson and Wetzel mention, they conducted several attacks on the cipher, which indicate that the cipher of the Bluetooth is not strong enough (Jakobsson and Wetzel 2001).
1.2.1.4 Mobile Internet
Due to 3G and the newest 4G technology, mobile Internet became much more popular than ten years ago. Nowadays, more and more mobile users are surfing Internet by using mobile phones any time, any place. Security issues still exist in mobile Internet even though some users do not care about that. However, for customers who use mobile Internet for financial transactions, security is one of the most important concerns.

The most common security solution used for Internet security today is SSL, which relies on complex combination of public key cryptography, symmetric key cryptography, hash function, digital certificates, and digital signatures. There is a security protocol for mobile Internet, a lightweight version of SSL, called Wireless Transport Layer Security (WTLS). WTLS functions very similar to SSL, but with a number of additional characteristics, such as compact coding, datagram support, optimized handshakes, dynamic key refreshing, fast encryption and hashing algorithms, client-gateway rather than client-server coverage and so on. (The Kannel Group n.d.).

Since it seems that WTLS covers many security issues of mobile Internet, maybe we could use it also for security of financial transactions. The answer is “NO” and the reason is that WTLS provides only communication security between client devices and WAP gateway. However, most mobile Internet transactions, especially financial transactions, require also security between the WAP gateway and backend Web servers. In other words, for mobile financial transactions end-to-end security is needed. There are two possible approaches to achieve this goal:

- SSL could be extended between the WAP gateway and the backend Web servers. In this way, client builds WTLS session with the WAP server and sends credential data (for example credit card information) to a WAP gateway in an encrypted format. Gateway decrypts the credential data and uses new SSL session to send it to Web servers. In this case, security relies on the administrative access to the WAP gateway, which is not suitable for financial transactions.
- Alternatively, the gateway can be hosted by the Web service providers and therefore placed behind the service provider’s firewall. This is also not a good solution, because Web service providers do not provide mobile Internet services to all their mobile users.

According to the security analysis described above, it is much more challenging to protect mobile networks than wired networks. A security architecture that allows application designers to develop applications to be as secure when running on mobile networks as they are when running on fixed networks is lacking.

Second, even though mobile handsets have achieved dramatic improvements. That makes them not only equipment for telephony, but also a small “computer” with many useful functionalities integrated inside. However, there are still many old devices being used with several limitations explained below, especially in developing countries. This becomes one of the barriers for deploying mobile transaction systems on a large scale. There are mainly five types of limitations (Wang and Higgins 2006):

- **Small screens and low resolutions:** Even though mobile phone vendors are trying to make screens bigger and bigger, it cannot be large enough, since in
that case mobile device will not be portable or convenient. Screen resolution is another problem.

- **Input limitations:** Current input capabilities of mobile phones are not user-friendly, which leads to slow input with spelling errors and inconvenient input methods.

- **Limitations when accessing Internet:** Due to differences between mobile phone wireless networks and cabled Internet networks and different coding methods, it is difficult to access ordinary Internet Web pages using mobile phones. Most ordinary Internet Web pages are distorted on mobile phone screens. Therefore, in order to enable mobile phones to browse websites, people have to design specific Web pages by using HDML, XHTML and other special coding technologies.

- **Lack of standardization and compatibility:** There are no standards in terms of communication protocols, functionalities and so on. For example, some countries, like US, are using CDMA2000, Japan is using W-CDMA and China is trying to use TD-SCDMA. Lack of compatibility is also a big problem for design and deployment of mobile transaction systems. Some mobile phones support E-mail functions, some do not. Even SMS services are not compatible in some countries.

- **Other limitations:** Some other limitations are limited storage and computing capabilities, limited length of short messages, disabled SSL and so on.

These limitations of mobile devices should be taken into account when designing large-scale secure mobile transaction systems. Due to these limitations most of the existing security mechanisms, used in wired networks, are too complicated for mobile environments, meaning that mobile devices can not perform complex computational procedures as standard computers. Hence, security mechanisms used in wired networks can not be directly applied to mobile environments, which is another research issue this thesis addresses.

The third research issue is large variety of mobile devices and platforms. There are more than thirty big manufactures of mobile phones and many small ones in the world producing thousands types of mobile devices with different screen sizes, processor abilities, memory sizes, and extra features. Moreover, those mobile devices are sold on the market only for a short period of time and usually quickly replaced by new devices. In addition, there are various runtime platforms, such as iOS, Android, RIM, Symbian and Windows Mobile. Each platform has its own specifications. Even for a specific runtime platform there are several versions. The situation described above introduces several problems:

- Serious challenges for mobile software developers. Either they just focus on one platform or they must create many versions of the same application;

- Software developed for one runtime platform can not be easily ported to other platforms, which do not support specifications of the original platform; and

- There is no standardized security protocol applicable to all operating systems, since the format of data and messages is not standardized.

The fourth issue is malware threatening mobile devices. The number of smart mobile phones expanded dramatically during the last several years. This represents
significant target for malware. Since the first security threat for mobile technology, detected in 2004 (Hypponen 2006), security issues, like viruses and malware targeting mobile devices, have grown rapidly every year. At the same time, more and more mobile applications require local storage of users’ data. Sometimes such data is so sensitive that it must be strongly protected, for instance, social security numbers, credit card numbers and bank accounts. Based on the situation described above, the expansion of smart phones and mobile applications brings serious risks to users’ data. Protection of mobile applications and users’ data is the next significant objective and one of the research contributions.

Finally, an issue is the integration of mobile transactions and applications with security services. As mentioned in Section 1.1, most wireless developers are not security experts and they usually consider adding security to their applications later, which causes many security issues. Based on the experiences and feedback reflected by this research, I believe that it is important to take security into account during the design phase of mobile transaction systems, which requires an overall architecture that includes not only mobile services, but also network security services.

Based on the situation mentioned above, we can conclude that one of the significant research issues is to design a mobile security architecture that provides secure applications and services for various devices in mobile environments. When this research was initiated four years ago, there was no such security framework providing comprehensive protection, which could be easily adopted by existing systems. This was one of the main topics of this research and one of the main research contributions.

1.2.2 Research Methodology and Approach

According to Martin Shuttleworth, research means performing a methodical study in order to prove a hypothesis or answer a specific question (Shuttleworth 2008). The process of research is called research methodology. There are many types of research methodologies. For different research subjects, the research methodology may be different. The approach called Design Science Research Methodology (DSRM) was adopted by me to conduct this research. Design Science is a research paradigm in which a designer answers questions relevant to human problems by creation of innovative artifacts, thereby contributing with new knowledge to the body of scientific evidence (Peffers, et al. 2007). Designed artifacts are both useful and fundamental in understanding targeted problems (Hevner, March and Park 2004). The process of DSRM is shown in Figure 1. I followed the nominal process sequence by starting with identifying research problems and defining research objectives. In order to solve the research problems, design and development phase followed. A prototype system was developed, demonstrated and deployed in order to get the feedbacks from users. Improvements of the system were created based on the feedbacks. During the design and development phase, several results were published. Finally, the system was evaluated and verified using both theoretical analysis and practical deployment activities.
1.2.3 Research Objectives

The main research objective this thesis achieves is a solution that provides a secure, flexible, scalable and extendible environment for mobile transactions. In order to achieve this research objective, there are three sub goals that need to be achieved:

a. Designing a trusted model for mobile environments, so that it is not only convenient and secure for mobile users to perform mobile transactions, but also easy for application providers to develop trusted and platform-independent application transactions;

b. Designing security services, like authentication and authorization, based on the proposed secure SOA, for protecting mobile transactions, which can achieve end-to-end security. This would solve the issues that many application developers are not security experts in the way that the application services created by these providers do not have to provide comprehensive security features;

c. Making mobile transactions or services available for all kinds of mobile devices in various environments, so that users could perform transactions anytime, anywhere without compromising trust and security.

1.2.4 Research Foci

Based on the challenges mentioned above, what is needed in the current situation is a concept, a set of standards, and implementation tools for a comprehensive security architecture providing various mobile security services.

The approach of this thesis was not only to extend basic principles of service-oriented architecture (SOA) to mobile environments, but also to provide security solutions to address the research issues mentioned in Section 1.2.1. The security architecture is modular, integrated, extendible, and scalable. It provides security services, such as confidentiality, integrity, authentication, authorization and non-repudiation for mobile applications and transactions.

During the design and development phase, we also focused on building each component, interface and message based on the solutions proposed in this thesis.
The reason is to verify that the solution is functionally correct, geographically scalable, compatible with other similar solutions and extendible with additional mobile services.

1.2.5 Design and Development Results

The architecture, secure services and methodology proposed in this research is expected to represent important contributions to a methodology for building secure environments for mobile applications and transactions. In this research, the following results have been achieved:

1.2.5.1 Secure SOA for Mobile Environments

A large-scale comprehensive SOA for secure mobile applications and transactions was designed. It is

- comprehensive, i.e. many security services are provided by the architecture;
- scalable, it provides the possibility for interlinking mutually independent deployments, if based on the described architecture;
- modular, i.e. new services, functions and components can be easily added to the architecture;
- expandable, i.e. mobile applications can easily be linked to the architecture and can utilize its services; and
- open, meaning that integration of new components is based on utilization of standard-based web services and interfaces.

Based on this secure mobile SOA, functions of all mobile applications are exposed to mobile users as services. These services are easy and flexible to use for end-users in the way that user can access services anytime, anywhere, and through various communication channels. It is also easy and flexible for service providers to manage their mobile services. A secure SOA for mobile environments was designed in this thesis and a practical system was implemented as the proof-of-concept prototype. The solution comprises service components, service interfaces, and messages exchanged between the components. Various security services were designed for mobile applications and transactions: registration, certification, authentication and authorization of users, secure messaging at an application-level (end-to-end security), protection of data in databases, and protection of each individual component.

1.2.5.2 Lightweight Mobile Security Services

In order to provide comprehensive protection for mobile environments, the following lightweight security services are designed, suitable for mobile networks and mobile devices:

- Mobile Registration and Identity Management Services;
- Mobile Certificate Management Services (CMP);
- Certificate-based Strong Authentication Services;
- Role-based Access Control Services;
- Location-based Authentication and Authorization Services;
- Message Protection Services.
1.2.5.3 Secure Mobile Financial Applications

The prototype system designed in this dissertation provides various applications using mobile phones and other mobile hand-held devices. It supports transactions with banks, direct client-to-merchant payments, person-to-person transactions, and other non-banking mobile financial applications. One of the main features of the system is to manage and use mobile pre-paid accounts (PPAs). These accounts may be used to deposit and withdraw cash and for other various mobile payments. The system is based on service providers for various mobile services, so that subscribers with pre-paid accounts are able to use the PPAs to pay for the services. These services and applications can be categorized into five groups:

- Mobile banking;
- Mobile financial transactions for unbanked users;
- Mobile commerce;
- Mobile ticketing;
- Mobile parking.

Mobile banking, transactions for unbanked users, and mobile commerce are financial applications, while mobile parking and mobile ticketing are applications that use the system for payments.

1.4 Delimitations

First of all, there was not enough time to conduct comprehensive evaluation of the proposed architecture and security services. Only evaluation of security aspects was performed. Evaluation of other aspects, such as time cost, processing and computational cost has not been performed.

Second, it is really difficult to load applets into the SIM/UICC chip for testing purposes because loading application into SIM/UICC chips requires permission and cooperation of MNOs, which decide what applications can be loaded into the chips. Hence, the applets were tested only under a simulated environment.

1.3 Outline of the Thesis

This dissertation consists of nine Chapters. Chapter 2 contains a review and analysis of existing solutions and related publications. In Chapter 3, a new secure mobile service-oriented architecture is described, including its components and services. Using this architecture and its components, several security services are designed for protecting mobile transactions. Chapter 4, Chapter 5 and Chapter 6 explain the following security services: mobile registration and identity management (m-IDMS), mobile Public Key Infrastructure (m-PKI), mobile authentication, access control and mobile authorization (m-AAA). Chapter 7 describes message security, namely the approach of protecting messages transmitted between components during mobile transactions. In Chapter 8, a prototype system, designed and developed in this thesis is described and the proposed solution is validated and verified using finite state model and regulatory framework. Finally, conclusions and outline significance of our research are described in Chapter 9.
Chapter 2: Related Work

As Isaac Newton said, “If I have seen further (than Descartes) it is by standing upon the shoulders of Giants”. This Chapter contains overview of some related work that have been reported in the research area of this thesis. It overviews not only theoretical work, but also some practical solutions.

In order to solve the research problems described in Chapter 1, we classified the research topics into five areas. The objective of this research was achieved by solving related research issues in each of the areas:

1. Secure global SOA for mobile environments;
2. Secure mobile registration and identity management services (m-IDMS);
3. Mobile PKI services (m-PKI); and
4. Mobile authentication, access control and authorization services (m-AAA);
5. Secure messaging services

The review of the existing reported results in the areas listed above is given in this Chapter. Existing contributions include standards, published research results, and existing deployed solutions.

2.1 Overview of Secure SOA for Mobile Environments

In 2006 Organization for the Advancement of Structured Information Standards (OASIS) defined a reference model for SOA. The reference model is an abstract framework for understanding significant entities and relationships between them within a service-oriented environment and for the development of consistent standards or specifications supporting that environment (OASIS 2006). It is very useful for understanding of the concept, development and deployment of SOA. It is one of the fundamental platforms and frameworks to design secure SOA for mobile environments.

Gurp listed problems with providing mobile services and defined SOA drivers that affect success or failure of mobile services (Gurp, Karhinen and Bosch 2006). These drivers are: usability, portability, deployability and scalability. Then they analyzed three different architecture styles according to these four drivers to highlight the properties of different architectural approaches: client-server with native client, client-server with mobile Java client, and client-server with mobile thin client. From this paper, we adopted important factors for building SOA for mobile environments and the comparisons of some of the existing architectural approaches.

In 2007, Tergujeff described the survey result of the status of necessary enabling technologies, programming interfaces, and the supporting device base in their paper (Tergujeff, et al. 2007). In addition, the review of availability of both mobile consumption and provision of services is given. Based on their results, they concluded that full-fledged mobile participation in SOA is not far in the future. Efficient offering of Web Services on small devices is much more challenging and remains a research subject for the future.
Tran analyzed the relationships between SOA, cloud computing and mobile commerce in his blogs (Tran 2010). He mentioned that retailers inherently suffer from “point-solution pollution”. As more point solutions are introduced to accomplish specific retail business processes, it becomes more difficult for retailers to innovate, as these solutions are typically tightly integrated. Therefore, replacing one solution with some other will have significant downstream effect on other solutions. Therefore, SOA and cloud computing are suggested by the author as part of retailers’ overall mobile commerce strategy. In addition, the author described some of the key considerations for retailers when creating, delivering and accessing mobile services in the cloud: interoperability, usability, security and privacy.

Jørstad investigated how the construction of mobile services can benefit from the service-oriented paradigm and they illustrated a new outlook of an SOA that supports mobile services, shown in Figure 2 (Jørstad, Dustdar and Thanh 2005). Their paper addresses support for mobile services in SOA and in particular, support for service continuity and personalization in such architecture. In addition, the paper provides an elucidation of the SOA and general discussion of equivalence between service components in order to enable the analysis of SOA for mobile services. They emphasize that SOA-based mobile services architecture builds on, and extends, the basic SOA concept with several conceptualizations, components and mechanisms:

1. An ontology for describing mobile services concepts;
2. An ontology for describing key state variables;
3. An ontology for describing interface variables;
4. A mobility controller component that can handle state transfers and coordinate the overall mobile services; needs to implement the service equivalence logic;
5. A mobility controller stub in each composite service which can coordinate actions towards the mobility controller;
6. Standardized interfaces for state transfer (between mobility controller stub and mobility controller);
7. Exposing service profiles as autonomous, self-contained services; and
8. Exposing service content as autonomous, self-contained services.

![Figure 2. A Simplified View of a SOA with Support for Mobile Services](image-url)
Sanchez-Nielsen proposed an open and dynamical SOA for mobile services, addressing three issues (Sanchez-Nielsen, Martin-Ruiz and Rodriguez-Pedrianes 2006):

1. dynamical integration of new services by service providers at any time,
2. dynamical discovery of available services,
3. the use of open source software to develop the solution.

As shown in Figure 3, the framework consists of four components: service providers, service managers, service clients, and UDDI registry, all of which are based on XML infrastructure which provides uniformity. Their proposed framework introduces an intermediate entity, called service manager, located between service providers and service clients, responsible for information flow between both components. A service manager is a web service entity, using dynamic invocation interface (DII). With DII, a service manager can invoke web services without knowing their communication interface at compile time. As a result, (i) invocations of web services, not previously known can be computed by service manager and (ii) service providers can create, update and change web services at runtime. During their implementations, they discovered three drawbacks of using general web services technology for developing mobile services:

1. Java Specification Request 172 (JSR-172) required for invocating web services from a mobile J2ME application does not support UDDI specifications and SOAP encoded messages,
2. The use of UDDI registry provides long time responses and also many of the services published in UDDI registry are not correctly published,
3. Specification implementations must be developed in order to support complex types when dynamic invocation interface is used.

![Figure 3. Dynamical m-Services Architecture](image-url)
Srirama proposed the concept of *Mobile Host* for mobile web service provisioning (Srirama, Jarke and Prinz, Mobile Web Service Provisioning 2006). In their papers, the desirability, challenges for design and implementations of mobile web services provisioning are described. In addition, the details of the prototype *Mobile Host* are described and analyzed to prove its feasibility for mobile web service provisioning. The core architecture of *Mobile Host* is shown in Figure 4. Their proposed approach truly paves scope for P2P, ad-hoc and distributed mobile information networks, using mobile devices not just as web service requestors, but also as mobile hosts that themselves can offer services in a mobile peer-to-peer arrangement.

Natcheto proposed a SOA for mobile applications (Natcheto, Kaufman and Shapiro 2008). In their architecture, a context-based XML compression technique was proposed, which efficiently minimizes the amount of data transferred to and stored on the mobile devices by using the knowledge of business process and data access statistics to identify data required only by the user. In addition, the architecture provides efficient connections to the back-end enterprise system, providing asynchronous message-based communication and various communication channels for mobile devices. Based on their proposed architecture, a sample m-business application was implemented. The application uses proactive data feeding that predicts what information will be needed by a user in the near future and sends it to the client when the connection is up, but idle. The proactive data feeding provides ability for users to store and retrieve server information when the connection is down, which is an important user acceptance factor for mobile environment.

Fonseca proposed a security framework for SOA applications in mobile environments (Fonseca, et al. 2009). The framework is based on the concept of *Mobile Host* proposed in 2006, which is a light-weight service provider built to run on mobile devices. It is developed as a Web Service Handler, performing parsing of received messages as SOAP envelope to an object that contains the details of the messages and uses it to obtain information necessary to implement the request. In addition, the proposed framework provides security services, like digital signature, to...
protect service messages. It provides a developer with a tool for rapid development of services in mobile environments and aggregates all services necessary for provisioning of the service for invocations in an infra-structured network, such as in P2P networks.

The above mentioned paper addressed the issues of adopting SOA in mobile environments. Most of them focused on Web services provisioning in mobile environments, which is difficult for large scale deployment, since it has certain requirements for mobile devices. In addition, these solutions didn’t address security issues, such as malware for mobile devices and other common attacks, even though some of them analyzed the potential attacks, the security requirements, and possible solutions. Therefore, open, feasible and comprehensive security architecture for mobile services is still needed.

2.2 Overviews of Secure Mobile Identity Management

According to Y.2720 (International Telecommunication Union 2009), identity management is a set of functions and capabilities (e.g. administration, management and maintenance, discovery, communication exchanges, correlation and binding, policy enforcement, authentication and assertions) used for:

- Assurance of identity information (e.g., identifiers, credentials, attributes);
- Assurance of the identity of an entity (e.g., user/subscribers, groups, user devices, organizations, network and service providers, network elements and objects, and virtual objects); and
- Enabling business and security applications.

Federation, defined in X.1250 (International Telecommunication Union 2009), is “an association comprising any number of service providers and identity providers”. From the definition we can conclude that Federated Identity Management (FIM) consists of identity providers, service providers, and functions provided either by identity providers or service providers for identity management. There are several advantages of utilizing FIM service (Chadwick 2009):

- it gives users single sign-on (SSO) capability, allowing them to switch between various service providers (SPs) without having to authenticate or log in again;
- it allows SPs to offload the cost of managing user attributes, passwords and login credentials to trusted identity providers;
- it allows identity providers (IdPs) to maintain close relationships with end-users and sell them additional services, as well as extract fees from the service providers they support.

European Network and Information Security Agency (ENISA) published a position paper that reports on information security risks and best-practice in the area of Mobile Identity Management (Mobile IDM). Thirteen threats to Mobile IDM are listed (European Network and Information Security Agency (ENISA) 2010):

1. Identity theft
2. Eavesdropping and spyware
3. Surveillance
4. Phishing  
5. Man in the middle attacks (MITM)  
6. Illegitimate utilization of interception software  
7. Collection and storage of private information beyond the stated purpose  
8. Vulnerable software  
9. Failure to recognize context  
10. Inadequate device resources  
11. Threats to protocols  
12. Intrusive authentication  
13. Lack of user awareness  

The paper also gives recommendations for secure Mobile IDM:

- **Interoperability** – The use of standards for security interoperability and identity management for the backend systems and also the mobile devices, following best practices for security-hardening trusted mobile devices is recommended;

- **User awareness** – public awareness of the privacy and security implications related to mobile identity management should be a key policy to defend against and requires long-term effort.

