Data Authentication Principles for Online Transactions

Sijan Gurung
Spring 2016
Acknowledgement

First of all, my sincere gratitude goes to my supervisor Professor Audun Jøsang for providing me with guidance and continuous supervision. His suggestions and feedbacks were very helpful and crucial for the completion of this thesis.

Also, I am grateful to the partners and the team of OffPAD project for providing me with the platform for implementing the project. I also want to acknowledge TazTag for providing me with TazTag device needed for the project. I was fortunate to participate in OffPAD workshops, from where I was able to get vital information that helped me during the project. Therefore, special thanks go to for Dr. Léonard Dallot, project manager for TazTag.

Finally, I must express my very profound gratitude to my friends and family for providing me with encouragement and unceasing support throughout the journey of this thesis. Special thanks go to my wife Pratibha Shrestha and my friend Suraj Shrestha for helping me with the correction and grammar checks during the final phase of thesis writing.

Most of this would not have been possible without them. Thank you all.
Abstract

In the online banking systems, authentication has always been the focus point due to increasing security threats. There have been various authentication systems tested and implemented for securing the user and data authentication. Among all entities of the online banking system, the client is the most vulnerable entity in terms of security threats. So the main focus of this thesis was to provide an authentication mechanism that preserves security even if the client is infected or has security vulnerabilities.

An online banking system implies various methods of authentication such as two-factor authentication, where an additional credential is added for authentication. But even in two-factor authentication, preventing that the occurrence of alteration of transaction details is very difficult. Users are not aware if the details have been altered on the client before sending it for further processing. This may lead to false transactions being successfully completed. So there has to be some mechanism that prevents any alteration in data, leading to rejection of the corrupted transaction. So, we implemented a system where any manipulation in the transaction details is detected and where such transactions are prevented from being executed.

Our system consists of a separate device, in which, the user enters transaction details, i.e. destination account and the amount. These values are directly passed to client browser via Bluetooth. This means that the user does not have to enter vital data of a transaction in the client browser. The transaction is then sent for further processing to the bank where it would be analyzed, and where the validity of the transaction is ascertained. In other words, the server effectively determines the authenticity of the transaction just by analyzing the received data. This system eliminates the use of an additional channel for the authentication purpose such as SMS.

Even though there were quite a few assumptions in the implementation of the system, we were able to achieve the principal goal. So by implementing data authentication in OffPAD using an
authenticated hash, the transactions could be successfully verified as authentic or not and, any unknown modification in the details could be detected precisely.

By using a dedicated device as an OffPAD for user and data authentication, we can successfully ensure the authenticity of the transactions and secure online banking system with a strong authentication mechanism in place.
Acronyms

A2DP  Advanced Audio Distribution Profile
ACK   Acknowledgement
API   Application Programming Interface
ATM   Automated Teller Machine
CA    Certificate Authority
CD    Checking Disabled
DCH   Dedicated Channel
DDoS  Distributed Denial of Service
DNS   Domain Name System
DNSSEC Domain Name System Security Extensions
DOI   Digital Object Identifier
DOM   Document Object Model
DSA   Digital Signature Algorithm
EDR   Enhanced Data Rate
GGSN  Gateway GPRS Support Node
GPRS  General Packet Radio Service
GSM   Global System for Mobile Communication (2G)
GSMA  Global System Mobile Association
HMAC  Hashed Message Authentication Code
HRL   Home Location Register
HSBC  Hongkong Shanghai Banking Corporation
ICANN International Corporation for Assigned Names and Numbers
ISI   Information Sciences Institute
ITU   International Telecommunications Union
ITU-T International Telecommunications Union-Telecommunications section
JRE   Java Runtime Environment
KID   Kundeidentifikasjon
LCD   Liquid Crystal Display
LE    Low Energy
LED   Light Emitting Diode
LUCIDMAN  Local User Centric ID Management project
MAC   Message Authentication Code
NS    Name Server
NSEC  Next Secure
OAEP  Optimal Asymmetric Encryption Padding
OCR   Optical Character Recognition
OffPAD Offline Personal Authentication Device
OS    Operating System
OSI   Open Systems Interconnection
OTP   One-Time Password
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAD</td>
<td>Personal Authentication Device</td>
</tr>
<tr>
<td>PGT</td>
<td>Password Generating Token</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number</td>
</tr>
<tr>
<td>PKI</td>
<td>Public-Key Infrastructure</td>
</tr>
<tr>
<td>RPF</td>
<td>Reverse Path Forwarding</td>
</tr>
<tr>
<td>RRSIG</td>
<td>Resource Record Signature</td>
</tr>
<tr>
<td>RSA</td>
<td>Rivest-Shamir-Adleman algorithm</td>
</tr>
<tr>
<td>SET</td>
<td>Secure Electronic Transactions</td>
</tr>
<tr>
<td>SGSN</td>
<td>Serving GPRS Support Node</td>
</tr>
<tr>
<td>SHA</td>
<td>Secure Hash Algorithm</td>
</tr>
<tr>
<td>SLD</td>
<td>Second Level Domain</td>
</tr>
<tr>
<td>SIM</td>
<td>Subscriber Identity Module</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SOBC</td>
<td>Secure Online Banking Companion</td>
</tr>
<tr>
<td>SPF</td>
<td>Sender Policy Framework</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layers</td>
</tr>
<tr>
<td>SYN</td>
<td>Synchronization</td>
</tr>
<tr>
<td>TCB</td>
<td>Transmission Control Block</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TLD</td>
<td>Top Level Domains</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>TMSI</td>
<td>Temporary Mobile Subscribe Identity</td>
</tr>
<tr>
<td>TTL</td>
<td>Time to live</td>
</tr>
<tr>
<td>TXT</td>
<td>Text</td>
</tr>
<tr>
<td>UK</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>US</td>
<td>United States</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>USIM</td>
<td>Universal Subscriber Identity Module</td>
</tr>
<tr>
<td>USPI</td>
<td>User Specific Personal Images</td>
</tr>
<tr>
<td>USSD</td>
<td>Unstructured Supplementary Services Data</td>
</tr>
<tr>
<td>UUID</td>
<td>Universally Unique Identifier</td>
</tr>
<tr>
<td>VLR</td>
<td>Visited Location Register</td>
</tr>
<tr>
<td>WAP</td>
<td>Wireless Application Protocol</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WTCA</td>
<td>WebTrust for CA</td>
</tr>
<tr>
<td>WTLS</td>
<td>Wireless Transport Layer Security Specification</td>
</tr>
<tr>
<td>iKP</td>
<td>Internet Keyed Payments</td>
</tr>
</tbody>
</table>
Table of Contents