- **Design objectives for mobile IDM:**
  - **User experience**: A mobile identity management mechanism needs to identify the user and his/her context in an efficient manner: minimizing the delay, the number of false identification results, and the user interruption.
  - **Access and authorization**: A management system must transparently handle the access rights that are pertinent to each identity and relevant to the particular service.
  - **Scalability**: Mobile identity management systems should not only be cost-effective and scalable from the perspective of the operators and/or service providers, but also from the perspective of users.
  - **Resilient connectivity**: At the physical layer, many attacks rely on isolating the “target” from the rest of the network. Provision of reliable connectivity, along with more resilient networks, and the support of mechanisms that allow devices to select appropriate network interface can prevent this kind of attacks.
  - **Malware defenses**: Security mechanisms should be implemented on user devices in order to resist compromise by malicious software.
  - **Control over privacy settings**: To provide tunable privacy protection mechanisms and tools for automatic customization and personalization of services.
  - **Delegation**: To provide privacy protection when various delegation mechanisms are employed.
  - **Accountability**: Service providers should be able to attest the accountability of their users.
  - **Identity selection and composition**: The determination of the type of data that will compose the mobile identity is of primary importance.
  - **Consumer trust**: Addressing user aspects of mobile identities should take place on the micro level of individual users and their devices as well as on the macro level of the environment in which mobile identities are used.
Roussos introduced an enacted view of identity in mobile business in their paper (Roussos, Peterson and Patel 2003) based on three principles: locality principle, reciprocity principle, and principle of understanding. They divided mobile identity into three classes (and their combinations): device-to-device, location-to-location, and context-to-context. In device-to-device identity, mobile identity can be used in the certification process to attest the authority of a particular individual to gain access to a specific resource while using different devices. In location-to-location identity, mobile identity can be used in the certification process to attest the authority of a particular individual to gain access to a specific resource/service while moving in a different geographic area. In context-to-context identity, the device may need to “select” the most preferable identity depending on the context – e.g. the people in their proximity. The interactions between different types of mobile business and mobile identity management are described. In addition, the identity in next generation mobile systems is also analyzed.

FIDIS project gives a comprehensible, technical survey on mobile identity management, focusing particularly on security and privacy issues for mobile users (WP3 2005). The study investigates the need for mobile identity management and derives requirements for mobile identity management from exemplary scenarios. Privacy threats for mobile users and usability of mobile identity management systems are both taken into account.

These papers listed above provide an overall picture of the field of mobile identity management, which includes the definition, common threats, requirements and recommendations. But so far, there are very few implementations of secure mobile identity management solutions.

2.3 Overviews of Mobile PKI

WAP Forum designed a WPKI protocol suitable for wireless environments (WAP Forum 2001). The solution comprises PKI portal, registration authority (RA), Certification Authority (CA), PKI directory, and end-entity application (EE). PKI portal acts as a Gateway Server, which is used to transform the requests from WAP clients to RA and CA servers. Instead of a standard X.509 certificate, the URL of certificate is used by WAP client. By using the URL of a certificate, other entity can download the certificate in order to verify client’s signature. The connection between WAP client and PKI portal is protected by WTLS and the communication between PKI portal and back-end servers is protected by wired SSL/TLS. It is regarded as a standard for PKI solution in wireless environments. However, as Trask and Jaweed pointed out, the implementation of WAP PKI met a lot of obstacles, which makes it difficult to be widely deployed (Trask and Jaweed 2001).

Inkyung and Kilsoo proposed a mobile PKI model for ubiquitous environments (Inkyung and Kilsoo 2008). They analyzed the importance of protecting private keys and designed a model to transfer user’s private key from a PC’s hard disk to mobile phone and to perform PKI service using SMS. This solution does not require mobile device to be able to generate public key/private key pair and perform procedure of creating certificate request, since both the key pair and the certificate are loaded from PC into user’s mobile phone. However, in this way, user can not request and use PKI directly without the PC and the PC S/W, where the key pair and certificate are...
stored. In addition, it is not secure to transfer private keys, since attacks may be directed to that operation. Private key should never leave its generator.

Trichina proposed a SIM-based mobile payment system based on nation-wide PKI, namely the Finish Electronic Identity (FINEID), as shown in Figure 5 (Trichina, Hyppönen and Hassinen 2007). It is an operator-agnostic technological solution to secure mobile payment transactions that can be utilized by financial institutions, mobile network operators, and independent third parties. Mobile operator issues PKI-SIM cards to the customers, containing two private keys generated on a card and a FINEID SAT applet used for customer authentication and signature. The corresponding public-key certificates are produced and officially registered in the FINEID directory, when the customer registers his/her SIM card at a police station. By using digital signatures, generated by FINEID SAT applet and the corresponding public-key certificate, service providers can authenticate customers and provide non-repudiation of transactions.

Figure 5. Public Key Platform Implemented in Finland

Nexus published a white paper, discussing the challenges and explaining methods for establishment and use of a mobile PKI (Nexus n.d.). It is pointed out that the challenge lies in the enrolment process for trusted key distribution and gave one example of enrolment scenario, as shown in Figure 6. As they mentioned, this is not the only possibility. In some cases it may be favorable to use an operator-independent mobile PKI solution. For instance, a service provider could offer this service and build up a mobile PKI with an external trust center or even the trust centers themselves.
Lee proposed a wireless PKI suitable for mobile phones (Lee, Lee and Song 2007). Their solution minimized data sizes processed in mobile phones and optimized protocols for certificate management and verification between mobile phones and server. ECC-based Elliptic Curve Digital Signature Algorithm (ECDSA) was used for generation of key pairs, which reduced the certificate size. Online Certificate Status Protocol (OCSP) instead of Certificate Revocation List (CRL) was used for validation of certificates, which reduces the computational work for mobile devices. The model of their WPKI solution is shown in Figure 7. They also evaluated the proposed solution by comparing the processing time of RSA and ECDSA and certificate and module sizes.
These works described above showed the current trend of mobile PKI and provided feasible solutions. The solution we proposed combined the advantages of these existing works and makes it easier and secure for mobile users.

2.4 Overviews of Mobile Authentication, Access Control and Authorization

As described in Chapter 1 the number of attacks, such as fraud or theft of identities, is exploding dramatically. Therefore, authentication and authorization for mobile transactions are becoming more and more important. Most commonly, these schemes depend on three factors: what you know (secret), what you have (token), and what you are (biometrics). According to the U.S. National Institute of Standards and Technology, there are four levels of authentication (National Institute of Standards and Technology (NIST) 2011):

- Level 1: No identity proof
- Level 2: Single factor remote network authentication
- Level 3: Multi-factor remote network authentication
- Level 4: Proof of possession of a key through a cryptographic protocol

Level 3 is normally considered as strong authentication. In terms of the authentication in mobile transactions, the biggest issue is due to the resource limitations of mobile devices, as mentioned in Chapter 1. Therefore, the authentication mechanism should be efficient and less computative.

Khan proposed an origin authentication of digitally signed messages using joint-signature scheme for mobile commerce (Khan, Khiyal and Ayub 2011). Joint-signature scheme was proposed by He and Zhang in 2004 (He and Zhang 2004). As shown in Figure 8, the message is digitally signed by message originator with the help of its network operator (message signer), both jointly produce the signature which is going to be verified by the service vendor (message verifier). This technique is more efficient than other traditional schemes. In comparison with some existing techniques, mainly server aided scheme (Chen, et al. 2007) and secure one-way mobile payment mechanism (Ham, et al. 2002), this technique overcomes all major disadvantages of existing technologies.

Lam proposed a lightweight security mechanism for mobile commerce transactions, which provides a secure means for authenticating end users and key exchanges (Lam, et al. 2003). Their solution is based on a trusted wireless protocol gateway and PIN, known only to the user and verifiable by gateway.

Generally, there are very few research results about authentication and authorization for mobile transactions. These are point solutions and there is lack of a systematic approach providing a comprehensive and light-weight way of authentications and authorizations. Since mobility is the most important feature of mobile transactions, location-based authentication and authorization are quite useful.
Denning were among the first authors to perform research studies about location-based authentication and to highlight its importance for improving network security (Denning and MacDoran 2006). In their paper they argued that location information can be used for both preventing network breaches and also for facilitating investigations in cases when breaches occur. In a virtual environment where physical borders are blurred, location determination during authentication can be helpful in many scenarios e.g. remote access to critical systems, authenticating financial transactions, enforcing export controls on software, and so on. They describe a technology by the International Series Research in USA, called CyberLocator, which is used to achieve location-based authentication by using what is called a location signature. A client that wants to access a protected resource is challenged to provide a location signature, which is then verified by the server. The server does this by also computing its own location and comparing it to the one provided by the client. Since the location signature is unique for each location at any given time, this information cannot be spoofed or replayed later. However, in order to achieve this, CyberLocator needs its clients to possess a special kind of GPS sensor that is different from the ones that are commercially available.

Cho propose a location–aware access control mechanism (LAAC) based on a WLAN infrastructure of wireless access points and wireless mobile devices, such as PDAs and wireless laptops (Cho, Bao and Goodrich 2006). Access is granted to a device located inside a region formed by overlapping coverage of multiple access points. Each access point periodically broadcasts a random nonce which is captured and used by the device to generate a location key. Devices outside the range of the access points won’t be able to receive these random nonces and consequently won’t be able to derive valid location IDs. In this way access is granted only within specific locations. Bao proposes similar mechanism using wireless access points (Bao 2008). His system is known as LENA (Location Enforced Network Access). LENA has two schemes, one known as LENA-SK (LENA using Security Keys) uses Diffie-Helman key exchange protocol to authenticate user location, authorize network access, and distribute a key for data encryption. The other scheme, LENA-PAP (LENA using Personal AP Protocol) uses mobility management protocol to ensure
authenticity of location claims. These mechanisms are designed specifically for controlling access to wireless networks.

Jansen designed a location–based authentication mechanism that involves policy beacons and mobile devices (Jansen and Korolev 2009). These policy beacons broadcast and communicate location data to mobile devices using Bluetooth. Mobile devices determine their proximity to beacons and calculate their location relative to them. Based on this location certain functionalities in the mobile devices are enabled or disabled accordingly. Policy beacons establish a perimeter with a distinct organization policy. Devices within this perimeter inherit this policy. Their setup, however, focuses only on controlling the use of mobile devices, especially in an environment such as in an organization and it requires a significant costly infrastructure setup and synchronization of policy beacons.

Takamizawa and Kaijiri proposed and designed an authentication method using location information obtained from mobile telephones suitable for web-based education applications (Takamizawa and Kaijiri 2009). A student who wants to login into the web-based application, in addition to using username and password, has also to provide his/her location through a mobile telephone in order to prove his/her authenticity. In their method, location from a mobile phone is determined using GPS. For that, mobile phone must be equipped with a GPS receiver and a clear view of the sky is needed for the process to work. QR codes are also used for web applications to prompt mobile phone for the location. The user has to scan the code from the screen using his/her mobile phone and therefore a phone needs a camera. In addition, the authors did not pay attention to security threats and vulnerabilities for their location–based authentication method and as such the mechanism may be susceptible to trivial attacks. For example, the location could be easily spoofed or modified.

Ardagna analyzed how location information can be used to strengthen access control mechanisms by adding features for defining and enforcing location–based policies (Ardagna, et al. 2009). They proposed design of a Location–based Access Control (LBAC) architecture and provided an extension to the XAMCL policy language (introduced by the Open Geospatial Consortium – OGC) for defining and describing geographic location coordinates. This extension is known as GEOXAMCL. They showed examples of how this can be used to express access control rules that can be used in a typical application.

2.5 Descriptions of Some existing Mobile Payment Systems

There are large numbers of companies putting great efforts on mobile payment services. This section gives a brief overview of some of these systems.

2.5.1 MobiPay

MobiPay is the first mobile payment solution in Namibia. It provides money transfer, purchase of airtime and electricity, point of sale payment and bill payment services. Their solution is based on the usage of voice and/or SMS (or Unstructured Supplementary Service Data (USSD)), which is easy and convenient for mobile
clients. A virtual account is generated for each mobile customer and a PIN code is used for customer authentication (MobiPay n.d.).

2.5.2 SEMOPS

SEMOPS (Secure Mobile Payment System) combines mobile technologies with legacy banking infrastructure, as well as their security solutions to provide mobile transactions, including real time account-to-account transfer, P2P payment, payments to unbanked with payout through agents or ATM terminals, mobile purchase, bill payment, as well as payments at POS terminals. Their solution is based on traditional four corner business model: payer, payer’s payment processor, payee and payee’s payment processor. And the security and trust are achieved in such a way that the payers and payees are in relation only with their own trusted partners, their own banks or MNOs, and there is no additional third party involved in transactions. In this way, the payers and payees do not identify themselves to each other, what enhances privacy and trust. In addition, all communications of sensitive data is encrypted using 1024 RSA mechanism and the same method is used for digital signature. User authentication is performed by payment processors using their own legacy systems and methods: PIN, password, OTP, digital signatures and so on (SEMOPS n.d.).

2.5.3 SmartMoney

SmartMoney is a system providing bank account transactions in Philippines. It allows banked customers to use their mobile phones to conduct bank account transactions, such as money transfer and money withdraw/deposit. PIN is set by the user during registration phase and used later for authentication purposes (Smart Communications, Inc n.d.).

2.5.4 PayPal

PayPal developed a mobile system for managing PayPal accounts and performing transactions, such as payment and transfer money, with the PayPal accounts. These transactions are provided by PayPal originally for online customers. With this mobile payment system, transactions can be performed using mobile phones. It utilizes both Web and SMS as communication protocols (PayPal n.d.).
Part II describes a new secure service-oriented architecture for mobile transactions. Chapter 3 explains the architecture, including trusted stack model and its components. Chapter 4 to Chapter 7 explain four types of security services whose design is based on the proposed architecture: registration and identity management service, mobile PKI service, mobile authentication and mobile authorization service, and secure messaging service.
Chapter 3: A Model of Secure Mobile Service-Oriented Architecture

This Chapter describes a new secure mobile SOA, which comprises four parts: introduction of approach, trusted stack model, the proposed secure SOA and the brief descriptions of the components and services.

3.1 Introduction and Approach

According to the concept of service-oriented architecture (SOA), there are three basic components: services, service provider, and service consumer. Services are business functionalities, built as software components that can be reused. Service provider creates services and publishes the interfaces or access information to a public registry. Service consumer locates these registered and published services in order to invoke one of these services. These three components interact with each other through well-defined and platform-independent interfaces, protocols and messages.

As mentioned in Chapter 2, most current solutions of adopting SOA into mobile focus on web service provisioning on mobile devices, which are difficult for large-scale deployment. And the security for these architectures is not adequate. Therefore, we propose a SOA for mobile environments, which is secure, flexible and extendible for large-scale deployment and various mobile devices.

We first defined the most important services in mobile environments. Based on the research problems described in Chapter 1, besides mobile application services, we assumed that two types of services are the most important for mobile SOA: communication services and security services. Therefore, we designed the architecture comprising service provider components and service consumer components that can provide and consume mobile business services as well as communication services and security services. As a result, we created a model, called **Trusted Stack Model** explained in Section 3.2.

After the design of the architecture, we designed communication services and security services, including business logic, interfaces, protocols and messages. The details of these services are described in later Chapters.

3.2 Trusted Stack Model

The proposed architecture can be abstracted as so called trusted stack model shown in Figure 9, which can be structured in the form of seven layers: **Secure Element (Chip), applets, middleware, mobile applications, communication network, message distribution server**, and **mobile services providers**. Equivalent as with OSI seven layer model, each layer provides services to its upper layer while receiving services from the layer below.
Layer 1: Secure Element (Chip)
This layer represents the chips used for mobile transactions, such as smart card, SIM/UICC (Universal Integrated Circuit Card) chip and Micro SD chip. The chip contains a microprocessor and small memory, used in the mobile phone. It is secure and tamper resistant, so that it works as Secure Element (SE), storing important information, such as credit card number and private keys. The chip stack defines how data is stored in the SE. It also defines the format of application protocol data unit (APDU) for upper stack to invoke the functionalities provided by the chips.

Layer 2: Applets
This layer represents the applets installed and functioning in the hardware of a chip. Applets interact with the chips using APDUs, invoking functions provided by chips and provides those functions to upper layer stacks.

Layer 3: Middleware
Middleware stack defines how the upper stack software applications invokes the functions provided by applets. It provides application programming interfaces (APIs), for upper layer mobile applications to invoke the functions provided by the applets. Middleware stack contains implementation of these APIs, the functionalities of which are to translate upper layer application data calls into APDUs and vice versa.

Layer 4: Mobile Applications
Mobile applications layer is closest to the end-user. It consists of secure mobile applications and mobile operating systems. Secure mobile applications interact with users and provides functionalities for users. Mobile operating systems provide APIs to mobile applications.

Layer 5: Communication Network
This layer comprises various mobile communication channels and networks, such as SMS, GPRS, Bluetooth, Wi-Fi and so on.

Layer 6: Message Distribution Server
This layer consists of communication switch and service switch. Communication switch provides communication services to mobile service consumers. Namely, it collects transaction requests, sent by service consumers via various communication channels or networks, and forwards them to the service switch. For example, communication switch connects with SMS aggregators in the front end, receiving transaction requests from communication stack through SMS. After receiving request, communication switch invokes APIs provided by the service switch to transfer those requests to the service switch. Service switch checks the data, pertinent to system-level processing associated with the transactions, translates those requests into the form required by SPs and invokes APIs of corresponding services provided by SPs. The search and publish of services, as well as the database for registered and published services, are also in this stack. SPs register and publish their services, so that whenever a transaction request arrives, service switch can find the proper and available services.

Layer 7: Application Services Providers
This layer comprises all the SPs and the implementations of service APIs, used by the switch layer of the stack. It also contains the database for transactions.
This model brings benefits both to mobile users and mobile service providers. With this model, mobile services are transparent to front-end mobile users, so that mobile users do not need to know who and where are the service providers, which reduces the work of looking for the services. And it provides multiple communication protocols for mobile users conducting mobile services, which increases user experiences. With this model, service providers can easily publish and manage their services provided by the service switch, so that it brings various service providers together, providing various large-scale mobile services.

As mentioned in Chapter 1, many attacks could happen during mobile transactions, so that protection of data becomes one of the important features of such solutions. Another more important feature is that this model provides end-to-end security for sensitive data during mobile transactions. The chip layer provides protection of locally stored data. These data is fetched from chips in the form of cyphertext,
protected by various cryptographic mechanisms, before transferring to open network environments. In addition, security service providers can be integrated with this model and provides security services to protect mobile transactions at the application level.

3.3 Secure Mobile SOA

Based on the above mentioned trusted stack model, we designed the concept, components and services of a large-scale, comprehensive architecture for secure mobile applications and transactions. We claim the following properties of the model:

- **comprehensive**, i.e. many security and usability features are provided by the architecture;
- **scalable**, it provides the possibility for interlinking of mutually independent deployments, if based on the described architecture;
- **modular**, i.e. new services, functions and components can be easily added to the architecture;
- **expandable**, i.e. mobile applications can easily be linked to the architecture and can utilize its services; and
- **open**, meaning that integration of new components is based on utilization of standard-based Web services and interfaces.

This secure mobile SOA solves all the issues mentioned in Chapter 1. In order to verify these goals and properties, a prototype system was implemented based on the designed model described in section 3.2. It is a 6–tier architecture, as shown in Figure 10.

The first tier (the first group of components) comprises various client components: browser access to the system, PC–based or device–based Point–of–Sale (PoS) applications, and Secure Mobile Client (SMC) running on mobile phones, as SIM/UICC applets or as smart cards applets.

The second tier is various mobile networks and corresponding communication protocols: large–area networks (Internet or GSM/3G) and proximity networks (Bluetooth or NFC).

The third tier is communication components: there is one component for each of the communication protocols provided by communication networks. These components support communication services at the network level – sending and receiving messages. Mobile Communication SP works as the communication switch, providing communication services at the application level – analyzing and dispatching messages to various Mobile Application Servers.

Behind the communication server, as the fourth tier, are various Mobile Application Servers. Figure 10 shows four of them: Payments Server – supports mobile financial transactions, Air–Time Server – supports payments and management of air–time using mobile phones, Ticketing Server – supports inquiries, purchase and dispatching tickets using mobile phones, and Parking Server – supports payments for parking using mobile phones. Those four Mobile Application Servers are only examples,
since other Mobile Application Servers can be easily added to the architecture. Those application servers are the instances of application services providers, described in Section 3.2.

The fifth tier is various back-end (“native”) servers supporting appropriate mobile applications and services: Bank IT servers support mobile financial transactions for banked users, telecom servers support administration and use of air–time, ticketing and parking servers support corresponding mobile applications and transactions.

The final sixth tier is various security servers, providing security services at the application-level for mobile transactions. The details of those security services are described in later Chapters.

3.4 Service Providers and Services

As shown in Figure 10, there are four categories of components: client components, communication components, application components and security components. These components cooperate with each other to provide various services for mobile transactions. This section contains description of the functions of those components and services. The details of each service are explained in later Chapters.
3.4.1 Mobile Service Consumer – Secure Mobile Client (SMC)

Client components include all entities which consume mobile services through various transactions, mobile devices used by those entities to perform mobile transactions, and the application platform running on mobile devices. They comprise layers 1 to 4 of the trusted stacks. There are various types of clients that consume services. For m-Commerce, for example, those are customer, agent, and merchant. Transactions or actions are strictly controlled based on their roles. The role also affects authorization policies, explained in later Chapters. Clients may use different types of devices to conduct mobile transactions. For example, a customer can not only use mobile phone to check his/her account balance, pay bills and so on, but may also use a smart card to pay over the counter. And for merchants, a PoS device can be used to accept payments over the counter.

As one of the most important client components, mobile applications interact with clients at the front-end and connect with mobile service providers at the back-end via various mobile communication channels, such as SMS, GPRS and Wi-Fi. Each mobile application comprises two components: generic service consumer (GSC) and specific service consumer (SSC). GSC provides common functionalities, such as connecting with networks, displaying messages and security features, while SSC provides specific functionalities, different for each application. For example, it provides payment function for Wallet application and registration of customers for Agent application.

GSC contains three modules:
- **User interface (UI) module** – invokes services for interacting with users, such as getting input from a user and displaying output to user;
- **Communication module** – consumes communication services provided by communication services providers, such as to connection open network (GSM, Bluetooth and 3G) or establish connection to local components, such as SIM/UICC chips;
- **Security module** – provides security services for mobile applications, such as encryption and decryption of data and processing of digitally signed messages.

SSC contains one module:
- **Business logic module** – invokes services provided by application services providers, specific data and messages, coordinates with other modules and functions, and delivers application services to mobile users.

3.4.2 Communication Services Providers

Communication services providers comprise multiple providers at the network level, called Communication Modules (CM) and one provider at an application level, called Message Dispatcher (MD), as shown in Figure 10. Each CM provides one type of communication services. They export their services to client components at the front-end and consume services provided by the MD at the back-end. Communication services can be instantiated in different forms, such as SMS, GPRS, Bluetooth, Internet and NFC, and can be accessed using connection service interfaces for mobile phones.
Each communication module exports two services: `EstablishSecureSession` and `SecureMessageExchange`. `EstablishSecureSession` service opens connection for all the supported communication channels and waits for the connection from client components. Upon successfully establishing the connection, `SecureMessageExchange` service starts to receive messages from clients and perform security checks, which are predefined between client components and CMs.

When CMs receive service requests via various communication channels, they create uniform request messages, whose formats are pre-defined and agreed between CMs and MD, and deliver them to the MD.

MD functions as application services broker, connecting with various CMs at the front-end and with different Mobile Application Servers, such as Payment Server or Ticketing Server, at the back-end. Communication channels between MD and Mobile Application Servers, as well as the message rules, are registered during initialization phase and managed by the MD to deliver messages, received from client components, to the back-end application services providers. Each of these communication channels corresponds to one type of mobile application or service. For example, the channel to Mobile Payment Server is for exchanging messages related to mobile payment transactions and the channel to Mobile Ticketing Server is for exchanging messages related to mobile ticketing transactions. Each message rule corresponds to one specific mobile transaction. For example, a message rule can be defined that the message starts with “AS” (Account Status) should go to Mobile Payment Server for checking the account status.

When MD receives uniform service requests from CMs, it parses them by analyzing the headers of messages. (The details of message syntax are explained in Chapter 7). After parsing the request, it delivers the message to the corresponding application server through the communication channel, based on the corresponding message rule.

Mobile Application Servers represent application service providers. Each application server manages one type of services. Therefore, the registration and management of services are those application servers. For example, mobile payment-related services provided by one or more SPs are performed by Mobile Payment Server. Upon the application server receiving service requests from the MD, it looks for the proper and available service among all those that are available and creates service request, invoking the service APIs provided by the corresponding SP. The syntax of such messages is based either on some international standard, such as ISO-8583, or on specifications from native SPs, if they have such specifications. Basically, Mobile Application Servers have two roles: they provide mobile services and at the same time they consume services provided by the back-end native services providers.

By using the communication services described above, client components do not need to know where are the SPs and how to connect with them and consume the services provided by them. The only necessary step is to connect with anyone of the CMs, which simplifies functioning of client components.
3.4.3 Security Service Providers

In this dissertation, four groups of security services are designed and explained: registration and identity management services, m-PKI and certificates management services, smart cards management services, and authorization services. This section describes the overview of each of these groups of services. The details of these services are described in Chapter 4 to Chapter 7.

In order to provide secure environment for mobile transactions, several security SPs are designed and integrated under the secure mobile SOA to provide the full scope of security services for protection of mobile users, transactions, applications and service data, both stored in the database and transmitted in the open environments. These security SPs, as well as protocols and messages, are defined in a standard manner of XML-encoded SAML assertions, which are compliant with SAML V2.0 standard (OASIS 2008). In this way, it is easy and secure for other components to identify and interact with security SPs. The following is an example of IDMS SP:

```xml
  <ds:Signature>
    ...</ds:Signature>
  <IDPSSODescriptorWantAuthnRequestsSigned="true">
    <KeyDescriptor use="signing">
      <ds:KeyInfo>
        <ds:KeyName>safe.com SSO Key</ds:KeyName>
      </ds:KeyInfo>
      <NameIDFormat>
        urn:oasis:names:tc:SAML:2.0:nameid-format:X509SubjectName
      </NameIDFormat>
      <NameIDFormat>
        urn:oasis:names:tc:SAML:2.0:nameid-format:permanent
      </NameIDFormat>
      <NameIDFormat>
        urn:oasis:names:tc:SAML:2.0:nameid-format:transient
      </NameIDFormat>
    </KeyDescriptor>
    <SingleSignOnService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      location="https://safe.com/IDMS/SSO"/>
    <SingleSignOnService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      location="https://safe.com/IDMS/SSO"/>
    <saml:Attribute
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
      FriendlyName="ClientName">
      <saml:AttributeValue>customer</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
      FriendlyName="ClientMobilePhoneNumber">
      <saml:AttributeValue>agent</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
      FriendlyName="ClientRoles">
      <saml:AttributeValue>merchant</saml:AttributeValue>
    </saml:Attribute>
    <IDPSSODescriptor/>
    <AttributeAuthorityDescriptor
      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
      <KeyDescriptor use="signing">
        <ds:KeyInfo>
        </ds:KeyInfo>
        <ds:KeyName>safe.com AA Key</ds:KeyName>
      </KeyDescriptor>
    </AttributeAuthorityDescriptor>
  </IDPSSODescriptor>
</br:EntityDescriptor>
```
Those SPs consume and export services from/to each other:

- **Identity Management Service Provider** is used for storing identity information of all the components and providing identity management services within a security domain, such as identity registration, identity verification, identity suspending and identity blocking. It implements and exports the following services for other entities to invoke:
  - **RegisterIdentity**: accepts registration data from clients
  - **RegisterIdentityViaAgent**: performs user registration by System Agent
  - **ConfirmRegistration**: confirms registration of entities
  - **FetchIdentity**: allows other entities to fetch identity data from database
  - **UpdateIdentity**: allows entities to update existing data stored in database
  - **VerifyIdentity**: verifies the identity of a specific entity
  - **StrongAuthentication**: performs strong authentication protocol based on FIPS 196 standard (National Institute of Standards and Technology (NIST) 1997).

- **Federated Identity Management Service Provider** provides registration and identity management services for domain level IDMS SPs and interoperates with other FIM SPs to form a federation, which provides cross-domain and cross-country identity management service. It implements and exports the following services:
  - **UnilateralRegistration**: registers domain level IDMS SPs
  - **MutualRegistration**: registers other FIM SPs
  - **UpdateReference**: accepts the updated information from other FIM SPs
  - **Deregistration**: deregisters domain level IDMS SPs
  - **DomainInquiry**: provides information about inquired domain level IDMS SPs
  - **FederationInquiry**: provides information about inquired infrastructure level FIM SPs
  - **FederationVerification**: performs federated identity verification
- **Strong Authentication**: performs strong authentication procedure based on FIPS 196.

  - **m-PKI Service Provider** is the core of m-PKI service. It interacts with CA Servers to provide certificate management services for client components, such as request certificate, revoke certificate and verify certificate. The involvement of m-PKI SP makes the m-PKI solution compatible with other existing PKI solutions, what improves flexibility, compatibility and adoptability. m-PKI SP implements and exports the following services to other entities:

    - **RequestCertificate**: creates certificate request message based on certificate requestor’s information
    - **RevokeCertificate**: creates certificate revocation request to revoke a specific certificate
    - **VerifyCertificate**: interacts with CA to verify certificates

  - **Authentication Service Provider** provides authentication services for entities involved in mobile transactions. It implements and exports the following service:

    - **StrongAuthentication**: performs strong authentication procedure based on FIPS 196.

  - **Authorization Service Provider** provides authorization and access control service according to the authorization policies before delivering resources or services and as such it guarantees that the resources or services can be accessed only by authorized entities. It implements and exports the following services:

    - **Role-based Authorization**: performs authorization based on client’s role

  - **Location Service Provider** – provides location-based services, such as check-in/out a location and location-based authentication and authorization for GPS-enabled mobile clients. This is mainly used to improve security based on mobile user’s location, which is especially useful in mobile environments. It implements and exports the following services:

    - **Location Registration**: registers mobile client’s location information into the database
    - **Location-based Authentication**: performs authentication protocol based on client’s location information
    - **Location-based Authorization**: performs authorization procedure based on client’s location
Chapter 4: Registration and Identity Management Mobile Services

This Chapter describes the registration and identity management services for mobile users. The architecture, functionalities of components, and three types of services: registration service, Federated Identity Management (FIM) service, and identity verification service are described. In addition, the evaluation of the proposed solution for its security and usability is described.

4.1 Overview of Registration and Identity Federation Mobile Services

In the current IT environments various systems, such as Web technologies, social networking, Email and financial systems, provide identity management services using conventional methods. Most of them use proprietary models and protocols for identity management, which are very complicated to extend and inconvenient for mobile users. Furthermore, most of them provide only limited security services and do not address privacy of identities and protection of users’ data stored at IDMS servers. Security and privacy issues are the most important issues that require to be addressed. In this paper, the authors mentioned that the exchange of user’s identity information between IdPs and SPs could reveal complete user’s identity and activities, since users not knowingly shares information between potentially malicious IdPs and SPs (Khattak, Sulaiman and Manan 2010). Therefore, there are many security and privacy issues in the current FIM solutions, such as fraud, identity theft, and identity abuses, which cause serious damages. According to the report (Public Safety and Emergency Preparedness Canada 2004), “during one-year period in 2002-2003, total losses to individuals and businesses related to identity theft in the United States were estimated at around $53 billion. In Canada the losses for 2002 were estimated at around CAN$2.5 billion”. The authors presented a threat model for FIM systems, which considers identity theft, misuse of identity information via single sign-on facility in IdPs and SPs, trustworthiness of subject, IdPs and SPs as the active concerns in FIM systems.

As mentioned in Chapter 2, there are few implementations for secure mobile registration and identity management services. Based on our proposed architecture and the property of mobile transaction environments, we designed a mobile registration and identity management solution.

In mobile environments, mobile phone number is one of the most convenient identifiers of mobile users, since every mobile user has at least one unique mobile phone number. It is suggested by industry consortia that mobile phone number should be used as Mobile Identifier (m-ID), which means that whenever someone wants to perform mobile transaction, they only need to know mobile phone number of the other entity (Mobey Forum 2003). There are several advantages of using mobile phone number as m-ID:
- It is convenient for people to perform transactions, since they already communicate using phone numbers;
- It can protect people's financial information, such as credit card number, since people do not need to provide sensitive information during mobile transactions. It is also more secure for servers rather than mobile applications to keep sensitive data;
- It can achieve interoperability, since mobile phone numbers follow standardized topologies and are based on global routing and roaming standards.

Other attributes, which are widely used in the Internet environment, like Email addresses, can also be used to identify mobile users. However, in that case, mobile services can just utilize mobile data network, such as Wi-Fi or 3G, but not mobile telecom networks, such as SMS. What's more important is that other identifiers are easier to get attacked, such as identity fraud, than mobile phone numbers.

Another issue is that there are many types of mobile devices and mobile platforms. Users may also use different communication technologies for different scenarios. Therefore, a generic identity management system, which must address security, privacy and interoperability issues in a wide range of mobile devices, is needed.

In this research, we analyzed current identity management systems and requirements of such systems for various applications and devices and concluded that:

- (a) Identity management systems should be generic in order to support multiple mobile applications;
- (b) They should support various communication technologies;
- (c) All information related to identities should be kept in a strongly protected environment;
- (d) They must support federated environments;
- (e) They should be easy to use even with simple mobile devices; and
- (f) They should be flexible and easy to extend and integrate with various mobile applications.

Based on the above requirements, we designed a secure FIM system for mobile environments using new methodology and mobile technologies. The details are described in this Chapter.

All entities involved in mobile transactions must first be registered. This is performed as the first step of the system. The system provides not only registration service for users and service providers, but also identity management service. It supports various services required for identity management in a single domain, and in addition, it also provides an infrastructure for federated identity management systems, verification of identities in local and federated environments and so on. The following are the distinctive features of our solution:

- Generic Security Objects – design of the identity management system is based on generic security objects;
- Generalized Message Format – we standardized XML message formats for various services and devices, like user registration, IDMS SP registration with
the central infrastructure-level SPs, and verification of identities in local and global environments;
- **Generic Communication Protocols** – communication protocols are generic, so devices transparently select communication protocol;
- **Identity Federation** – different domain-level IDMS SPs are federated using infrastructure level global SP, so users in one domain can securely access resources and use services in another domain;
- **Authentication** – users are authenticated across multiple domains or organizations;
- **Security** – identity information is kept in a secure storage, so only authenticated and authorized users can access it; and
- **Handling of Security Credentials** – identity management services transparently handle the required security credentials based on applied policies.

### 4.2 Layered Architecture for Identity Management Mobile Services

Cross-domain identity management is very important for mobile environments, since mobile services consumers usually move from one place to another. Therefore, we used a hierarchical approach for cross-domain and cross-country identity management. As shown in Figure 11, the proposed mobile identity management architecture comprises three layers of components.

The bottom layer represents service consumer components, i.e. mobile users, who are directly participating in mobile transactions. The second layer is a domain layer, comprising domain-layer IDMS SPs, which are responsible for actions to register and manage identities of service consumer components and resources within a domain. The third layer is infrastructure layer comprising infrastructure-layer federation SPs, which are responsible on one hand to register and manage domain-layer components and on the other hand to route cross-country messages. They also coordinate with other infrastructure-layer SPs to form a federated infrastructure.
4.2.1 Services Consumers Layer

Every user involved in mobile transactions must perform registration as the first step. We consider users as being service consumers. As mentioned in Section 4.1, the most important attribute for each client component is mobile phone number. Mobile number is used as the unique identification number to identify the client. Additional registration information is associated with that primary identifier. Therefore, each user must submit his/her mobile phone number in order to register in a mobile transaction system. Some additional attributes can be introduced to associate with the mobile phone number, such as names, roles and address.

Each component comprises not only entity, but also an SMC, which is described in Chapter 3. SMC is located between client entity and back-end mobile transaction systems, helping clients with registrations. It provides user interface to collect client registration data, sends it to the system, and displays response from servers. The SMC also securely stores user’s identity data in the SE for later authentication and authorization purpose.

4.2.2 Domain Layer Service Providers

Domain-layer components comprise domain-layer IDMS SPs, which provide registration and identity management services for all the entities involved in mobile transactions within a domain. These components store consumer’s identity data, provide interfaces for other components to fetch those identity information for authentication purposes, and connect with infrastructure layer components for cross-domain identity federations.

4.2.3 Infrastructure Layer Service Providers

Infrastructure-layer components include federation SPs, which provide registration and identity management services for domain layer components. Federation SPs share domain information with each other in order to achieve cross-country identity federation.

4.3 Mobile Registration Services

FIM architecture, shown in Figure 11, provides two types of registration services. The first one is Consumer Registration, which is used for service consumer registrations with domain layer IDMS SPs. The second one is Domain Registration, used for domain layer IDMS SPs’ registrations with infrastructure layer federation SPs.

4.3.1 Consumer Registration

All entities involved in mobile financial transactions must be registered. From the system’s point of view, the system could import registration data from other existing databases, which should be secure and reliable.

Our system provides secure and flexible registration services. As explained in Section 4.1, mobile phone number is good identifier in mobile environments. A unique ID
(the format of the ID is explained later) is generated during the registration phase and used as the m-ID to identify entities involved in mobile transactions. Each m-ID is linked to user’s mobile phone number. In this case, the system provides not only advantages mentioned in Section 4.1, but also anonymity of service consumers. In order to register, user needs to provide at least his/her name (first name and last name) and mobile phone number. Other identity attributes may be included.

The system supports two types of registration services: simple registration and full registration. Simple registration service is based on SMS messages, which is especially useful for areas without any other communication networks except GSM network and for users who are using mobile phones that support only phone calls and SMS functions. It works as follows: service consumer sends Registration_Request SMS message containing client’s First Name and Last Name to the system’s SMS CM, which forwards it to the MD. The MD checks message type, which is in this case Customer_Simple_Registration and invokes the RegisterIdentity service provided by the domain IDMS SP. This SP stores service consumer’s identity data, generates m-ID and System PIN and returns them to the MD, which delivers Registration_Response to the SMS CM as an XML message containing data received from domain IDMS SP. SMS CM sends SMS message, including m-ID and user’s System PIN, to the consumer’s mobile phone. In that moment, consumer’s registration status is “Pending” and the consumer is not allowed to perform any mobile transaction until his/her registration data is confirmed, when registration status becomes “Completed”. Service consumer completes registration process by sending additional Registration_Confirm message. The advantage of this service is that it is easy to use without any hardware and software requirements. The disadvantage is that this protocol does not provide any application-level security, since no application is loaded and used to perform registration in the consumer’s mobile device.

The other full registration service is more complex, but it provides more security features than simple registration. Its process consists of two phases: the first phase is submitting registration data and the second phase is face-to-face verification process, performed by the System Agent. System Agent is a trusted entity that has been certified and authorized by the system administrator. Examples of agents are banks, post offices and supermarkets. Initial certification phase depends on deployment requirements and is different for various deployment cases. Therefore, it is not described in this thesis. Every System Agent maintains several System PINs. Each System PIN is protected inside an encryption envelope with a serial number outside appended to the envelope. These PINs are generated and stored by the IDMS SPs and the serial number is an identifier of each PIN. The procedure is the same as that producing PIN code for credit cards.

In the first phase of the strong registration procedure, client sends his/her identity data and the mobile number of one of the System Agents to the CM, which forwards to the MD. The MD invokes RegisterIdentityViaAgent service provided by the IDMS SP. When the IDMS SP receives registration request, it verifies whether the client’s mobile number has already been registered or not. If it is not registered, IDMS SP sends a service response containing a randomly generated number to the consumer.
In the second phase, the consumer needs to tell the received random number to the System Agent and show the valid identity, such as ID card or passport, to the System Agent, who fetches service consumer’s information from the IDMS SP by sending consumer’s random number. After receiving user’s registration data from the IDMS SP, System Agent verifies user’s registration information by comparing it with user’s valid identity. If the verification succeeds, System Agent sends Registration_Confirm message to the MD via CM, which invokes ConfirmRegistration service provided by the IDMS SP. Registration_Confirm message is signed by System Agent’s private key. IDMS SP verifies System Agent’s signature and stores user’s identity data. After that, IDMS SP sends signed Registration_Complete service response, containing random serial number of the cryptographic envelope that includes PIN code, to the System Agent. IDMS SP also sends Registration_Complete message, containing user’s m-ID to the user. Then, the System Agent gives the envelope to the user and user sends Registration_Complete_Confirm message, which contains serial number of the envelope, to the IDMS SP.

User’s m-ID consists of three components: country code, system code and user registration number. For example, if the user is the first one registered with the No. 2 in the USA, then the user ID will be 102000001 (the assumption is that maximum number of users in one domain is 1 million). As mentioned in section 4.3.1, domain ID comprises country code and domain code, which makes it unique. Therefore, the domain ID with user code is also unique, since the user code is unique within the domain. By using this identifying mechanism, users are able to conduct cross-domain and international transactions. PIN is used by the system to authenticate the user and it is securely stored in the SE by using SMC.

4.3.2 Service Provider Registration

The SPs also must be registered before providing any services. Each SP can use the same registration service as described above, meaning that each SP needs to have one mobile phone number and will get an m-ID after registration. This m-ID is used not only for the system to identify the SP, but also for service consumers to identify the SP, so that they can perform mobile transactions with each other. By using this approach, the SPs only need to make their m-IDs public for users, instead of publishing their mobile phone numbers, which protect the privacy of SP’s information.
4.3.3 Domain Registration Service

In order to support cross-domain transactions, each domain-level IDMS SPs must itself be registered and certified. It is achieved by introducing infrastructure level FIM SPs. In order to register domain level IDMS SPs, a Unilateral Registration Service (URS) protocol was designed. The prerequisite for the URS is that both SPs need to have their certificates. Furthermore, each message exchanged between IDMS SP and FIM SP is digitally signed using XML signature standard (W3C 2008) and is enveloped using XML encryption standard (W3C 2002). The URS protocol works as follows: IDMS SP initiates the process and invokes StrongAuthenticationService provided by FIM SP, as shown in Figure 13. After successful authentication, IDMS SP consumes UnilateralRegistrationService exported by the Federation SP. This service request contains originator’s name, receiver’s name, a randomly generated session ID, a domain ID, IP address, and port number of the IDMS SP.

The domain ID consists of two components: country code and domain code. Country code is the country number for telecom. For example, the country code for Sweden is 46 and for U.S.A is 1. There can be multiple domains within one country, so we introduced domain code for each domain. For example, if the system is installed in the U.S.A and the domain code is 6, then the domain ID is 106 (the assumption is that maximum number of domains in one country is one hundred).

When FIM SP receives service request, it verifies the IDMS SP’s signature using public key corresponding to the digital signature certificate of the IDMS SP. Upon successful verification of the signature, FIM SP opens the enveloped message and saves domain ID, IP address, and port number in a temporary storage accessible only by the Security Administrator (SA). The SA checks stored information and certifies domain ID. After that, FIM SP stores domain ID, IP address, and port number in its registration database, which is cryptographically protected to ensure protection and...
integrity of stored domain information. Certified domain ID is in the format of \textit{PKCS7SignedData} and contains verifiable domain ID. Then the IDMS SP creates \textit{UnilateralRegistration} service response, verifies digital signature, and opens enveloped data. After that, the IDMS SP stores certified ID in its database, which completes the service for domain registration.

4.4 Federated Identity Management Service

The infrastructure level FIM SPs are registered with the corresponding FIM SPs in order to form a federated infrastructure. The purpose of this infrastructure is verification of users’ identities of international and cross-domain transactions. We designed \textit{MutualRegistration} service which securely creates county-level federation.

In this service, any FIM SP can be the initiator. It invokes strong authentication service provided by the Responder FIM SP, as shown in Figure 14. After successful authentication, the Initiator invokes \textit{MutualRegistration} service, provided by the Responder FIM SP, which verifies the initiator’s signature and compares the session ID. The information contained in \textit{MutualRegistration} service is the same as the information contained in \textit{UnilateralRegistration} service request.