Chapter 1  Introduction ........................................................................................................ 1
  1.1  Focus .............................................................................................................................. 2
  1.2  Motivation .................................................................................................................... 2
  1.3  Research Question ....................................................................................................... 3
  1.4  Related Work and Projects .......................................................................................... 3
       1.4.1  Secure Online Banking Companion (SOBC) Device: .............................................. 3
       1.4.2  User Specific Personal Images (USPI): ................................................................. 4
       1.4.3  The Cronto System ................................................................................................ 4
       1.4.4  SMS - Based Authentication ................................................................................. 4
  1.5  Online Banking and E-commerce ............................................................................... 5

Chapter 2  Background ........................................................................................................ 6
  2.1  Authentication ............................................................................................................. 6
       2.1.1  Entity Authentication .............................................................................................. 6
       2.1.2  Data Authentication ............................................................................................. 10
  2.2  X.800 Recommendation- ITU-T ................................................................................ 11
       2.2.1  Services .................................................................................................................. 11
       2.2.2  Mechanisms ........................................................................................................... 12
       2.2.3  Attack ..................................................................................................................... 14
  2.3  Threats specific to Online banking / E-commerce ....................................................... 15
       2.3.1  Phishing – Identity theft attacks ............................................................................. 16
       2.3.2  Man in the Middle – Channel Breaking attacks ...................................................... 16
       2.3.3  Man in the Browser – Content Manipulation .......................................................... 17
  2.4  Server Authentication .................................................................................................. 18
       2.4.1  Types of Server Authentication .............................................................................. 20
       2.4.2  Domain Name System ............................................................................................ 20
  2.5  DNSSEC (Domain Name System Security Extensions): .............................................. 23
       2.5.1  How does DNSSEC work? ..................................................................................... 25
       2.5.2  Other techniques for countermeasure ................................................................... 26
  2.6  Mobile Banking ........................................................................................................... 28
       2.6.1  Mobile Transaction Procedure .............................................................................. 30
       2.6.2  SMS Banking .......................................................................................................... 31
       2.6.3  USSD Banking ......................................................................................................... 32
       2.6.4  Mobile Banking in GPRS ...................................................................................... 32

Chapter 3  Research Method ............................................................................................... 36

Chapter 4  System Overview .............................................................................................. 38
  4.1  Requirement study ....................................................................................................... 38
       4.1.1  OffPAD device ......................................................................................................... 38
       4.1.2  Client / Browser System .......................................................................................... 38
       4.1.3  Server System .......................................................................................................... 38
  4.2  Assumptions ................................................................................................................ 39
       4.2.1  Shared Key ................................................................................................................. 39
4.2.2 OffPAD as a dedicated device ............................................................. 40
4.2.3 OffPAD solution as an application ..................................................... 42
4.3 Prototype design and development ....................................................... 43
4.3.1 Device ............................................................................................. 43
4.3.2 Implementation .............................................................................. 43
4.3.3 Application Prototype ................................................................... 48
4.3.4 Client Server Prototype .................................................................. 53