Upon successful verification of the signature, the Responder FIM SP stores country code, IP address and port number in a temporary storage, accessible only by the Security Infrastructure Administrator (SIA). The SIA views stored information and certifies the requested FIM SP as the trusted one. The Responder FIM SP then stores the country code, IP address and port number in its local database and consumes \textit{MutualRegistration} service provided by the requestor FIM SP.

![Figure 14. Mutual Registration Service](image)

The Initiator FIM SP verifies Responder FIM SP’s signature. After that, it processes \textit{Registration_Request} and stores country code, IP and port number in a
Management of the cross-domain federation includes exchange of newly registered domain IDs with IDMS SPs in a federation and deregistration of the domain IDs. When a FIM SP certifies a new domain or a domain SP is removed from the domain list, the FIM SP consumes UpdateReference service exported by the corresponding FIM SPs that exist in a federation to synchronize the domain name list and makes it consistent. The UpdateReference service request contains originator’s name, receiver’s name, session ID, and the list of references. Similarly, if an IDMS SP decides to leave the federation or SIA finds that a particular IDMS SP is compromised, either IDMS SP invokes Deregistration service provided by FIM SP or FIM SP directly deregisters the IDMS SP and delivers a message as acknowledgement. Corresponding IDMS SPs upgrade domain names list accordingly.

All these protocol messages mentioned above are digitally signed, based on XML Signature Syntax and Processing (W3C2008), in order to provide message authentication and to ensure integrity of the content of messages.

4.5 Identity Verification Protocol

This section explains how users’ identities are verified within one domain and in cross-domains. As mentioned in Section 4.3.2, IDMS SP issues a certified unique m-ID and a PIN code to each registered consumer. Consumer must provide correct certified user ID and PIN code to the system for verifying his/her identity. If the consumer moves to another domain (guest domain), he/she needs to provide his/her certified user ID and PIN to the guest domain IDMS SP for verification purpose. The guest domain IDMS SP checks consumer’s ID, actually the country code and system code, to see which country and domain the consumer comes from.
If the country code is the same as guest domain IDMS SP’s country code, it means that service consumer is from another domain, but within the same country as the guest domain. The identity verification service works as follows: if home domain IDMS SP’s ID is already in the Domain_Servers_List stored at the guest domain SP, guest-domain IDMS SP invokes directly VerifyIdentity service provided by IDMS SP. Otherwise, guest domain IDMS SP invokes DomainInquiry service provided by the FIM SP. The DomainInquiry service request contains home domain IDMS SP’s ID. FIM SP checks the ID and sends DomainInquiry service response, which contains information of the inquired home domain IDMS SP, such as IP address and port number, to the guest domain IDMS SP. Guest domain IDMS SP updates its Domain_Servers_List with the new IDMS SP’s ID. Then it invokes VerifyIdentity service of the home domain IDMS SP. Home domain IDMS SP verifies consumer’s information and notifies guest domain IDMS SP with the verification result sent in the VerifyIdentity service response. The inter-domain verification process is shown in Figure 15.

If country code is not the same as guest domain IDMS SP’s country code, it means that service consumer comes from the domain which is in another country. In this case, the service works as follows: guest domain IDMS SP invokes FederationInquiry service provided by the FIM SP, which contains consumer’s identity information. FIM SP verifies county code in order to obtain service consumer’s home FIM SP. If service consumer’s home FIM SP’s ID is already in the FIM_Servers_List, stored at the guest FIM SP, guest FIM SP invokes directly FederationVerification service provided by the FIM SP, which contains consumer’s identity data. Otherwise, these two FIM SPs perform MutualRegistration procedure described in Section 4.1. After successful mutual registration, guest domain invokes FederationVerification service. Home FIM SP invokes VerifyIdentity service of corresponding home domain IDMS SP, which verifies user’s information and sends verification result to the home FIM SP. Home FIM SP then sends FederationVerification service response to the guest FIM SP, which finally notifies guest domain IDMS SP about verification result. Guest-domain IDMS SP stores consumer’s m-ID in an authenticated guest ID list. Later when the consumer wants to access service or resources from the guest domain, guest-domain IDMS SP first checks the list, what saves a lot of time. International Identity Verification service is
shown in Figure 16.

4.6 Evaluation

When designing any secure system, it is important to evaluate whether it can provide expected security services. This section gives evaluation of the security of the proposed FIM system. The evaluation is based on the usage of the threat model. First a threat model was defined using attacker-based approach. We analyzed the characteristics of the identity management system from attacker's point of view, and considered five categories of possible attacks: eavesdropping, replay attack, impersonation, data tampering, and repudiation. Then we applied security services of against these threats defined by the threat model in order to verify whether these security services are capable to solve the security issues.

During system registration process, all messages are digitally signed by sender’s private key and encrypted with receiver’s public key. This prevents interception attack and replay attack. Since strong authentication protocol is expected before each session, server registration protocol is also resistant to impersonation. All data stored at the server is encrypted and can only be modified by the security administrator. Security administrator must access the server by using his/her smart card, which was issued during the initial registration phase (Zhang, Muftić and Schmöelzer 2011). This protects data from tampering.

During user registration process, one of the weakest points is the communication between client device and backend server. Because of various capabilities of mobile client devices, we designed three assurance levels depending on user’s device. If the Low Assurance Policy is applied, it means that client’s device is not able to provide extra protection except using security mechanisms provided by the device and networks. The example is using SMS or USSD. Those security mechanisms are not adequate to resist any of these four attacks since the security of GSM has been cracked. Therefore, it is not recommended to use such policy to perform high-value transactions. If the Medium Assurance Policy is used, only data integrity can be achieved. The example is using some simple mobile applications that can perform simple cryptographic functions, such as hash. Therefore, the protocol can prevent only data tampering attack. If the High Assurance Policy is applied, the abilities of client device are as strong as to support most security mechanism, such as digital signature. In this case, all messages are digitally signed with sender’s private key and encrypted with receiver’s public key. That prevents interception, replay-attack and data tampering. Before each session, the communication entities perform strong authentication in order to establish secure session, which can also prevent impersonation. The example is using the SMC designed in this dissertation.

During the identity verification procedure, user sends identity data and credentials to the guest domain server. The security of this step depends on the assurance policies, explained above. The security of all other procedures of identity verification is the same as system registration, which is High Assurance Policy. The summary of evaluation is shown in Table 1.
Table 2. Evaluation of Security of FIM Service

<table>
<thead>
<tr>
<th></th>
<th>Eavesdropping</th>
<th>Reply Attack</th>
<th>Impersonation</th>
<th>Data tampering</th>
<th>Repudiation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Registration</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High Assurance Policy</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>User Registration</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High Assurance Policy</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Medium Assurance Policy</td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Low Assurance Policy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Identity Verification</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High Assurance Policy</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Medium Assurance Policy</td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Low Assurance Policy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.7 Summary

This Chapter describes secure registration and identity management service, supported by two components: FIM SPs and domain-level IDMS SPs. In addition, three types of services: registration service, identity management service and identity verification service are designed for federated identity management. The details of the infrastructure, functions of components, services and messages are described in this Chapter. Finally, the evaluation of the security of the proposed FIM service is given, based on the attack-oriented threat model. It is helpful for analyzing and improving security of the FIM service.
Chapter 5: Mobile PKI (m-PKI) Service

This Chapter analyzes the issues and limitations for using standard PKI in a mobile environment and then explains a new concept of mobile PKI service (m-PKI) based on trusted authorities. The architecture of the m-PKI involves client components and three types of service providers in a single security domain: m-PKI SP, Communication SP, and Local Certificate Authority (LCA) SP. This Chapter describes the designed m-PKI architecture, functions of these components, lightweight certificate management services, and messages exchanged between these components. In addition, evaluation of the proposed m-PKI for its availability and security is also described.

5.1 Overview of m-PKI Services

One of the important issues for mobile commerce transactions is how can the entities, using mobile devices, verify each other's identities, exchange encrypted messages, and create and verify digital signatures for messages. PKI is a well-established approach to address this issue. It has been broadly used in wired networks for various Web-based transactions. There are three functional prerequisites needed in each application in order to be able to use PKI:

- Generate public-private key pairs;
- Parse and store certificates; and
- Generate and verify digital signatures.

There are several limitations for performing those functions when using mobile devices, which make it difficult to deploy PKI in mobile commerce environments. Those limitations can be classified in three groups:

1. Data transmission – some communication services, such as SMS or USSD, commonly used by mobile commerce systems for Over-the-Air (OTA) transactions, do not support transfer of large amount of data;

2. Data processing – most mobile devices are not capable to create standard PKCS#10 certificate request message and process the resulting PKCS#7 certificate response message that contains X.509 certificate. Therefore, it is difficult for mobile devices to apply for certificate, to receive certificates, and to verify other entity's certificate; and

3. Data storage – protection of private key is the most important issue for PKI service. If the private key of an entity is compromised or revealed, PKI service becomes useless for that entity. Since mobile devices are easily lost or stolen, it is more important to protect private keys and other sensitive information when used with those devices.

In order to solve these issues, a secure, flexible and lightweight PKI solution for mobile commerce environments was designed based on mobile trusted authorities. Mobile devices do not have to do some complex computational works, such as parsing, storing and verification of certificate. And it can be used to build secure and trusted environments for mobile commerce transactions. In addition to standard PKI functions and features, our solution also provides the following distinguished features:
• **Flexibility** – mobile users can perform PKI anytime, anywhere via various available mobile communication channels, such as Wi-Fi, SMS, Bluetooth and GPRS;

• **Adaptability** – it can be easily adapted to and integrated with existing standard PKI systems, meaning that no modifications are needed for existing PKI systems. This makes it much easier for deploying our m-PKI solution and for expanding the existing PKI systems for mobile environments;

• **Security** – besides the secure certificate management protocols, the mobile trusted authorities provide extra trust and security for PKI services, such as face-to-face identity verification during certificate enrolment process, which most existing solutions have not achieved.

This Chapter is organized as follows: Section 5.2 describes the details of the proposed m-PKI architecture, which includes two types of components: m-PKI client and m-PKI SP. Section 5.3 describes the details of two types of certificate management services: certificate request service and certificate revocation service, and certification messages exchanged between these components. Section 5.4 analyses the proposed m-PKI service from the aspects of usability and security. Summary of contributions is given in Section 5.5.

### 5.2 m-PKI Architecture

The m-PKI system was designed based on the approach and architecture described in Chapter 3. As shown in Figure 17, the m-PKI architecture comprises two groups of components: m-PKI client and m-PKI servers.

![Figure 17. m-PKI Architecture](image)

m-PKI Client is one of the security modules in SMC, which provides a secure environment on mobile devices for provisioning and storage of user's private/public key pairs and credentials. As mentioned in Chapter 3, in a mobile environment, SIM/UICC chip is suitable to be used as SE. It contains enough memory and is...
tamper-resistant to securely store sensitive data, it contains microprocessors capable to perform public key cryptographic computations, and it can be used by any type of mobile phones, already deployed worldwide. Based on those features, we have selected SIM/UICC chip as a platform for implementing m-PKI client. The implementation is in the form of an applet, which is called m-PKI Applet. The applet, pre-loaded and installed on the SIM/UICC chip, provides public key cryptographic functions, such as generating and storing key pairs, creating digital signatures and so on. The applet is compliant with Personal Identity Verification (PIV) standard (National Institute of Standards and Technology (NIST) 2006). In addition, an m-PKI Application running on a mobile phone was implemented, which provides user-friendly interface to interact with users. It provides a complete set of functions and user interfaces. It communicates with m-PKI Applet through middleware providing a set of functions, which can translate application layer’s commands into APDUs and vice versa. The m-PKI Applet, combined with m-PKI Application and the middleware, constitutes m-PKI Client.

Server components of are organized in two layers: domain layer and infrastructure layer. There are four SPs in a single security domain: m-PKI SP, communication SP, Local Certificate Authority (LCA) SP, and mobile IDMS (m-IDMS) SP. Each m-PKI SP is linked to higher level PKI components: Top CA (TCA) SP and Policy CA (PCA) SP. All SPs are managed only by Security Administrator. TCA SP provides trusted certificate to the LCA SP of each security domain, based on the security policies defined by the PCA SP.

5.3 m-PKI Client’s Functionalities

The prerequisite for using m-PKI Client is that a user has been registered and his/her identity data has been reliably stored in the m-IDMS database. The procedure is described in Chapter 4. This means that client components, especially SE, have been initialized and personalized before release. Namely, m-PKI SP should generate and issue certificates for m-PKI Applications. Public key of m-PKI SP is also preloaded into m-PKI Client. Functionalities of m-PKI Client are the following:

- Generating and securely storing public/private key pairs – user initiates m-PKI service by invoking Generate_KeyPair() function of m-PKI Applet to generate a public/private key pair and save it for later use;
- Creating digital signature – whenever user needs to send a message that requires digital signature, user invokes Digital_Signing() function of the m-PKI Applet, which creates hash of a message and generates digital signature. The m-PKI Applet encrypts the hash with user’s private key and returns the result to m-PKI Application, which sends it to other entities through mobile networks; and
- Verifying digital signature – when m-PKI Application receives signed messages, it invokes Verify_Signature() function of the m-PKI Applet to verify the signature using the public key of the message originator.

5.4 m-PKI SPs’ Functionalities

m-PKI SP is the core component. It works as a proxy between mobile clients and a standard CA SP and solves the problems that the messages exchanged in
conventional PKI solutions are not suitable for mobile devices. It takes over the burden of processing conventional PKI messages from mobile devices, such as creating PKCS#10 certificate request, parsing PKCS#7 certificate response and so on. This makes our m-PKI solution easy to be integrated with existing PKI solutions, what means flexibility and interoperability for its deployment.

The functions of Communication SP are described in Chapter 3. LCA SP issues certificates for all components within a security domain. It may be configured as a Single Certificate Authority (SCA), in which case it has self-signed certificates. Otherwise, it may also be linked to higher layer PKI SPs to establish certification infrastructure, in which case the higher layer CA SPs (TCA and PCA SPs) issue certificates to the LCA SP. Based on certificate request type, LCA SP can issue two types of certificates: standard X.509 certificates for static (Internet) users and lightweight certificates for mobile users. The details of certificate management services are described in Section 5.5.

5.5 Lightweight Certificate Management Services

The purpose of PKI is to manage certificates. Therefore, a set of lightweight CMPs were designed in order to provide PKI services. There are three services provided by m-PKI SP: certificate request service, certificate revocation service, and certificate verification service. This section describes the details of the first two services. Certificate verification service is described in Chapter 6.

One of the distinguished features is that a Certificate ID (CID), which is used as the substitution of the X.509 certificate, is introduced. It is much more convenient to transfer and store CID, compared with X.509 certificate. The CID is generated and issued by the m-PKI SP.

5.5.1 Certificate Request and Certificate Distribution Service

Certificate request service functions as follows, as shown in Figure 18:

1. Service consumer uses m-PKI Client to generate a public/private key pair and a password and sends Certificate_Request message to the Communication SP, which forwards it to the m-PKI SP. The message contains consumer's identity data and m-ID of one of the m-PKI Agents, who is authorized to verify consumer's identity. The user's identity data should at least comprise user's name and mobile number for validation purpose. The message is signed with user's private key and consumer's public key is attached;

2. m-PKI SP first verifies consumer’s digital signature with the public key attached. If successful, it sends consumer identity and a randomly generated Authorization_Code to the m-PKI Agent specified in Certificate_Request. Meanwhile, it sends a randomly generated Authentication_Code, to m-PKI Client. These two codes together are used to identify a specific certificate request;

3. When the m-PKI Client receives the message, it displays the Authentication_Code to the user. The user goes to the m-PKI Agent,
specified in Step 1, and shows his/her **Authentication Code** and valid ID, such as ID card and passport. The Agent compares the consumer’s identity, received from the m-PKI SP, with consumer’s ID and sends the **Identity_Verification_Result** message, containing verification result (match or not) and these two random codes, to the m-PKI SP. The **Identity_Verification_Result** message is signed by m-PKI Agent’s private key;

4. m-PKI SP verifies m-PKI Agent’s signature and checks these two random codes to identify a specific certificate request. If all verifications are successful, it creates standard PKCS#10 certificate request message and invokes **RequestCertificate** service provided by the LCA SP. The service request contains the PKCS#10 certificate request message;

5. LCA Server issues the certificate based on consumer’s identity and sends PKCS#7 certificate response message to the m-PKI SP, as service response. The message, containing consumer’s X.509 certificate is signed by the LCA SP;

6. m-PKI SP first verifies LCA SP’s signature with LCA SP’s certificate, loaded during initialization phase. If successful, m-PKI SP generates a CID and a directory entry in order to store the consumer’s CID as well as X.509 certificate issued by LCA SP. The CID contains signed value of the digest of m-PKI Client’s mobile number and public key. The CID attribute is lightweight compared with X.509 certificate, which is commonly used in wired networks. Then, m-PKI SP sends **Certificate_Response** message, containing consumer’s CID, signed by m-PKI SP, back to m-PKI Client. The m-PKI Client stores the signed CID as the substitution of X.509 certificate.

---

**Figure 18. Certificate Request Service**
5.5.2 Certificate Revocation Service

Revocation of public key certificates is necessary in many cases. For example, when private key is compromised or lost. There are mainly two approaches to perform revocation and verify whether a certificate is revoked. One is based on the usage of CRL and the other is using OCSP. The later approach is adopted. In this case, m-PKI SP acts as the OCSP responder. Certificate revocation protocol works as follows:

1. Service consumer sends a Certificate_Revocation_Request message that contains his/her CID, mobile number and revocation reason code to the Communication SP, which forwards it to the m-PKI SP;
2. m-PKI SP searches for consumer’s certificate based on the CID and verifies the mobile number. If verification is successful, m-PKI SP creates Certificate_Revocation message signed by its private key and sends it to the LCA SP;
3. LCA SP first verifies m-PKI SP’s signature. If valid, LCA SP revokes the certificate and sends signed Revocation_Result message to the m-PKI SP;
4. m-PKI SP verifies LCA SP’s signature and removes the revoked certificate from the local list. Then m-PKI SP sends Certificate_Revocation_Response message, signed by m-PKI SP’s private key, to the m-PKI Client as acknowledgement.

![Figure 19. Certificate Revocation Service](image)

5.6 Evaluation

This section describes the evaluations of the solution from two aspects: usability and security.

5.6.1 Evaluations of Usability

As mentioned in Section 5.1, m-PKI solution provides some distinguished features:

a. Users don’t need PCs for certificate enrollment. Instead, they use their mobile phones, which they are carrying with them every day;

b. Users can select various mobile communication protocols, such as SMS, 3G and Wi-Fi, so that users are able to perform m-PKI anytime, anywhere; and
c. Because of the use of m-PKI SP, the proposed m-PKI solution can be easily integrated with the existing standard PKI systems and it can interoperate with each other very well.

There are some prerequisites for using the proposed m-PKI solution, as mentioned in Section 5.1. Mobile devices must be able to generate public/private key pairs, and create hash value, and digital signatures.

5.6.2 Evaluation of Security

Evaluation of security is based on the threat model. We defined four possible attacks: replay attack, impersonation, data tampering, and repudiation. Then we verified whether the proposed protocols could prevent consequences of those threats. The results are shown in Table 2.

In the certificate request phase, since a session ID is randomly generated and included in each session, the replay attack can be prevented. Each message is confirmed and digitally signed by the initiator, so repudiation can also be prevented. The concept of mobile agent makes our m-PKI solution more secure, since it provides face-to-face identity verification during the certificate enrollment process. And, impersonation attack can be prevented. All messages are encrypted by recipient’s public key, so that only the correct entity can see the messages. Tampering of data is also prevented. In addition, each transaction is confirmed based on challenge-response mode and signed by the initiator. If any attack is attempted, the recipient of messages can be aware of it, so that transaction can be promptly cancelled in time.

For certificate revocation service, these four attacks can also be prevented for the reasons mentioned above. In addition, usage of certificates provides additional protection, such as integrity and authentication of messages.

<table>
<thead>
<tr>
<th></th>
<th>Replay Attack</th>
<th>Impersonation</th>
<th>Data Tampering</th>
<th>Repudiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate Request</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Certificate Revocation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

5.7 Summary

Conventional PKIs can not be directly adopted to mobile environments due to functional limitations of mobile devices and networks. In addition, security, usability and interoperability of the PKI solution must be taken into account in the context of providing secure mobile transactions.

As the result, we designed an m-PKI solution based on trusted authorities suitable for mobile environment, which includes m-PKI Client and four types of servers: m-
PKI SP, Communication SP, IDMS SP, and CA Server and three services: certificate request service, certificate revocation service, and certificate-based authentication service. Our solution not only solves the issues mentioned in Section 5.1, but also provides flexibility and interoperability for its deployment. Finally, the evaluation of usability and security is also given. Based on the evaluation results, we may conclude that the proposed m-PKI solution is easy and secure to use.
Chapter 6: Authentication and Authorization Mobile Services

This chapter describes two security services: mobile authentication, and authorization services. These security services are based on the same architecture, described in Chapter 3, and utilize the results of reliable registration, the identity management, and agent-based mobile PKI, which are described in the previous chapters. The details of the components and protocols of each service are described in this Chapter.

6.1 Overview of Authentication and Authorization Services for Mobile Transactions

While many concerns for mobile transactions are identical to the concerns of network electronic transactions, in general there are several aspects specific for mobile transactions. Among all these concerns mentioned in Chapter 2, security is the most important one. The reason is based on the fact that lots of financial and sensitive information is transmitted over the public wireless networks. Based on the properties of mobile transactions, we established that the following four security requirements must be provided in order to establish secure environment for mobile transactions:

- **Authentication** – it is very important for mobile transactions, since entities involved in mobile transactions usually can not meet each other and therefore trust must be established between the entities;
- **Confidentiality of communication** – the property that guarantees that communication between entities is protected from illegal disclosure and eavesdropping;
- **Integrity of messages** – the property that guarantees that information exchanged between transaction entities is not modified during transmission, accidentally or by unauthorized entities;
- **Access control** – different entities should have authorized access to resources and services.

Besides those requirements, other security requirements, such as privacy of communication and data, anonymity, non-repudiation, and audit may also be taken into considerations.

Therefore, a secure communication channel must be established between the transactions parties. Authentication is necessary before establishing such channel. Once mutual authentication is successfully performed, a secret key can be established between communication entities. This chapter describes authentication and authorization services for mobile transactions, which include all components and protocols. These services are secure, light-weight and suitable for mobile environments. The evaluation is also given from security’s point of view.
6.2 Authentication and Authorization Mobile Service

Authentication is one of the most important services of mobile transactions. Two types of authentication are used: local authentication and remote authentication. Local authentication is for the application running on a smart phone to authenticate user who wants to access it, which protects access by unauthorized entities. Remote authentication is for entities involved in mobile transactions, to authenticate each other, since they can not meet in person.

Local user authentication is based on usage of the PIN. User must set a PIN, which is securely stored locally either in the SE of a mobile phone or in a smart card, during application initialization phase. After that, whenever a user activates the application, PIN is required for user authentication. Protection of the PIN is the following: first hash value of the PIN is calculated using one-way algorithm, such as SHA-1. Hash value is used as the key for the AES algorithm. Then PIN is encrypted using AES and the encrypted value is stored locally, either in the memory card or in the smart card. In this way, PIN is strongly protected, so that even if the stored value is exposed, the original PIN will not be revealed.

Two approaches are proposed for remote authentication, one is based on the usage of certificates, called certificate-based authentication, and the other is based on usage of mobile client's location, called location-based authentication. This section describes the details of these two authentication protocols.

6.2.1 Certificate-based Authentication

In order to use certificate-based authentication, the entities must first get their certificates. The procedure for mobile clients to request certificates was explained in Chapter 5. Other entities, such as servers, that are not mobile devices, can request certificates using the following protocol:

- The entity which requires certificate, should already pre-load root trusted certificate during initialization phase of the system, generates public/private key pair, and sends PKCS#10 CSR message to the LCA SP;
- CA server generates X.509 certificate and sends it, together with the signed serial number of the certificate (CID), within a PKCS#7 message, back to the requesting entity;
- Entity uses root certificate to verify CA server’s certificate for the validity of the certificate issued by CA server and stores both X.509 certificate and its CID.

Once the user gets the certificate, he/she can use it for authentication before performing transactions with servers. As mentioned in Chapter 5, OCSP is implemented at the m-PKI SP for checking certificate status. Authentication protocol is based on Strong Authentication Protocol, specified in the Federal Information Processing Standard (FIPS) 196 (National Institute of Standards and Technology (NIST) 1997). It provides mutual authentication based on digital signatures and exchange of certificates. Certificate-based authentication protocol works as follows:
1. User sends Transaction_Request message through Communication SP to the mobile SP Server. The message contains session ID, user ID, and an indicator, indicating that this is an authentication request;

2. The SP server decides whether to continue and therefore it initiates or terminates the authentication process. If it decides to authenticate the user, the SP server generates a random number challenge, $R_{[sp]}$ and returns it with a session ID to the user;

3. User generates a random number challenge, $R_{[user]}$ and sends it with the following data to the SP Server: $R_{[sp]}$, session ID, his/her CID that was obtained by the process described in Section 3.1 and $S_{[user]}(R_{[user]}, R_{[sp]})$. $S_{[x]}(M)$ means digitally signed message $M$ with X’s private key;

4. After receiving the message, SP server first verifies user’s certificate by sending Cert_Verification_Request message that contains user’s CID to m-PKI SP. That SP searches for the certificate based on the CID and verifies the status of the certificate using OCSP protocol with the CA server. CA server sends verification result, as OCSP response message to the m-PKI SP. It is signed by CA Server. m-PKI SP first verifies CA’s signature using CA’s public key, which was loaded during initialization phase. Then it returns the verification result to the SP server in the signed Cert_Verification_Response message. If the verification is successful, Cert_Verification_Response message includes also user’s public key. SP Server first verifies m-PKI SP’s signature using its public key which was also loaded in the initialization phase. Then, it verifies the value of $R_{[sp]}$, whether it is the same as that retained in step 2. If all verifications are successful, it means that SP Server successfully authenticated the user. Otherwise, authentication failed. Finally, in addition to the authentication result, SP Server also returns a message that contains $R_{[sp]}$, $R_{[user]}$, SP Server’s CID and $S_{[user]}(R_{[user]}, R_{[sp]})$ to the user;

5. User first verifies SP Server’s certificate by sending Cert_Verification_Request message, which contains SP Server’s CID, to the m-PKI SP through the Communication Server. m-PKI SP does the same thing as described in step 4. User verifies the value of $R_{[user]}$, whether it is the same as that retained in step 3. If all verifications are successful, it means that the user and the SP Server successfully authenticated each other. Otherwise, it means that only the SP Server authenticated the user.

6. After mutual authentication, client mobile application in a mobile phone and SP server may agree on and establish symmetric key for encrypting session messages. The process of key sharing can be the following: client generates a message, which consists of a session key and an algorithm ID, signs the message, encrypts the signed value with SP Server’s public key and sends the encrypted message to the SP Server. After receiving this message, SP Server decrypts it with its private key and verifies client’s signature, using client’s public key. SP Server then encrypts all communication messages using the algorithm and key specified by the client.
6.2.3 Role-based Authorization

Various participants, such as customer, agent, merchant, system administrator and service providers, communicate with each other when performing mobile transactions. However, each entity must be able to authenticate and access authorized resources. To achieve authentication between communicating entities, Strong Authentication (SA) Server is used to perform Strong Authentication based on FIPS-196 standard as mentioned in Section 6.2. For authorization services an Authorization SP (also called Policy Decision Point) is introduced in order to provide authorization services based on the authorization policies stored locally. There are two authorization policies: role-based access control (RBAC) policy and location-based access control (LBAC) policy.

The RBAC policy can be defined using different policy languages and the most common one is XACML (eXtensible Access Control Makeup Language) (OASIS 2005). Mobile clients may have three different roles: customer, agent and merchant. Access rights for each role are different. RBAC is working as follows:

Initially mobile client performs strong authentication with the SA SP, shown as step 1 in Figure 21. Upon successful authentication, the SA SP returns a SAML_Ticket (OASIS 2006) to the client. SAML_Ticket is generated and signed by the Authorization SP.
When a client wants to access a resource, it sends a request for a specific resource to some Mobile Application Server. The request comprises: SAML_Ticket (Subject), the name of the resource (Object), and requested action (such as read, write and modify). Mobile Application Server receives the request and consults Policy Enforcement Point (PEP) SP. PEP creates SAML_Authorization_Request and sends it to the Authorization SP. Authorization SP evaluates the request against the policies and sends a decision back to the PEP in the form of SAML_Authorization_Response. The PEP receives the response and acts accordingly.

6.2.4 Transaction Authorization

In addition to the RBAC described above, various financial policies with rules and restrictions for mobile transactions are also applied. For example, a policy can be set only allowing transferring maximum $100 every day. By applying these financial policies, authorization of mobile transactions can be enforced. These policies are stored and managed by an Authorization SP. Every transaction is checked according to these policies during its processing. The process is the same as the RBAC protocol mentioned in the previous section, except that the RBAC is performed before processing of the transaction and the transaction authorization is performed during transaction processing.

6.2.5 Location-based Authentication and Authorization

Smart phones are gaining popularity all over the world, especially in the US, Europe and Asia (Kellogg 2011), (Gartner 2011), (IMS Research 2011). A lot of innovation is constantly occurring in the area of smart phones technology. Most of these smart phones are equipped with inbuilt GPS chips that can accurately detect the location of the user. This is evident by the explosion of location-based services, such as Google Maps, Foursquare, Gowalla and Yelp. In addition, companies such as Skyhook, Google and Apple continually improve their location detecting technologies by creating large databases of wireless access point and cell tower locations. The overall results have been the improvement in the accuracy of the detected locations. It is
now feasible to determine the location of a user within meters of her actual location. As such, the use of smart phones can be a potential solution in location-based authentication and authorization schemes. Phones may be used as devices to detect and send the location of a particular user to backend servers. This is a good solution due to several reasons:

1. It is not necessary to set up a specialized infrastructure. Instead, the mobile network infrastructure, which has been widely established, is used;
2. Compared with most exiting solutions, described in the Chapter 2, where special devices must be used for location-based services, users do not have to use specific devices rather than a smart phone, which they are carrying everywhere, every day;
3. It is easy and flexible to integrate our solution with any existing authentication and/or authorization systems. Our solution can work as a security plugin for existing systems;
4. These smart phones have stable platforms on which secure applications and services can be built on.
5. Smart phones possess a variety of location technologies, which enable locations of objects or people to be accurately determined within meters (Watzdorf and Michahelles 2010);
6. Provides application level end-to-end security for protecting location data;
7. Multiple location sensing technologies, built in the smart phones are used, which can provide better services, compared with other solutions where only single technology is used.

6.2.5.1 Architecture and Components

Based on abovementioned, we designed a solution that uses GPS technology provided by smart phones for location-based authentication and authorization. Its architecture is based on the overall SOA, described in Chapter 3. As shown in Figure 23, it comprises six components, which are combination of various service providers and related applications:

- **Location-based ID (LBID) SP** – stores location information and provides registration and verification services handling reliable user identification and location data;
- **Local Certificate Authority (CA) SP** – provides certification of all system participants, issues their certificates, manages and distributes certificates within a security domain;
- **Authentication SP** – provides accurate location-based authentication service for all participants;
- **Authorization SP** – stores and manages authorization policies and enforces location-based authorization service for all system participants, based on those policies;
- **Service Provider (SP) Server** – provides various mobile services, whose use is based on location-based policies; and
- **Location-based Client (LBC) Application** – an application running on user’s mobile device, capable to collect location information from trusted Location Providers (LP) and providing user interface to register, store and verify location data. The LBC can be integrated with SMC to provide location-based security features for mobile users and transactions.
The architecture and components are shown in Figure 22. System setup is performed by Security Administrators, establishing location-based authorization policies and certifying all system components. Initiation of the system is performed by merchants who register their locations. The system is then used by users that use LBC interacting with LBID SP and Authorization SP. After valid and successful registration of his/her current location, user submits access request to Authentication Server, which interacts with LBID SP and Authorization SP to evaluate the request based on the information registered in LBID SP and on authorization policies registered and stored in the Authorization SP. The result is sent to the targeted SP Server, which decides accordingly to either allow or deny access request.

One of the most important concerns of location-based services is security and privacy of the location information. In order to protect location information, we use cryptographic techniques, based on Public Key Infrastructure (PKI) and certificate mechanisms. The details of certification issuance and distribution in a mobile environment (m–PKI) are described in Chapter 5. It is assumed that every entity has already received certificate from the LCA SP. All certificates are issued by LCA SP in a standard way. However, certificates of users/LBC have special extension containing client’s location, which is obtained from LBID SP. In that way, the correctness of location data sent from LBC to LBID SP can be verified. All the messages exchanged between all components are digitally signed by message initiator.

6.2.5.2 Location Registration Service
Location registration is performed initially with user registration, explained in Chapter 4. Before the process begins, LBC must be downloaded and installed on a mobile device. It comes preloaded with the trusted Root CA certificate. During its activation, as the first step, user specifies his/her PIN. Then, user’s identity
After registration of identity data and obtaining his/her certificate, as the final step in this process, user registers his/her location. For that, he/she first sends location-registration request to the LBID SP. The request is signed by the user and accompanied by user's certificate. Upon receipt and successful validation of user's certificate and the request, LBID SP generates a random number, associated with the received information, and sends it back to the user.

User then activates location-registration function of the LBC and enters the received random number. The LBC captures user’s current static location and sends it together with the random number to the LBID SP. The location information is encrypted using LBID SP’s public key extracted from its certificate. When LBID SP receives the data, it first matches the received location information with the previously received user’s registration request by using the included random number. LBID SP decrypts user’s location information using its private key and stores user’s location information together with user’s identification data.

After completion of both steps, user’s registration information (identification and static location data) is stored in the database and results are sent back to the LBC which then presents a confirmation message to the user. Figure 23 below shows the overview of the registration process.

6.2.5.3 Location-based Authentication and Authorization

This is the second phase when the actual location-based authentication process is performed. The protocol is performed every time when the user requests some service from the system. The process begins when the user tries to access the protected resource (e.g. login into her account). The process is initiated by user sending a service request to the Communication SP. Communication SP directs
service request to the Authentication Server to authenticate the user. The Authentication Server sends an authentication challenge to the user. The user then responds to the challenge by sending his/her security credentials back to the Authentication Server for the authentication purpose, which can be any existing authentication protocol depending on the implementation of the particular system. For example, it can be a username/password, OTP or certificate-based challenge/response authentication.

Upon successful verification of user credentials, Authentication Server sends a location verification request to the LBC running on user’s smart phone. The LBC prompts the user to enter PIN. User responds by entering the PIN in order to authorize a location response. LBC then determines user’s location and sends it back to the Authentication Server within location response message. The location information is signed by LBC’s private key and encrypted by LBID SP’s public key.

After that, the Authentication Server delivers user’s location information to the LBID SP. LBID SP decrypts the message using its private key and verifies user’s digital signature. If the verification is OK, it compares user's location information with the location data stored in its database during registration and sends back verification result with user’s location information to the Authentication SP. If the authentication succeeds, authentication server sends authorization request, comprising user's access request and user’s verified location information, to the Authorization SP. Authorization SP verifies user’s access request by comparing subject, object, action and user’s location with entries stored in the authorization policy and sends authorization result to the Application SP Server. Finally, the Application SP Server decides whether to approve user’s service request based on the authorization result. Figure 24 shows the sequence of steps involved in the location-based authentication and authorization protocol.

6.2.5.4 Location Verification

One of the most crucial steps in a location-based authentication and authorization mechanism is the verification of the location claim provided by the user. The security
of the whole systems can either succeed or fail depending on the effectiveness of this step. The decision whether a user is authenticated or not depends on the validity of his/her presented location. When a user authenticates to the system he/she presents his/her location, which has been captured and calculated by the location sensing client. The verification algorithm then has the responsibility to check this location claim, verify its validity, compare it to specified authorized locations and make a decision whether the user is authenticated or not. The goal of this process is to prevent location spoofing and make sure that the clients are really in the locations that they claim to be at.

There are several ways in which the location provided by a smartphone can be spoofed in order to fool a location-based security system. These attacks can happen on different levels of the smartphone platform stack. As such they can be categorized as follows:

- **Spoofing on the hardware level**
  This is done via the location (GPS) module. An attacker can directly hack into the GPS hardware or module or simulate it in software and modify it to provide fake location signals to the smartphone operating system and consequently to the location-based client. In this way, the signals are intercepted at the lowest possible level before they reach the client and as a result it ends up deducing an incorrect location.

- **Spoofing on the OS level**
  This is done by intercepting and modifying the location APIs in the smartphone operating system so that they report a fake location to the client application. This can be achieved by modifying or running a modified version of an operating system which has been programmed to report the locations that an attacker desires. This kind of an attack is possible in open source type of smartphone OSs where the source code is accessible and can be modified.

- **Spoofing on the application level**
  This kind of spoofing happens directly on the location client application by modifying its source code or by intercepting and modifying the final location result that is sent to the location-based server. A fake or modified copy of the location client application can be installed in the smartphone and report spoofed locations to the server. In addition, the location results from a valid client can also be intercepted in transit and modified to desired values before they reach the server.

Another interesting attack (Anon n.d.) that targets location services in smartphones works by simulating certain environmental parameters i.e. access points, cell towers in order to fool the location modules that depend on this information to deduce the location of a device. This kind of attack however is not reliable and requires a lot of effort to carry out especially in a global scale.

A number of different techniques can be used for location verification in order to mitigate the mentioned methods of location spoofing. One is distance bounding (Chiang, Haas and Hu 2009). This kind of technique takes advantage of the physical limitations of wireless technologies to deduce and verify the location of a particular client. For example, a nearby Wi-Fi access point can collaborate with the location client in a smartphone to verify that the phone is in a location that it claims to be in. Another approach is to use IP addresses to geo-locate the client that is communicating with the server. Some research has shown that IP addresses can be
used to roughly deduce the general location of a client (Balakrishnan, Mohomed and Ramasubramanian 2009). These IP addresses are issued by mobile network operator (MNO) and cannot be easily changed by the client. Therefore, it is more difficult to spoof these IP addresses, which makes the approach more secure. This approach also has some disadvantages, including the frequent changing of client IP addresses, the level of accuracy of the location estimates and the use of proxies/gateways which may hide true locations of clients. Another location verification technique is based on network latency measurements (Katz-Bassett, et al. 2006). By comparing the latency (round-trip) time to a client with other known reference points, a locating server can use this technique to verify the location of a particular client. All of these approaches individually have their strengths and weaknesses.

Our location verification mechanism solves these problems by employing a hybrid approach. It combines different techniques and takes advantage of their strengths in order to verify the location of sufficiently with enough confidence. The approach can be considered as a “strength-in-depth” method, in which different individual methods complement each other and act as a layer of verification steps, where if one method is fooled the others can detect and prevent spoofing.

The location verification mechanism takes into account the following parameters:

1. Two sets of location coordinates from two different location sources;
2. IP address of the client (smart phone);
3. MAC (Media Access Control) address of a nearby access point with the strongest signal.

**Registration**

During the registration process the location information of the user is collected by the LBC running in the smart phone, sent to the server and stored in the database. This information includes the latitude (LAT), longitude (LNG) and accuracy (in a certain unit of measurement e.g. meters) as obtained from the location API of the smart phone. As shown in Figure 4, it represents the exact location and a range that the user wants to signify as an authorized location, from which access can be granted.

![Figure 25. Location Information](image)

In addition, the MAC address of the strongest nearby Wi-Fi access point is detected and stored in the database as an additional parameter. This value is obtained by comparing the RSSI (Received signal strength indicator) of all detected Wi-Fi signals and choosing the one with the highest signal strength.

All of these values are then stored in the database as part of the user registration information. Figure 25 below shows an example of location database scheme.
Authentication and authorization

During the authentication and authorization stage, the location of the user is detected and verified by the LBID server using the following mechanism:

**Step 1:**

In most of the major smart phone platforms there is usually more than one source (APIs) for obtaining location. For example, on Android there is the normal Android Location API, Skyhook and so on. This is the same for iOS, Blackberry and others. Based on this, our location verification utilizes two sets of location coordinates (LAT, LNG and accuracy) that are obtained from two different location APIs. In this way, the reliability and accuracy of the location is ensured since the result does not depend only on one source.

These two location coordinates are then compared to see if the location areas they represent overlap. In normal circumstances these two areas should always overlap, as shown in the left part of the following Figure 26. If there is no overlap at all between these two location areas, as shown in the right part of the following Figure 26, it is a good indication that one or both of the sources of location are not correct and may have been compromised through one of the methods described previously.

![Figure 26. Step 1 of Location Verification](image)

If there is no overlap location, the verification fails and the process stops. If the two results overlap, the process continues to step 2.

**Step 2:**

The public IP address of the client as observed by the server is recorded and used to estimate the location coordinates (LAT, LNG and accuracy) of the user using IP2Location service [16]. The result of this is then compared with both sets of coordinates from Step 1 to see if location areas are contained within the area represented by the coordinates from Step 2.

Both location areas, described in Step 1 should be contained within the location area that is calculated based on the IP address, as shown in Figure 27. The location area obtained from IP address is usually of lesser granularity which covers a bigger area and should contain both locations described in Step 1, which are more precise with smaller areas. If both locations coordinates are not contained within the larger range of area from the IP address then it is good indication that one or both of the location coordinates from Step 1 are inaccurate and may have been spoofed.
Therefore, the location verification process fails and stops if both locations from Step 1 are not contained in location from Step 2. If this check is successful the process continues to the final Step 3.

**Step 3:**

The MAC address of the access point, with the strongest detected Wi-Fi signal is captured and compared to the one saved during the registration process. If the two values do not match the verification process fails and stops. On the other hand if the values match, the verification process succeeds and becomes completed.

These three steps provide a series of checks to ensure that user’s location as detected and reported by the LBC is correct and has not been spoofed or tampered with. After all the steps have been completed successfully, there is a high confidence in the validity of the location reported by the user.

The most accurate location result from Step 1 – the one with the highest accuracy (small range) represents the location of the user and thus is used to make further location-based authorization decisions depending on the authorized location registered by the user.

### 6.3 Evaluations

In this section we provide an analysis of our proposed authentication and authorization solutions in order to evaluate and assess their security against different possible attacks.

#### 6.3.1 Certificate-based Authentication and Role-based Authorization

The results are shown in Table 3. Since a session ID is randomly generated and included in each session, the reply attack can be prevented. Each message is digitally signed by the initiator, so that the repudiation can also be prevented. All the messages are encrypted by recipient’s public key, so that only the correct entity can see the messages. The tampering of data is also prevented. In addition, each transaction is confirmed based on challenge-response mode and signed by the initiator. If an attack happens, the recipient of the messages can be aware of it, so that the transaction can be promptly cancelled. In addition, usage of certificates provides additional protection, such as integrity and authentication.
<table>
<thead>
<tr>
<th>Certificate-based Authentication</th>
<th>Replay Attack</th>
<th>Impersonation</th>
<th>Data Tampering</th>
<th>Repudiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role-based Access Control</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

### 6.3.2 Location-based Authentication and Authorization

We compare our proposed location-based authentication and authorization mechanisms with other popular authentication and authorization systems and show that it represents an improvement over the other protocols. In addition, we will also consider non-security factors, such as user-friendliness and cost of the comparison. Table 4 shows comparison of our solution against a set of other chosen solutions. We have chosen the following solutions for comparison with our own design: (1) username/password combination; (2) token-based system (using the mobile phone as the token), such as proposed by Hallsteinsen et al. (Hallsteinsen, Jørstad and Thanh 2007); (3) OTP schemes – which are becoming a popular alternative for multi-factor authentication, which are commonly used by some online services (National Institute of Standards and Technology (NIST) 2011), (Bejar 2011); and (4) biometric-based schemes, such as fingerprint and voice scanners, considered to provide a higher level of security.
<table>
<thead>
<tr>
<th></th>
<th>Username/Password</th>
<th>Token-based</th>
<th>OTP</th>
<th>Biometric-based</th>
<th>Location-based using smart phones</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Password Guessing</strong></td>
<td>Vulnerable to this attack if a poor password is chosen</td>
<td>Not vulnerable</td>
<td>Not vulnerable</td>
<td>Not vulnerable</td>
<td>Not vulnerable</td>
</tr>
<tr>
<td><strong>Eavesdropping</strong></td>
<td>Vulnerable unless strong encryption is used</td>
<td>Vulnerable unless strong encryption is used</td>
<td>Vulnerable unless strong encryption is used</td>
<td>Vulnerable unless strong encryption is used</td>
<td>Not vulnerable – strong encryption is used for communication through TLS</td>
</tr>
<tr>
<td><strong>Man-in-the-middle attacks (MITM)</strong></td>
<td>Can be performed unless end-to-end security is used</td>
<td>Can be performed unless end-to-end security is used</td>
<td>Can be performed unless end-to-end security is used</td>
<td>Can be performed unless end-to-end security is used</td>
<td>Prevented through the use of TLS</td>
</tr>
<tr>
<td><strong>Impersonation</strong></td>
<td>If passwords are simple or somehow become disclosed, impersonation is trivial and not easily detectable</td>
<td>Relatively difficult since the token has to be physically compromised and stolen. This is easier to detect.</td>
<td>Relatively difficult unless the source (seed) is compromised and the sequence of passwords can be predicted</td>
<td>Relatively easy – for instance fingerprints and voice can easily be captured through various techniques</td>
<td>Very difficult since password has to be compromised, smart phone has to be stolen, and attacker has to be at the authorized location</td>
</tr>
<tr>
<td><strong>Replay attacks</strong></td>
<td>If captured they can be replayed</td>
<td>Not possible since a token is used</td>
<td>Prevented since passwords are used only once</td>
<td>If biometric information is captured it can be replayed</td>
<td>Not possible since possession of a mobile phone is needed</td>
</tr>
<tr>
<td><strong>Theft</strong></td>
<td>Can be stolen without easy detection</td>
<td>If a token is stolen, security can be compromised</td>
<td>The source which generates OTPs can be stolen to compromise the system</td>
<td>Biometric features cannot be stolen, but their measurements can be copied</td>
<td>Even if a smart phone is stolen it can't be used unless in the authorized location</td>
</tr>
<tr>
<td><strong>User-friendliness</strong></td>
<td>Easy to use, but once the number of systems increase it becomes a problem</td>
<td>Not very convenient since usually it requires a user to carry an additional separate device</td>
<td>Not very user-friendly since it requires the user to generate a random each time she wants access</td>
<td>Easy to use. However some technology such as iris scanners can be a little intimidating to some people</td>
<td>Easy to use. It does not require any additional device besides a mobile phone. Location determination works transparently</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Inexpensive to implement</td>
<td>Requires additional device which might be expensive</td>
<td>Varies depending on the mechanism used. Initial setup and synchronization of passwords may introduce some significant management costs</td>
<td>Special scanners which are usually expensive are required</td>
<td>The only cost is for GPS-enabled smart phone</td>
</tr>
</tbody>
</table>
Chapter 7: Security of Mobile Messages

Previous Chapters describe the architecture, components and security services for mobile transactions. This chapter describes protection of messages exchanged between the components during mobile transactions. Protection is based on applying cryptographic mechanisms to individual messages.

7.1 Overview of Mobile Messages Security

As mentioned in Chapter 1, many attacks are possible in mobile environments and a lot of sensitive data is exchanged while performing mobile transactions. Therefore, protection of messages is crucial for mobile services. Our system provides message security services, namely message integrity and message confidentiality, by applying cryptographic mechanisms for protection of messages. In addition, messages are protected, so that each participant in a transaction has access only to those data which are necessary and enough for his/her to perform the transaction and nothing more. In this way, privacy of transaction entities is also protected.

Transmission of messages is performed in two steps: the first one is before messages arrive to the Communication SP and the other one is after messages arrive to the Communication SP. During the first step, messages are transmitted through wireless environment, where it is more important and difficult to protect those messages. During the second step, messages are transmitted through wired networks, where they are easier to protect.

7.2 Message Security Options

Our system provides three options for security of mobile messages, as shown in Figure 28.

- **Basic security**: this option uses standard GSM security features of messages during their transfer through radio network and strong security only from the Communication SP further to mobile service providers. End-to-end security is not provided. This is the case when people use standard SMS service without loading and using any third party software. This case is very common in rural areas in developing countries, since advanced mobile phones, which support Java platform, Bluetooth, large storage capabilities and so on, are not commonly used in those areas. In this case, data is encrypted using GSM A5/1 algorithm during transmission between a client and the Communication SP, and then Communication SP encrypts data using AES (symmetric crypto algorithm) and sends such data to backend application servers or other service providers’ servers.

- **Medium security**: with this option, end-to-end security is provided by sharing a secret key between mobile client and service providers. In this case, Communication SP is just a bridge between a client and service providers. However, users’ mobile phone must be able to perform cryptographic functions, since Secure Mobile Client must be loaded in order to perform transactions, instead of using basic SMS services by every mobile phone.
High security: with this option, end-to-end security is provided by using asymmetric crypto functions. In this case, user needs not only to load Secure Mobile Client, but also his/her certificate. In addition, user is able to prevent data from being revealed to the Communication SP. It is working like this for each transaction: user first encrypts sensitive data with randomly generated symmetric key, then envelops that key using public key of the service provider, attaches the encrypted result to the data known by the Communication SP (IP address of the targeted service provider, user’s mobile phone number and so on), sends the message to Communication SP.

7.3 Message Format

In order to achieve end-to-end security, a format for enveloping of messages is defined. Message format is shown in Figure 29.

There are three security options:

![Figure 28. Message Security Options](image-url)
### Message Confidentiality Protection

- **Encryption** – security option is set to ENCRYPTION in the security header. Message encryption key (MEK) is randomly generated and used to encrypt message content for message confidentiality. MEK is then enveloped with recipient’s public key and combined with the encrypted content. Finally, secure message is packaged by combining encrypted message and encrypted MEK. At the receiving end, recipient will inspect security header to check which “security option” was selected. If it is ENCRYPTION, recipient decrypts MEK with his/her private key and thereafter decrypts encrypted message using the MEK. The process is shown in Figure 27.

---

**Figure 27. Format of Secure Messages**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHL(2)</td>
<td>phone number length (2 bytes)</td>
</tr>
<tr>
<td>CML(2)</td>
<td>client message length (2 bytes)</td>
</tr>
</tbody>
</table>

**Figure 30. Message Confidentiality Protection**
- **Integrity** – security option in the security header is set to INTEGRITY. Hash of the message is calculated using SHA-256 to get Message Digest, which is signed with the sender’s private key. Message is packaged and ready to be sent using any transport protocol. At the receiving end, the recipient will inspect security header to check “security option”. Recipient independently calculates Message Digest using SHA-256 and compares it with the received “digital signature” after decrypting it with sender’s public key. The process is shown in Figure 31.

- **Encryption and Integrity** – security option in the security header is set to SIGNED_ENCRYPTED. To sign and envelop a message, the steps required for signing and encrypting are performed one after the other. In the first step, message is signed by calculating Message Digest and then signing it to get Digital Signature. In the second step, MEK is generated to encrypt message content along with digital signature. MEK is then encrypted with recipient’s public key and packaged. At the receiving end, recipient checks security header for “security option”. Recipient decrypts MEK using his private key, and using MEK decrypts the received message which contains digital signature of the sender and actual message content. Recipient verifies digital signature to confirm message integrity and secure authentication. The progress is shown in Figure 32.
The SMC can also encrypt transaction data using the public key of the SPServer. Message header is encrypted with the public key of the Communication SP. Upon receiving a message from a mobile client, Communication SP decrypts message header with its private key, so that it can deliver the message to the correct Mobile Application SP. Mobile Application SP decrypts the transaction information with its private key to process the transaction. When the Mobile Application SP delivers transaction response, it encrypts transaction information with the public key of the mobile client and message header with the public key of the Communication SP. In this way, end-to-end security is achieved when exchanging messages.

As mentioned in previous Chapters, the Communication SP is responsible to transform the messages received from mobile clients into XML formats. Therefore, messages exchanged between Communication SP and back-end servers are in the XML format. Encryption and signature of those XML messages are complaint with XML Encryption Syntax standard (W3C 2002) and XML Signature Syntax standard (W3C 2008).
PART III: CONCLUSIONS

Part III contains conclusions and summary of the thesis. Chapter 8 explains the approach and results for validation of this thesis. Chapter 9 summarizes the thesis and describes future research activities.
Chapter 8: Validation and Verification of Research Results

Previous chapters describe the background, addressed issues and proposed solutions of our research. This chapter describes validation of our research results, which shows the value of our research contributions and proves the validity of our proposed solutions.

Verification and validation of our research are performed in two steps. First, we designed and implemented a prototype system based on our proposed solutions. Second, we evaluated the prototype system using finite state model (FSM) and a regulatory framework.

8.1 Descriptions of the Prototype System

In this section, we give the descriptions of the implemented prototype system, including its components and functionalities.

The system is called Secure Mobile SOA (SMSOAP) System. It was designed and implemented based on the existing proven technologies and standards and it addresses all research problems described in Chapter 1 through the secure SOA architecture and various security mechanisms, described in the previous Chapters. As the proof of concept, the SMSOA system provides comprehensive security features, as well as high flexibility, adaptability and usability.

All servers of SMSOA system are running on standard OS platforms – various versions of Windows, using Java Run–Time Environment, SQL–compliant database servers, and TCP/IP protocols. Deployment of client mobile applications deserves additional attention and considerations. Namely, the development goal was that the mobile application should be portable to different types of mobile phones and in different environments (developed and developing countries).

Considering those limitations, there are several versions of mobile applications. The simplest version is based on SMS messages, without the need of preloading of any software. The next version is the software that provides only GUI for SMS messages. The third one is called USSD application, which provides USSD service for mobile customers. The most comprehensive version provides more security features. If the mobile application is loaded into the phone, then there are two sub-versions: one can be downloaded through mobile Internet and the other one is pre–loaded in the UICC chip. Thus, our mobile applications support a variety of mobile phones, with alternative capabilities and wireless communication protocols. More details about Secure Mobile Client (SMC) are given in this chapter.
8.1.1 System Components

The first component of the system is **Administrative Station** which is used by system administrators for all system administration and management functions. This component provides supporting functions, such as administrative functions for **Communication Server** and **Payment Server**, the functions for management of banks, customers, their bank and mobile accounts, and their transactions. For connection protocols, SMSOA system supports multiple SMS Gateways as message switches, but each must be connected to the different mobile number of the system. GPRS data connections and Bluetooth connections are also supported. Payment Server manages mobile accounts, but at the same time it represents mobile banking switch, i.e., it can be connected to multiple banks, bankcard processors, and/or other financial service providers (remittance, bill payments, micro-finance, and so on). For security services, the system has **IDMS Server**, where users/customers and all other System Servers are registered and **CA Mobile PKI Server**, supporting mobile certification protocol, issuing and distributing certificates to other components of the system.

SMSOA system includes several versions of the **Secure Mobile Client (SMC)**:

- A Java–based application that may be loaded in the memory of Java-enabled mobile phones;
- Similar applications for iPhone, Android, and Blackberry smart phones;
- SIM SMC, that can be loaded in the SIM chip (Zhao 2011);
- Java card SMC that can be loaded in UICC and smart card chips; and
- NFC SMC loaded in NFC MicroSD cards or NFC chips in mobile phones.

For agents, the system has **Secure Mobile Agent**, an application for different types of smart phones that supports agents’ functions. For merchants, the system provides **Secure Mobile Merchant**, also an application for various smart phones.

For users, in addition to various versions of the Secure Mobile Wallet, the system provides **Smart Card**, where smart card Wallet is loaded in a smart card chip. The card uses biometric (fingerprint and facial) authentication and can be used with contact and contactless payment protocols.

For merchants, the system provides **Mobile PoS Station** that can be used with standard bank cards (EMV transactions) and also with SMSOA Smart Cards, supporting contact and contactless protocols and with NFC Wallet loaded in mobile phones.

SMSOA system is based on secure mobile applications architecture (“mobile ecosystem”), which comprises the following twelve components:

1. **Mobile Communication Server** is connected with SMS Gateways or Internet of various mobile operators or telecom aggregators. It receives SMS messages or data messages through mobile Internet or Bluetooth connections, recognizes them as messages for particular service provider, and passes those messages to Payment Server or to the corresponding Mobile Applications servers.
2. **Mobile Payment Server** maintains registration and payment data and credentials for all customers. This Server handles database with mobile accounts, and in combination with mobile services database, processes financial mobile transactions. If the system is deployed to support mobile banking (access to bank accounts using mobile transactions), then Mobile Payment Server must be connected in the background either to the Bank IT Server, in order to process transactions based on mobile accounts and/or bank accounts, or to the Bank-Card Processing system, in order to process bank-card payments.

3. **Mobile Web Portal** is used for Web access to the system.

4. **Mobile Application Servers** are available in different forms, depending on mobile services being provided. For instance, for mobile parking transactions, this server is Mobile Parking Server. For mobile ticketing, this server is Mobile Ticketing Server. If payment is needed for those mobile transactions, each of those Mobile Application Servers is connected to the Mobile Payment Server. Mobile Application Servers accept transactions for mobile services of various providers and store them into a database as active transactions. These database entries are used in combination with the Mobile Payment Server to perform payment transactions.

5. **Web Payment Module** is a module that can be combined with standard Web e–commerce servers to support payments using SMSOA mobile system. This module is accessed through the “Pay” button available at the check–out page of the Web E–commerce servers. When selected, it asks users for their mobile account and payment amount, it connects to the Mobile Payment Server and performs payment.

6. **Administration Station** is used by system administrators to perform administrative functions with the Mobile Communication Server and Mobile Payment Server and to review various system reports.

7. **Secure Mobile Client** is an application supporting secure mobile financial transactions in SMSOA system. It is available in several forms:
   - as smart phone application, in which case it must be loaded into mobile phones,
   - as standard smart card applet, in which case it must be loaded into the chip of the standard smart card,
   - as UICC smart card applet, in which case it is loaded in the SIM chip of the mobile phone, where it supports financial and security functions of the SMC mobile phone application,
   - as Wiblet application, in which case it is loaded in the SIM chip of the mobile phone and can be used without mobile phone application,
— as NFC application, in which case it must be loaded in the NFC–
enabled MicroSD card.

8. **Secure Mobile Agent** is a smart phone application supporting mobile
financial transactions for agents.

9. **Secure Mobile Merchant** is a smart phone application supporting
mobile financial transactions for merchants.

10. **Smart Cards Station** is used by system administrators to enroll users
for issuance of smart cards, capturing their photo and two fingerprints for
biometric authentication.

11. **Smart Cards Server** is used by system administrators to load and
personalize Wallet as applet into smart cards.

12. **Mobile PoS Application** is an application loaded in the Point–of–Sale
devices that are used for over–the–counter payments using SMSOA system.
The devices and Mobile PoS Application accept magnetic stripe (EMV) cards,
contact and / or contactless chip smart cards.

### 8.1.2 System Functions and Services

The purpose of SMSOA system is to provide various types of secure mobile services. Currently, the system is ready to support use of mobile accounts, mobile banking, and international bank–to–bank transactions (remittance). For these three groups of services, each deployment needs minor customization of the system needed in order to link it to the back–end data processing server in each participating bank.

For other groups of services (payment and administration of air–time, paying bills, payment of tickets, payment for parking and so on) the system must be linked to the servers of providers of those services. In this case, SMSOA system supports two groups of mobile services: front–end services of the specific system and also payment system for those services. For instance, for air–time, the system provides functions to inquire the plans, rates, the status of accounts and other details and when buying air–time, the system supports payments. This customization is specific for each deployment and for each individual service provider.

### 8.1.2.1 Mobile Accounts

One of the main functions of SMSOA system is to manage and use mobile accounts. These accounts may be used to deposit and withdraw cash and for various mobile payments. The system is combined with Secure Mobile Servers for various other mobile services described below, so that subscribers with mobile accounts are able to use those accounts to pay for those services. In addition, the system provides cash–in/cash–out options at local stores, where shop–owners act as agents of the system. One particular type of purchase transactions may be selling air–time using mobile
phones and it may be extended to payments over the counter to pay goods and services offered by local stores.

8.1.2.2 Secure Mobile Banking
For customers with bank accounts, SMSOA system provides possibilities to perform secure mobile banking. With this system customers can access their bank accounts using mobile phones, transfer money from those accounts to their mobile accounts and back, and perform other m–banking transactions (paying bills, paying subscriptions and so on). This mobile financial transactions system also registers all agents of the system, event promoters and other partners of various service providers, so that it also supports back-end financial transactions.

8.1.2.3 Secure Payments Over–The–Counter
One of the services that SMSOA system can support is various forms of payments over–the–counter (OTC). Those could be payments to merchants in shops, at supermarket check–out counter and tickets for public transportation.

The system is designed in such a way that it can support different combinations of those mobile services at the same time. For example:

- When used for mobile banking, only bank accounts of bank’s customers must be registered and transactions can be performed between accounts (transfers), cash–in/cash–out using those accounts, or as national/international bank–to–bank account transfers.

- When used with mobile accounts only, the system supports multiple mobile accounts all linked to an escrowed (collective) bank account, belonging to the service provider. Transactions between mobile accounts are local to the system and not reflected to the escrow account, while cash–in/cash–out or various payments are reflected to the escrow account.

- Mobile accounts and bank accounts may be mutually linked, so that amounts deposited to mobile accounts may be transferred from bank to mobile accounts and back.

- If the system is used for m–payment services by other m–services systems (bill payments, ticketing, bus fares, parking, purchase of airline tickets and so on) then both mobile or bank accounts may be used for payments of m–services offered by those other systems.

8.1.2.4 Additional Financial Services
Customers with mobile or with bank accounts using m–payment services of SMSOA system can also pay other expenses, like paying bills, invoices and Web orders.

8.1.2.5 International Financial Transfers (Remittance)
Customers with mobile or with bank accounts accessible using m–payment services of SMSOA system, are able to make international transfers, either to their bank
accounts in their home countries or to mobile accounts in SMSOA system installed and activated in their home countries.

### 8.1.2.6 Secure Payment and Administration of Air–Time Accounts

Customers with mobile or with bank accounts accessible using m–payment services of SMSOA system, can also use funds to pre–pay air–time with their telecom operators.

### 8.1.2.7 Secure Mobile Ticketing Services

Using SMSOA system, Web ticketing service may be extended to provide secure mobile transactions – purchasing tickets using mobile phones. Tickets may also be distributed to mobile phones, as venue booths are upgraded to perform their verification. This system provides inquiries about all shows for which tickets are available, inquiries about pricing of individual tickets, ordering and payments of tickets using mobile phones.

### 8.1.2.8 Secure Mobile Parking Services

The SMSOA system can be used to pay parking. In this case, all parking places, parking meters, and hourly rates are registered in the system. Parking may be paid by indicating parking location and parking time. The system performs automatic payment using mobile or bank accounts and warns user promptly of the paid parking time is about to expire.

### 8.1.3 System Activation and Administration

When the Administration Station starts, the system will display login panel:

![Figure 33. Administration Station Login Panel](image)

If smart card is used for authentication, **PIN** field will be enabled. Otherwise, select “**Password**” for login:
After that, the system will go through the initial sequence. That sequence comprises three steps. In the first step, Mobile IDMS Server is configured, in the next step its registration data (preconfigured) are entered in the database and in the third step its security administrator is registered.

Configuring of the system: select DB Server Type. User name and password should be as selected when installing MySQL Server.

Registration of the Security Domain: registration data for security domain is predefined.
After registration of the security domain for the system, security administrator must be registered. This is performed in the third step of the initial sequence, when the following form is displayed.
8.1.3.1 Administration
After activation, the system will display initial panel (Figure 38). Initial panel is used to access various administrative functions of the system.
Drop-down menu item “Local Security Management” is used to manage security parameters of the Administrative Station: configuration parameters, certificates, log entries and so on. “Identity Server Management” is administrative interface to the Mobile IDMS Server, it is used to register users and various servers in SMSOA system. “Mobile Financial Server” is used to administer Mobile Financial Server, and “Mobile Communication Server” is used to administer Mobile Communication Server.

8.3.2.1 Registration of additional security administrators
Security administrator registered during the initial sequence may register additional security administrators. For that, switch to “Identity System Management” subsystem. The following panel will be displayed:

![Figure 39. The Main Panel of the Identity Management System](image)

When “Persons” drop-down menu item is selected it shows the list of functions: “Register Person” or “List Persons”.

![Figure 40. “Persons” Drop-down Menu](image)

When “Register Person” is selected it displays the same form as in Figure 40. In this case, “Security System Role” will be enabled and the role for the new user who should also be system administrator should be “Security Administrator”.

8.1.3.2 Registration of mobile servers
Each server in SMSOA System must be registered. Currently, Mobile Communication Server and Mobile Financial Server must be registered. For that,
select “Identity Server Management” drop-down menu item from the “SAFE System” drop-down. The system will display the panel for management of identities, shown in Figure 41. Select “Servers” drop-down menu, and then select “Register Server” and the following panel will be displayed:

![Figure 41. Registration of Mobile Payment Server](image)

Registration of the Mobile Communication Server and Mobile Financial Server is mandatory. Select appropriate Server Type, select Server Name (arbitrary name), Server IP number, select Server’s administrator out of registered administrators and his/her password to login into the Server. This password may be the same as administrator’s login password or may be different. See Figure 34.

8.1.3.3 Registration of message channels
In order to process mobile transactions, the message rules must be defined. This must be done at the Mobile Communication Server through Administration Station. As shown in Figure 42, the Mobile Communication Server is started by “right-click” the server name.

![Figure 42. “Login to server” Pop-up Option](image)
After Login, the functions of Mobile Communication Server are activated so that the administrator can register message channels and message rules, as shown in Figure 43, and list the logs of all messages, as shown in Figure 44.

![Figure 43. Registering Message Channels and Message Rules](image)

The first step is to register message channels, which is to define routing of messages. Registration of message channel requires several data, which is shown in Figure 45.

![Figure 44. List Messages](image)

![Figure 45. Registering of Message Channel](image)
8.1.3.4 Registration of message rules
After registration of message channel, registration of message rules is required, as shown in Figure 46.

![Figure 46. Registration of Message Rules](image)

8.1.3.5 Registration of system setting
Select “Settings” drop-down menu on the Mobile Financial Panel. The following form will be displayed:

![Figure 47. Registration of System Setting](image)

Select the country of deployment. “SAFE System ID” is the sequence number of the instance of the system in the selected country. The purpose of system ID and its usage are explained in Chapter 4.
8.1.4 Secure Mobile Client

As one of the components of SMSOA system, secure mobile client represents secure, stable, convenient and easy-operational application, which is pre-loaded into customer’s mobile device. This section describes design, implementation and demonstration of Secure Mobile Client.

8.1.4.1 Versions of the Secure Mobile Client

SMSOA system provides four versions of secure mobile wallets: SMS Client, Thin Client, USSD Client, and Thick Client. The section gives overviews of these four versions of the client.

**SMS Client**

SMS client is based on usage of the standard SMS messages. Since all mobile phones support this function, it will be the most widely spread version of secure mobile client. It has the least requirements for customer’s mobile device. The disadvantage of this version is that it could not use other communication protocols, such as Bluetooth or GPRS. Therefore, there is no session created during transaction process, which directly influences the speed of the transaction process. Figure 48 shows an example of the SMS Client.

![Image of SMS Client](image)

**Figure 48. Example of the SMS Client**

**Thin Client**

The concept of a Thin Client comes from the “Command Prompt” function in most operating systems. It supports text-based (command-line) function. It is easier and faster to perform transactions. Besides, it could use Bluetooth and GPRS to build a
session with Mobile Communication Server. In Thin Client, there is a text box, whose size is as large as the mobile phone’s screen. From the text box, customers can enter transaction command as transaction request. After the user clicks “Enter”, transaction request is sent out to the server through available communication protocol and Thin Client receives the response from the server. The response is shown in the same text box, but one line below the first line. Then user can keep entering the text command on the first line and will get the response at the same position as the previous response. Previous response will be deleted from the text box. The example of Thin Client is shown in Figure 49.

![Figure 49. Example of the Mobile Thin Client](image)

**USSD Client**

USSD is a capability of all GSM phones. It is generally associated with real-time or instant messaging type phone services. There is no store-and-forward capability, typical of other short-message protocols (in other words, an SMSC is not present in the processing path). Response times for interactive USSD-based services are generally quicker than those used for SMS. USSD Client is based on the combination of USSD and Thin Client. USSD Client has also one big text box for user entering command, except that USSD code instead of text command is entered. The menu is sent back from the Server and shown in the text box. Customer could choose from the menu by entering the number in front of the item. The usage example of USSD Client is shown in Figure 50.
Thick Client
Thick Client provides the most comprehensive set of functions and supports for customers. It gives full GUI that contains drop-down selections and data entry forms. Therefore, the requirements of running this version are relatively higher than the other three versions. All kinds of communication protocols, such as SMS, Bluetooth, 3G and GPRS, are supported by the Thick Client. Customers may choose the most proper one for transactions based on different environment situation and contexts. For example, when user is at home and wants to buy a ticket, SMS or GPRS are more suitable since the distance from a client to the server is too far to use Bluetooth connection. However, if the user goes into a shop to buy over the counter, Bluetooth is the most convenient and fastest protocol to use. The example of the main interface is shown in Figure 51.
**Figure 51. Example of the Mobile Thick Client**

**SIM/UICC Client**
All the versions listed above are based on the use of memory card of a mobile device. Another version that is loaded inside the SIM/UICC chip of the mobile phone, was developed in order for the users with simple mobile devices to conduct mobile transactions. The SIM/UICC chip is the smart card used in mobile phone in GSM or UMTS networks. Since it is a smart card, it inherits all the security features of smart cards. It provides a secure storage of data. The functionalities and applications are the same as Thick Client except that user interface of the SIM/UICC Client is simpler than that of Thick Client due to the limitations of storage and process ability of SIM/UICC chips.

For the implementation of SIM/UICC Client, a middleware, located between upper layer application and lower layer smart card applet, was developed. The middleware receives requests from upper layer, transfers them into APDUs, and sends the APDUs to the applets running on the SIM/UICC chip. Then, the middleware transfers the responses from SIM/UICC chip to the desired format and sends the response to the upper layer application. The example of the SIM/UICC Client is shown in Figure 52.
8.1.4.2 Design and implementation

The methodology for the design and implementation of the secure mobile client is called *Generic, Secure and Modular (GSM)*. The distinguished features of *GSM* methodology are the following:

- It treats mobile applications in a holistic way, meaning that it can be applied to development of applications for any platform and using any programming language;
- It is a modular approach and structures mobile applications into several groups of modules. Each module may be independently designed and implemented, what makes an application flexible, expandable and reusable;
- It provides several ready-made core security modules, which can be used by developers to integrate security in their applications already during the design and development process.

*GSM* methodology structures mobile applications into four groups of modules: User Interface (UI) modules, communication modules, security modules, and business logic modules. The reason for this approach is that these four groups of modules are necessary for most of the secure mobile applications. In other words, in order to create a secure mobile application, at least these four groups of modules should be included.

Every mobile application must have user interfaces for displaying information to users and interacting with them. Some of them are simple and some are complex, which depends on the hardware and operating system of mobile phones. All mobile applications need also to communicate either with open networks or with internal modules of mobile devices. Communication modules exchange messages with networks, with other devices, and/or with other internal modules of mobile devices.
Next, security of mobile applications is also very important. Therefore, security modules that provide security features, such as generation of RSA keys, encryption and digital signatures, are also necessary. These three groups of modules are not sufficient to create a complete secure mobile application, since they are separated and independent of each other and they do not perform any function specific to an application. Hence, business logic modules are needed to link all other modules together and coordinate them based on the services or functionalities provided by a specific mobile application.

It may be noticed that security and communication modules are generic, meaning that they can be used by any application so, only their invocation is needed. UI modules are usually equivalent, but not identical for various applications, meaning that some support selection of functions (menu screens) and others support data entry and result displaying functions (data forms screens). Therefore, UI modules should be available as templates. Business logic modules are specific and, therefore, must be created for each application. Generic structure of a secure mobile application is shown in Figure 53. During research and development, whose results are described in this thesis, several mobile applications have been created based on the G3M methodology. As a result, generic security and communication modules and UI templates were developed, which are available for design and implementation of other mobile applications.

Its internal structure comprises four groups of modules:

- **GUI module** – responsible for creating all user interfaces, related objects, such as forms, text editors, choice boxes and so on.
- **Communication module** – responsible for creating communication interfaces for all supported protocols between client and server,
- **Business logic module** – responsible for creating transaction request messages and processing responses messages,
- **Security module** – a “black box” that takes responsibility to utilize and combine security mechanisms providing security features, such as authentication, confidentiality, integrity for transactions and operations.

The internal structure and components of the Secure Mobile Client are shown in Figure 53.
By following GSM method, the secure mobile client was implemented as various versions for different mobile platforms, such as Android, iOS and Windows Mobile. In order to securely store data in the SIM/UICC chip, a middleware was developed. The UICC middleware is a transparent, lightweight, secure and autonomous software module used by the SMC. It was designed and implemented in such a way that the process of communicating with the UICC is transparent to the SMC, which does not distinguish any difference between fetching and storing data from/to the smart card and manipulating data as files on its own system. Figure 54 shows an example of storing data on the UICC.
The SMC provides several security services for its financial transactions, including user authentication, message confidentiality and integrity, non-repudiation, authorization, etc., as mentioned in previous Chapters. For those services, MobileSecurityProvider module was designed and implemented. It provides various security functions, such as generateRSAKeys, AESEncrypt/AESDecrypt and generateMessageDigest. All security services, provided by the SMC, are based on these functions. Therefore, whenever an application requires new security features, a developer just needs to extend the MobileSecurityProvider module for new functions and cryptographic algorithms. Implementation of the MobileSecurityProvider object can be different, depending on the platform, programming language, APIs and so on.

The UICC middleware was also designed to function in a secure way. The middleware has no semantic knowledge of any data that it parses and passes in both directions. This gives mobile applications an option to store encrypted data in the UICC. Also, the middleware has no knowledge of passwords or keys used for authentication or cryptography. Another security feature is that the middleware keeps no data in its internal memory. All changes and modifications of data are saved directly in the UICC, which provides strong protection of data and also prevents synchronization issues (mostly when dealing with online, real-time transactions).

Finally, the middleware follows the authentication mechanism required by the SIM/UICC chip, that is, a strong authentication protocol and secure channel between an application and UICC(CardContact n.d.) As a result, mobile users must authenticate themselves to the SMC applet before being able to read and write data. If a user fails to provide the correct password after a predefined number of attempts, the UICC locks and can only be unlocked by the administrator.

8.2 Functional Verification with Finite State Machine

There are many ways of describing complex process logic of information systems and FSM is a tool that can be used to model process logic. It has been widely used for many years in lots of areas. FSM defines a set of states and regards a process as the transitions of these states.

AFSM was created according to the prototype SMSOA system. First we defined seven states as the correct states from the system’s point of view: System Installed, System Configured, System Ready, User Registered, Agent Registered, Transaction Pending, Transaction Processed and Transaction Error. There are three roles that can affect the states of SMSOA system: System Administrators, Customers and Agents.

- **System Installed** state represents the fact that system administrator successfully installed the system;
- **System Configured** state represents the fact that system administrator completed configuration of the system;
- **System Ready** state represents the fact that system administrator successfully tested and adjusted the system so that it is ready for providing services;
- **User Registered** state represents the fact that at least one customer successfully registered in the system;
- **Agent Registered** state represents the fact that at least one agent is successfully registered in the system;
- **Transaction Processed** state represents the fact that the transaction has been processed by the system;
- **Transaction Pending** state represents the fact that the transaction, which needs confirmation, is pending and waiting for confirmation;
- **Transaction Error** state represents the fact that the transaction can not be cleared by the system due to some reason, such as wrong transaction information, transaction is against system rules.

Second, we defined the actions that represent state transitions. These actions are various mobile transactions supported by the SMSOA system and can be modified or extended according to the system requirements. Based on the functions and transactions supported by the current implemented system, the following actions were defined:

1. System Administrator configured the system;
2. System Administrator tested the configuration of the system
3. Customer successfully registered in the system
4. Any transaction is triggered
5. Transaction that does not need confirmation is triggered
6. Agent successfully registered in the system
7. Transaction succeeds
8. Transaction error
9. Transaction that needs confirmation is cleared
10. Transaction confirmed

In addition, the following rules, which affect the conditions of state transitions, were defined:

- Only system administrators are allowed to install and configure the system
- Only registered customers with mobile accounts are allowed to perform mobile transactions
- Only registered agents with mobile accounts are allowed to perform mobile transactions that require agents
- Only registered agents with mobile accounts are allowed to register customers or merchants
- The transaction can not be cleared if the balance of the targeted account is not sufficient.
As a result, a finite state model was created, as shown in Figure 55. According to this model, it is proved that the system is always in the correct state, meaning that the system is functioning correctly and does not allow any operations against the rules. For example, if a user wants to withdraw 100 units from the account that only has 50 units, this request will cause transaction error and the system automatically goes to original state.

### 8.3 Regulatory Framework for Mobile Payment System

Another way of validating our system is through a regulatory framework for mobile payments in Nigeria. This regulatory framework is developed to conform to international best practice and standards. We verified the SMSOA system by comparing it with regulatory framework to see whether it can fulfill the requirements listed in this regulatory framework, especially the technical part and rules that system shall follow. The reason for choosing this regulatory framework is that this framework is comprehensive and specially suitable for mobile transaction systems. The results are described in this Section.
Based on the definitions given in this regulatory framework, the SMSOA system belongs to non-bank led mobile payment model and provides mainly stored value (e-Money) account based transactions. Once connected with financial institutes, such as banks and credit union, the SMSOA system also provides card account based and bank account based transactions. There are certain rules that the system should comply with, listed as follows:

1. All system-based accounts shall have an identification system that generates unique identifier per user account within the mobile payment system. (Compliant)
2. All system-based accounts shall only be accessible through the mobile payment system. (Compliant)
3. The user may however specially request other means of access to his/her system-based account other than specified in 2 above. The liability of the user shall be clearly stated before granting request. (Not compliant)
4. All accounts and transaction details shall be stored in an encrypted format within the mobile payment system. (Compliant)

In the technical part, the regulatory framework outlines several technology standards that the technology implemented for mobile services shall comply with. These standards are classified into nine categories: modularity of technologies, solution initialisation, compatibility, interoperability, message format, reliability, flexibility, user interface and security. And we listed the following standards that are directly relevant with the system:

Modularity of Technologies:

1. The technology deployed in the delivery of mobile payment services shall comprise a set of interoperable infrastructure modules that work seamlessly. (Compliant)
2. Provided the security requirements of this regulatory framework are met, the mobile payment service shall use any mode of communication including, but not restricted to, the following: (Compliant)
   - Secure SMS
   - WAP/GPRS and
   - USSD 1
   - EDGE
3. Provided the security requirements of this framework are met, the mobile payment service shall use any mode of user interface, including, but not restricted to, the following: (Compliant)
   - Secure SMS
   - Menu driven USSD 1 application
   - WAP/GPRS
4. The mobile services shall not use plain SMS. (Compliant)
5. Only secure channels shall be used in providing mobile payment services. (Partial compliant)
6. The mobile payment service shall ensure non-repudiation. (Compliant)
Solution Initialisation
1. The mobile payment solution shall ensure simple initialization of the payment application. (Compliant)

Compatibility
1. The mobile payment solution shall be compatible and interoperable with the network infrastructure of different telecommunication companies, solution providers, and scheme providers and the Nigeria Central Switch. (Compliant)

Interoperability
All schemes shall be able to interoperate:
1. with other scheme or solution providers (Compliant)
2. with other payment channels like cards, ATM, POS, etc. (Compliant)

Message Format
Mobile payment solutions deployed shall adhere to the following message format:
1. encrypted end-to-end (Compliant)
2. ISO 8583 compliant (Compliant)

Reliability
1. Payment instruction shall be consistently executed. In the event of failure, immediate reversal shall be automatic. (Compliant)
2. Users shall get immediate value for every successful transaction. (Compliant)

Flexibility
1. Users shall be able to switch between service providers without any bottlenecks. Switching from one solution to another shall be as easy as possible. (Compliant)

User Interface
1. The user interface shall, at the minimum, be menu-driven. (Compliant)
2. If private or personal data in the application are directly accessible through this menu (for example, memorizing the PAN-Primary Account Number), the access to this menu shall be protected. (Compliant)
3. Administrative functions – for example, tracing, certification/confirmation of transaction shall be provided. (Compliant)
4. PIN shall be encrypted at the point of entry. (Compliant)

Security
The overall security framework shall ensure:
1. encrypted messaging / session between consumer’s phone and third party service provider / Telecom Company. The minimum encryption standard to be specified in Triple Data Encryption Standard (3-DES) encryption; (Compliant)
2. all subsequent routing of messages to be scheme providers’ servers must be with the highest level of security with dedicated connectivity like leased lines (E1 links) / VPNs; (Compliant)
3. that any sensitive information stored in third party systems is restricted with appropriate encryption and hardware security standards; (Compliant)
4. all transactions on an account shall be allowed only after authentication of the mobile number and the PIN associated with it; (Compliant)
5. that mobile payments application shall not allow the option of saving the PIN either on the handset or on the application; (Compliant)
6. all accounts activated by the consumer on the mobile application is linked to the mobile phone number. This mobile number shall be used as the second factor authentication for mobile transactions; (Compliant)
7. the PIN does not travel in plain text during the transaction; (Compliant)
8. that proper system of verification of the phone number shall be implemented; (Not compliant)
9. the payment authorization message from the user’s mobile phone shall, at the minimum, be triple DES encrypted and checked for tampering by the service or scheme provider. It shall not be possible for any interceptor to change the contents of the message; (Compliant)
10. logical access controls to data, systems, application software, utilities, telecommunication lines, libraries, system software, etc. exists; (Compliant)
11. proper infrastructure and schedules for backing up data. The backed-up data shall be periodically tested to ensure recovery without loss of transactions in a time frame as given out in the security policy. (Not compliant)

As described above, we can say that the SMSOA system comply with almost all the rules, defined in the regulatory framework, which a mobile payment system shall follow. Therefore, from the system’s point of view, the SMSOA system is compliant with this regulatory framework, which can practically validate our proposed solution.
Chapter 9: Conclusions

This Chapter summarizes the achieved results and suggests future research activities in this area.

9.1 Conclusions

This thesis defined and developed a concept of secure service-oriented architecture for various mobile transactions and applications. The architecture is comprehensive, scalable, modular, expandable and open. Based on this infrastructure, as the proof of concept, a system that supports a secure, convenient and reliable large-scale infrastructure for mobile financial transactions was developed and tested. The system supports secure transactions with multiple banks, direct client–to–merchant payments, person–to–person transfers, and other, non–banking mobile applications. The distinguished feature of the system is its strong security for users, their transactions, messages and applications.

Infrastructure components of the system, using secure messages exchanged between them, provide a number of secure financial services. These services may be used for various types of mobile transactions: m–Banking, m–Commerce, m–Ticketing, m–Parking, m–Loans, and so on. All those mobile services are supported by corresponding Mobile Provider Servers, which are combined and integrated into the secure mobile service-oriented architecture.

In this research, the following research problems have been solved and we claim the following scientific contributions in the area of secure mobile transactions:

9.1.1 Security Architecture for Mobile Environments

The first contribution of this dissertation is the proposed secure, flexible, scalable and expandable architecture. This architecture addresses the most important research issue of this dissertation, “lack of an open, comprehensive, adaptable and secure infrastructure”, described in Section 1.2.1. It also achieves the research objective (a), described in Section 1.2.3.

The secure architecture was published in the paper entitled “Secure Service-Oriented Architecture for Mobile Transactions”, authors Feng Zhang, Sead Muftic and Gernot Schmöelzer, (published in the Proceedings of IEEE World Congress on Internet Security (WorldCIS-2011), 21-23 February, 2011, in London, UK). We described secure service-oriented architecture for mobile transactions. The architecture comprises components, protocols, applications and interfaces and it provides various security services to various mobile applications: registration, certification, authentication, and authorization of users, secure messaging at an application–level (end-to-end security), protection of data in databases, and security services for protection of its own components. The architecture is modular, integrated, extendible and scalable. This paper describes the design of the architecture, the status of its current implementation, and future research and development plans. The ideas and results of this paper have been included in Chapter 3, Chapter 4 and Chapter 6 of this dissertation.
A prototype system was designed and developed based on the proposed secure architecture. The system was described in the paper entitled “SAFE System Secure Applications for Financial Environments Using Mobile Phones”, authors Feng Zhang and Sead Muftic, (published in the Proceedings of the IADIS International Conference e–Society, Barcelona, Spain, 2009). We have introduced the concept of SAFE system (Secure Applications for Financial Environment) that represents a secure, convenient and reliable infrastructure for mobile financial transactions. The infrastructure comprises Mobile Wallet, three servers: Gateway, IDMS and Bank servers, security protocols, and messages between all components. The ideas and solutions of this paper have been included in Chapter 8 and Appendix B of this dissertation.

The comprehensive descriptions of SAFE system were published in my Licentiate thesis entitled “Secure Applications for Financial Environments (SAFE) System”, (published by Royal Institute of Technology (KTH), Stockholm, Sweden, June 2010). I analyzed the security and privacy issues existed in mobile financial environments and gave the details of the concept design and current implementations of the SAFE system, which represents a secure, convenient and reliable large-scale infrastructure for mobile financial transactions. The ideas and results of this paper have been included in Chapter 1, Chapter 3, Chapter 7, Chapter 8 and Appendix B of this dissertation.

9.1.2 Secure Mobile Financial Services for Developing Countries

The second contribution of this dissertation is the system for secure mobile financial services in developing countries, where IT infrastructure is poor, but everyone has a mobile phone. Since most mobile devices used in rural areas are very simple, we designed and developed secure protocols that any mobile device is able to use. This addresses another research issue “resource limitations of mobile devices and networks”, described in Section 1.2.1 and achieves the research objective (c) mentioned in Section 1.2.3.

One of the most important mobile financial services is the concept of mobile-ATM, which was published in the paper entitled “Experiences on Mobile-ATM Development in Developing Countries”, authors Amila Karunanayake, Kasun De Zoysa, Feng Zhang and Sead Muftic, (published in the Proceedings of The 1st International Conference on M4D Mobile Communication Technology for Development (M4D 2008, General Tracks), 11–12 December, 2008, Karlstad University, Sweden). We described the social, economic and technical impact of the Mobile-ATM system, which was developed by the authors. The proposed Mobile-ATM system is especially useful in rural areas, where accessing financial and banking services is a critical issue due to the distance barriers. Moreover, this paper points out the essential value-added services provided by our system with respect to financial transactions services such as security and confidentiality. Although the Mobile-ATM is technically feasible and practically deployed, it is important to have community acceptance. This paper discusses the community acceptance of this system and related issues. The idea and solutions for this paper have been included in Appendix B of this dissertation.
In the paper entitled “Mobile-ATM for Developing Countries”, authors Amila Karunanayake, Kasun De Zoysa, Sead Muftic and Feng Zhang, (published in the Proceedings of 7th ACM/Sigmobile Annual International Conference on Mobile Systems, Applications and Services (MobiSys2009), June 22-25, 2009 in Kraków, Poland), we have explained and made a demonstration of the Mobile-ATM system, which was developed by the authors. The idea and solutions for this paper have been included in Appendix B of this dissertation.

9.1.3 Security Services for Mobile Transactions

The third contribution of this dissertation is the concept of security services for mobile transactions. These security services are light-weight, therefore suitable for mobile devices, networks and applications. This addressed the research issue “most security mechanisms used in wired networks are not suitable for mobile environments”, described in Section 1.2.1 and achieved the research objective (b), described in Section 1.2.3.

One of these security services is location-based authentication and authorization, published in the paper entitled “Location-based Authentication and Authorization Using Smart Phones”, authors Feng Zhang, Aron Kondoro and Sead Muftic, (published in Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 25-27 June, 2012, Liverpool, UK). We have proposed a location-based authentication and authorization scheme for mobile transactions using smart phones, based on a new factor: where you are (location). The paper describes the distinguished features and architecture of the proposed solution and the core of our design, including three parts: location registration, authentication and authorization as well as location verification. The ideas and results of this paper have been included in Chapter 6 of this dissertation.

Another security service is mobile-PKI, described in the paper entitled “Public Key Infrastructure Based on Mobile Trusted Authorities”, authors Feng Zhang and Sead Muftic, (under the review for the 9th International Conference on Trust, Privacy and Security in Digital Business (TrustBus’12), 3-7 September, 2012, Vienna, Austria). We have analyzed the issues and limitations of adopting PKI in mobile commerce environments and proposed a new mobile PKI (m-PKI) solution based on mobile trusted authorities. The proposed solution provides PKI service anytime, anywhere for mobile users, based on mobile trusted authorities and light-weight certificate management protocols. It improves the security, flexibility and adaptability of PKI systems in mobile commerce environments. In addition, it provides face-to-face identity verification during certificate enrolment process. The ideas and results of this paper have been included in Chapter 5.

In the paper entitled “Is Your Email Box Safe?”, authors Feng Zhang and Rasika Dayarathna, (published in Journal of Information Privacy & Security, 2010, 6,1, ABI/INFORM Global pg. 28), we have examined security and privacy protection mechanisms of four leading email service providers: Gmail, Yahoo Mail, Hotmail, and AOL Mail. A number of observations and explanations were conducted in order to understand existing security and privacy protection mechanisms of these providers. After that, we proposed some recommended protection mechanisms, which can be implemented by service providers, system developers, and email users. The ideas and results of this paper have been indirectly reflected in Chapter 4. These
ideas and results are valuable for the designs of security services for mobile transactions.

9.1.4 Secure Mobile Applications

The fourth contribution of this dissertation is the design and development of secure mobile applications. The security of mobile applications becomes more and more important. Therefore, a methodology for design and implementation of secure mobile applications was proposed. Based on this methodology, several versions of secure mobile clients were developed, described in Section 8.1.4. This contribution addresses the research issue “large variety of mobile devices and platforms”, described in Section 1.2.1 and achieves the research objective (c).

The methodology was described in the paper entitled “Generic, Secure and Modular (GSM) Methodology for Design and Implementation of Secure Mobile Applications”, authors Feng Zhang, Ioannis Kounelis and Sead Muftic, (under the review for the Sixth International Conference on Emerging Security Information, Systems and Technologies, (SECUREWARE 2012, 19-24 August, 2012, Rome, Italy). We have proposed a Generic, Secure and Modular (GSM) methodology, which provides a generic approach for design and development of secure mobile applications. It is applicable to multiple mobile phone platforms and mobile operating environments. This approach treats mobile application in a holistic way and structures it into four groups of modules: User Interface (UI) modules, communication modules, security modules, and business logic modules. These four groups of modules can be designed and implemented independently and finally integrate them together. It not only simplifies the process of design and development of mobile applications, but also improves the reusability and robustness of mobile applications. In addition, we proposed a trusted layer model for designing the security modules of mobile applications, which provides generic application interfaces and comprehensive data protections. This trusted layer model is based on the model described in Chapter 3. The ideas and results of this paper have been included in Chapter 3 and Chapter 8 of this dissertation.

As an example, the concept of mobile wallet was published in the paper entitled “The Secure Mobile Wallet”, authors Hao Zhao, Sead Muftic and Feng Zhang, (published in Cutter IT Journal, 2010, Vol. 23, No. 7, pg. 32). We have analyzed the trends in mobile technology and applications and introduced the concept of secure mobile wallet, which is the subscriber’s component of the large and open service-oriented architecture. The ideas and results of this paper have been included in Chapter 8 of this dissertation.

9.1.5 Conclusions for Research Contributions

As conclusions, it is expected that the secure mobile SOA, proposed in this dissertation, is an important contribution to the concept of a secure, trusted, flexible and scalable framework for large-scale mobile transactions and also significant improvement of the current state-of-the-art of mobile transaction security.

9.2 Future Work

This dissertation has developed and shown a secure architecture and its properties, but several interesting issues remain to be further addressed and solved. This section
lists a number of questions that have been either left unanswered or only initially solved by this thesis.

9.2.1 Evaluation of Secure Mobile Systems and Applications

This thesis has designed and developed four types of mobile security services. The evaluations of these services are important for adjustments and improvements. These services have been only evaluated based on their security features. Therefore, one of the future activities is to conduct comprehensive experiments to evaluate the proposed security services, which include not only security, but also time, computational and bandwidth costs.

9.2.2 Secure Provisioning of Mobile Applications

The architecture proposed in this thesis has provided a framework to develop and deploy secure mobile transactions. It does not include secure provisioning of mobile applications, which requires extension of the proposed architecture in this thesis. Examples of mobile application provisioning systems are based on the concept of Trusted Service Managers (TSM) and Over-the-Air (OTA) provisioning protocols.

9.2.3 Privacy for Mobile Users

There can be various service providers integrated under the proposed architecture, providing different mobile services. Therefore, the privacy issues related to user information shared with multiple service providers must be addressed. In addition, a solution to provide anonymity for users while conducting mobile transactions is necessary.

9.2.4 UICC as Secure Element

The approach of using UICC as secure element for mobile applications/services is the trend for secure mobile transactions, because of security properties of UICC chips. This approach is included in the trusted layer model proposed by this dissertation. However, due to limited access to UICC chips, this approach has not yet been implemented and tested. Therefore, one research activity in the future is the design and development of UICC-based secure mobile applications.
## APPENDIX A: LIST OF ABBREVIATIONS

The following table describes the significance of various abbreviations and acronyms used in this thesis. The page on which each one is defined or first used is also given.

<table>
<thead>
<tr>
<th>Abbreviations</th>
<th>Explanation</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM</td>
<td>Automatic Teller Machine</td>
<td>19</td>
</tr>
<tr>
<td>3G</td>
<td>3rd generation mobile telecommunications</td>
<td>19</td>
</tr>
<tr>
<td>4G</td>
<td>4th generation mobile telecommunications</td>
<td>19</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
<td>20</td>
</tr>
<tr>
<td>MNO</td>
<td>Mobile Network Operator</td>
<td>20</td>
</tr>
<tr>
<td>DSRM</td>
<td>Design Science Research Methodology</td>
<td>21</td>
</tr>
<tr>
<td>HDML</td>
<td>Handheld Device Markup Language</td>
<td>23</td>
</tr>
<tr>
<td>XHTML</td>
<td>eXtensible Hyper Text Markup Language</td>
<td>23</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code division multiple access</td>
<td>23</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
<td>23</td>
</tr>
<tr>
<td>iOS</td>
<td>iPhone Operating System</td>
<td>24</td>
</tr>
<tr>
<td>SOA</td>
<td>Service-Oriented Architecture</td>
<td>25</td>
</tr>
<tr>
<td>CMP</td>
<td>Certificate management protocol</td>
<td>26</td>
</tr>
<tr>
<td>PPA</td>
<td>Pre-paid account</td>
<td>27</td>
</tr>
<tr>
<td>IDMS</td>
<td>Identity Management System</td>
<td>27</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
<td>27</td>
</tr>
<tr>
<td>AAA</td>
<td>Authentication, Access control and Authorization</td>
<td>27</td>
</tr>
<tr>
<td>UDDI</td>
<td>Universal Description, Discovery and Integration</td>
<td>31</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
<td>31</td>
</tr>
<tr>
<td>DII</td>
<td>Dynamic invocation interface</td>
<td>32</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
<td>32</td>
</tr>
<tr>
<td>J2ME</td>
<td>Java Platform, Micro Edition</td>
<td>32</td>
</tr>
<tr>
<td>SSO</td>
<td>Single sign-on</td>
<td>33</td>
</tr>
<tr>
<td>FIM</td>
<td>federated identity management</td>
<td>33</td>
</tr>
<tr>
<td>SP</td>
<td>service provider</td>
<td>33</td>
</tr>
<tr>
<td>IdP</td>
<td>identity provider</td>
<td>34</td>
</tr>
<tr>
<td>MITM</td>
<td>man-in-the-middle attack</td>
<td>34</td>
</tr>
<tr>
<td>RA</td>
<td>Registration authority</td>
<td>35</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate authority</td>
<td>35</td>
</tr>
<tr>
<td>WAP</td>
<td>Wireless Application Protocol</td>
<td>35</td>
</tr>
<tr>
<td>SIM</td>
<td>subscriber identification module</td>
<td>36</td>
</tr>
<tr>
<td>OCSP</td>
<td>Online certificate status protocol</td>
<td>37</td>
</tr>
<tr>
<td>CRL</td>
<td>Certificate revocation list</td>
<td>37</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal identity number</td>
<td>38</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
<td>39</td>
</tr>
<tr>
<td>WLAN</td>
<td>wireless local area network</td>
<td>39</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal digital assistant</td>
<td>39</td>
</tr>
<tr>
<td>QR code</td>
<td>Quick response code</td>
<td>40</td>
</tr>
<tr>
<td>USSD</td>
<td>Unstructured Supplementary Service Data</td>
<td>40</td>
</tr>
<tr>
<td>OTP</td>
<td>One-time pad</td>
<td>41</td>
</tr>
<tr>
<td>APDU</td>
<td>application protocol data unit</td>
<td>46</td>
</tr>
<tr>
<td>API</td>
<td>Application programming interface</td>
<td>46</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
<td>48</td>
</tr>
<tr>
<td>NFC</td>
<td>Near Field Communication</td>
<td>48</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------</td>
<td>----</td>
</tr>
<tr>
<td>GPRS</td>
<td>General packet radio service</td>
<td>50</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
<td>52</td>
</tr>
<tr>
<td>OTA</td>
<td>Over-the-air</td>
<td>68</td>
</tr>
<tr>
<td>SHA</td>
<td>Secure Hash Algorithm</td>
<td>77</td>
</tr>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard process</td>
<td>77</td>
</tr>
<tr>
<td>XACML</td>
<td>eXtensible Access Control Markup Language</td>
<td>79</td>
</tr>
<tr>
<td>PEP</td>
<td>Policy Enforcement Point</td>
<td>80</td>
</tr>
<tr>
<td>FSM</td>
<td>Finite State Model</td>
<td>96</td>
</tr>
<tr>
<td>EMV</td>
<td>Europay, MasterCard and VISA</td>
<td>99</td>
</tr>
<tr>
<td>OTC</td>
<td>Over-the-counter</td>
<td>100</td>
</tr>
<tr>
<td>SMSC</td>
<td>Short message service center</td>
<td>103</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
<td>105</td>
</tr>
<tr>
<td>DES</td>
<td>Data Encryption Standard</td>
<td>113</td>
</tr>
<tr>
<td>PAN</td>
<td>Primary Account Number</td>
<td>113</td>
</tr>
</tbody>
</table>
APPENDIX B: MOBILE APPLICATIONS

B.1 Mobile Banking (m-Banking)

Mobile banking (m-Banking) involves use of a mobile phone or another mobile device to perform various financial transactions with a client's bank account. Mobile banking is one of the newest approaches to provision of financial services through GSM or wireless Internet network, made possible by the widespread adoption of mobile phones, even in low-income countries. The rollout and functional capabilities of mobile telephony have been rapid and have extended usage well beyond classical (telephone calls and short messaging) applications. There is mounting evidence of positive financial, economic and social impacts of those technologies all over the world.

B.1.2 Cash Dispensing: Mobile and Static ATM

Mobile and static ATM are two innovative approaches to dispensing of cash, especially suitable in regions where there are no standard banking ATMs. With "Mobile ATM", cash will be distributed by the specialized bank agents, located areas without standard ATMs. They will dispense cash upon receipt of the authorization messages from bank servers. The procedure is equivalent to static ATMs, where cash will be dispensed by post offices, eventually merchants, and other cash distribution agencies.

The sequence of steps and exchange of messages for this transaction are the following: Customer who needs cash comes to the location of the Mobile (Bank Agent) or Static ATM. Using his/her phone, the customer sends Cash_Request message to the specialized bank server through the portal servers. The bank server has direct connection into the banking network and verifies the status of the customer’s account. If the confirmation is received from the bank, the server sends Cash_Confirmation message to the Bank Agent or corresponding cash dispensing agencies.
agent (like Post Office). When the message is received, cash can be given to the customer. The sequence of messages is shown in Figure 56.

Transactions of the Mobile-ATM system are explained below. In order to perform a transaction, a customer with a mobile phone comes to the Mobile-ATM agent, who has another mobile phone. Figure 57 illustrates message flows in the Mobile ATM system.

1. A customer visits a Mobile-ATM agent and sends a secure SMS message to the bank (withdrawal request) with Mobile-ATM agent's (Mobile-ATM) phone number and requested cash amount.
2. The bank verifies status of the customer's account and if OK sends an authorization SMS message to the customer together with a confirmation number (a random number).
3. At the same time, the bank sends a payment authorization SMS message to the Mobile-ATM agent (Mobile-ATM) together with a transaction number (a random number which is different from the confirmation number).
4. The customer tells the confirmation number to the Mobile-ATM agent (Mobile-ATM).
5. The Mobile-ATM agent (Mobile-ATM) sends a confirmation SMS message to the bank together with the transaction and the confirmation number.
6. The bank transfers the amount from the customer's account to the Mobile-ATM agent's (Mobile-ATM's) account and sends a transaction confirmation SMS to the Mobile-ATM agent.
7. The bank also sends a transaction confirmation SMS message to the customer.
8. Mobile-ATM agent hands in the money to the customer.

Two random numbers are used in all transactions to order to provide non-repudiation. Moreover, it is a strong evidence to confirm that the transaction has been fully completed.
B.1.3 Account-to-Account Transfers Transaction

This transaction may be performed between two personal accounts or between a personal and a corporate account. In both cases one customer is the sender (initiator of the transactions) and the other customer is the recipient. This transaction may be used for remittance, personal payments, bill payments, etc. It may be performed between two customers with accounts in the same bank or with accounts in different banks.

If the two customers have accounts in the same bank, then the sender initiates the transfer of certain amount of money from his/her account to the account of the recipient. Transfer Request message is sent from the sender’s mobile phone to the server, which, after verification and effective transfer performed by the bank, informs the recipient about the transfer.

If recipient’s account is in another bank, then after receiving authorization for the transfer from the sender's bank, sender’s mobile bank server will inform recipient’s mobile bank server about the transfer. Recipient’s server will notify recipient’s bank and the recipient.

B.1.4 Application and Administration of Loans

The SMSOA system also supports various transactions for administration of loans. Those could be mortgages, home equity loans, or micro-loans in developing countries. Applicants may apply for a loan and after approval, loan provider may transfer the amount to the applicant’s account using account-to-account transaction, described in the previous section.

Applicants may also administer their loans, such as reviewing the status of the loan, payment schedule, initiating payments, etc. For this transaction, the messages are the
following: applicant applies for a loan. Applicant’s portal server will pass the application to the portal server of the loan provider. When approved, applicant’s portal server and bank server will be notified and the loan will be activated. Finally, the applicant may also use various transactions with its portal server to administer the loan. The process is shown in Figure 59.

B.2. Mobile Commerce (m-Commerce)

Mobile-Commerce is the latest concept of enabling financial transactions, such as mobile payments using mobile phones and hand-held devices. With the rapid development of the society, m-Commerce applications play a vital role. This section describes mobile commerce transactions supported by our system. m-Commerce are mainly payment transactions and digital cash dispense transaction (cash in/ cash out).

B.2.1 Transactions and Components

Mobile commerce supports broad range of financial transactions, such as:

- Registration of merchants and customers,
- Enrollment of merchants and customers for issuance of the smart card,
- Opening of merchant’s and customer’s PPAs associated with cards,
- Accepting OTC payment transactions using smart cards for authentication of customers,
- Accepting cash as deposits to customers’ PPAs.
B.2.2 Operations of m-Commerce System

Customers can perform mobile commerce transactions in two ways, one is Over-the-Counter (OTC) and the other is Over-the-Air (OTA). For OTC transactions, customers use biometrics chip cards and merchants use specialized PoS terminals capable of reading smart cards. For OTA transactions, both customers and merchants use mobile phones and a secure mobile wallet loaded into the phone. The cards used in the system are issued and managed by our Smart Card Management System, appropriately modified to manage financial smart cards. On-line transactions using PPAs for customers, agents and merchants are handled as follows: when using smart cards, by PoS application extended with the Smart card client, and when using mobile phones, by the Secure Mobile Wallet loaded into the phone.

B.2.2.1 Digital Cash Dispensing and Micropayments

Instead of cash, our system can also distribute “digital cash” which is stored in mobile phone and later used for micro-payments. The prerequisite for this application is that merchants’ PoS terminal is equipped with hardware and software supporting appropriate proximity protocol and micro-payment application. If so, customers do not need cash, because they are using bank agents or cash dispensers described in the previous section. The sequence of steps and transactions is the following: Customer sends Cash_Request to the portal server. After validation as before, “digital cash” is debited from customer’s account, transferred to and stored in his/her mobile phone. Thus, mobile phone becomes “digital wallet”. When the customer comes to the PoS, he/she performs payment transaction using mobile phone. The payment amount is reduced from the customer’s “digital wallet” and transferred to the merchant’s PoS terminal. It sends Cash_Reclaim message to the portal server which contacts merchant’s bank server to make deposit into the merchant’s account.
B.2.2.2 Credit/Debit Card Payment

Standard debit and credit card payments are today performed using plastic debit/credit cards with magnetic stripe and somewhere with chips. In our system, magnetic stripe data (credit card number and other data) are stored in the mobile phone. Merchant’s PoS terminal must be capable to accept such data through proximity protocol. All other steps with this application are the same as in today’s debit and credit card transactions. Debit/credit card data are entered into the customer’s mobile phone either during registration or during the process equivalent to debit/credit card issuance.

As shown in Figure 62, the process is the following: customer uses his/her mobile phone to provide card number and other data to the merchant’s PoS terminal through the proximity protocol. Merchant either connects to the portal server to verify the authorization of the transaction or connects directly to the existing Card Payment Gateway. When the authorization is received, the payment transaction is completed. Later, merchant sends Credit_Request message to the portal server or Payment Gateway to request payment.

B.3 Mobile Parking (m-Parking)

Nowadays most parking systems work like this: customers estimate their parking time before parking and pay the parking fees in advance. This is not convenient, since customers have to remember their parking expiration time. Our system can be extended to support pay-by-cell phone parking transactions, where customers can send parking payment using SMS messages and the parking bill will be either paid using pre-paid account or later delivered with the mobile phone monthly fees. This approach reduces the cost of time during parking procedure. It also allows customers to park their cars as long as they want, without estimating the time before parking.
B.3.1 Transactions and Components

Our m-Parking system provides the following functions to three types of uses:

- For customers (drivers), registering their identification and financial data, paying parking using cell phones, sending warning messages, and inquiring regarding payment transactions and account status,
- For Parking Enforcement Staff, reviewing of payment status for individual parking spaces for issuance of parking tickets,
- For Parking Authorities, it provides a wide range of reporting, security, and financial services.

The following components are included in our parking system and the architecture is shown in Figure 63:
Our m-Parking system provides user-friendly experience that will gain rapid popularity and promote quick uptake. It enables the user to send parking space and selected parking time data directly using user’s mobile phone with simple SMS messages. Registration data are entered on-line using parking website, where new users will enter basic driver and payment information necessary to link each user’s virtual account to a payment method.

**B.4 Mobile Ticketing (m-Ticketing)**

Mobile ticketing is an application that enables customers to inquire, order, pay for, obtain and validate tickets from any location and at any time using mobile phone or other mobile handsets. It reduces the production and distribution costs of traditional paper-based ticketing channels and increases customer convenience by providing new and simple ways to purchase tickets.

**B.4.1 Transactions and Components**

Nowadays, there are many websites selling tickets online. Using our system, this service can be extended to provide secure mobile transactions – purchasing tickets using mobile phones. This system will provide inquiry about all shows for which tickets are available, inquiry about pricing of individual tickets, and ordering of tickets using mobile phones. Our m-Ticketing system provides the following transactions to two types of users.

- For customers, registering their identification and financial data, booking and paying ticketing using cell phones, inquiring regarding tickets information and account status,
- For Tickets Distributors, publishing tickets information, manage customers’ registration and tickets data, issuing electronic tickets to customer’s mobile phone and verifying tickets.

The architecture for mobile ticketing system is shown in Figure 64. After installing, configuring and activating all system components, Events and Accounts Database

![Figure 64. System Architecture for Mobile Ticketing](image-url)
must be populated first. For that, Ticket Web Server must be modified to feed all those information to the Mobile Tickets Server and Mobile Payments Server. Events Database will contain one entry for each seat at each event. This database will be populated continuously as new events are announced and new customers are registered in the system using Ticket Web Server.

After that, ticketing and payment transactions may be performed. For each transaction, based on the mobile number, the system recognizes the customer, the event, and the selected seat, and creates ticket purchase transaction in the Tickets Database. At the same time, the system triggers payment transaction by the Mobile Payment Server.

If tickets are sold over Web, Ticket Web server must be modified to provide on-line and in real–time two types of information to our system:

- Information about new events, seats, tickets, prices, etc., i.e. all information needed for mobile customers to select the desired tickets,
- Registration data for customers, including their payment options, bank accounts (if any), etc.


Smart Communications, Inc. SMART Money. n.d.  

SPIDER. Mobile ATMs for Developing Countries. 2007.  


SunTrust. Mobile Banking. n.d.  
https://www.suntrust.com/PersonalBanking/EverydayBanking/MobileBanking (accessed November 22, 2010).