Chapter 5 Technical Description .................................................................. 59
5.1 Choice of technology ........................................................................... 59
5.1.1 RSA-Asymmetric cryptography ......................................................... 59
5.1.2 SHA-256 Hash ............................................................................. 60
5.2 Devices and Development .................................................................. 61
5.2.1 Android Device .......................................................................... 61
5.2.2 Client and Server System ............................................................... 61

Chapter 6 Discussion .................................................................................. 63
6.1 Threats to Online-banking: ................................................................. 63
6.1.1 Phishing: ..................................................................................... 63
6.1.2 Man in the Middle: ..................................................................... 63
6.1.3 Man in the Browser: ................................................................. 63
6.2 Entity Authentication: ....................................................................... 64
6.3 Focus of Thesis: .................................................................................. 65

Chapter 7 Conclusion .................................................................................. 68
7.1 OffPAD as a mobile application: ......................................................... 68

Chapter 8 Limitations and Future work ...................................................... 70
8.1 Diverse device support ....................................................................... 70
8.2 Use of SHA-256 and RSA ................................................................. 70
8.3 Client-Server Security Mechanism ...................................................... 71
8.4 Shared Key ....................................................................................... 72
8.5 Future Work ..................................................................................... 72
8.5.1 OffPAD as a Separate Device ....................................................... 72
8.5.2 User Authentication ..................................................................... 73
8.5.3 Key Sharing / Signing Mechanisms ................................................ 73
List of Figures

Figure 1: Trusted Interactions in malicious environment .......................................................... 9
Figure 2: Authentication class for user-side entities from Figure 1 ............................................. 10
Figure 3: Authentication class for SP-side entities as shown in Figure 1 .................................... 10
Figure 4: DNSSEC validation process ...................................................................................... 25
Figure 5: Mobile Banking Trusted Service Manager ................................................................. 34
Figure 6: Shared key encryption ............................................................................................... 40
Figure 7: One-time code generator with numeric keypad .......................................................... 41
Figure 8: OTP without PIN identification .................................................................................. 41
Figure 9: OTP with PIN-code ................................................................................................... 41
Figure 10: The basic working diagram of proposed system ...................................................... 47
Figure 11: OffPAD main menu .................................................................................................. 49
Figure 12: OffPAD home screen .............................................................................................. 49
Figure 13: Bluetooth turn on access .......................................................................................... 50
Figure 14: Bluetooth status information ................................................................................... 50
Figure 15: Transfer Information ............................................................................................... 51
Figure 16: Transfer success information ................................................................................... 51
Figure 17: Transfer failure ........................................................................................................ 51
Figure 18: Transfer history in ascending order of date ............................................................. 51
Figure 19: Transaction history in descending order of date ...................................................... 52
Figure 20: About information ..................................................................................................... 52
Figure 21: OffPAD Client overview .......................................................................................... 53
Figure 22: Instructions for transfer operation ............................................................................. 54
Figure 23: Injected Information .................................................................................................. 55
Figure 24: Adding additional fields ........................................................................................... 55
Figure 25: Transfer request processing in server ....................................................................... 56
Figure 26: History showing transaction details .......................................................................... 56
Figure 27: Value altered by Trojan ........................................................................................... 57
Figure 28: Transaction rejected due to inconsistency .................................................................. 57
Figure 29: History showing failed status ................................................................................... 58
Figure 30: Information about the client ..................................................................................... 58
Figure 31: 1024-bit public key sample ....................................................................................... 60
Figure 32: 1024-bit private key sample ....................................................................................... 60
Figure 33: SHA-256 Hash function snippet ................................................................................. 61
Figure 34: Transaction table schema for MySQL ....................................................................... 62
List of Tables

Table 1: Resource record types for DNSSEC ................................................................. 24
Table 2: Steps in a DNS query and response with optional DNSSEC data ....................... 25
Chapter 1

Introduction

Authentication has evolved along with the evolution process of information technology, network services and the Internet. With the advent of the Internet, the need to secure information is continuously growing. Security of information remains vital to business organizations as well as to individual users. In the current context, security of information for businesses regarding their clients as well as their own operations plays a vital role for the success or failure of any organization. The ability to access information from anywhere and at any time through different cloud computing solutions has increased challenges for information security and authentication. The use of Internet offers immense capabilities to remain connected to different users who are geographically separated. Such remote connectivity requires a higher level of security procedures, and brings new threats in terms of use of false identities, data corruption, unauthorized access, etc.

In previous years, single workstations without any Internet connections were accessed using username and password combinations. However, in today’s world of interconnected networks, this simple mode of authentication creates vulnerabilities.

Tampering with data is another important issue regarding authentication. Tampering with data by Trojans can have severe implications. These alterations can lead to corruption or deletion of data. Users might not have any information regarding such attacks until the entire transaction is completed. Therefore, a concept of a system which can take precautionary actions to discover any alterations and prevent completion of corrupted transactions was the main goal this project.

In this thesis, we propose the use of SHA-256 hash value function for data authentication purpose while the transaction or transmission of data is encrypted using RSA asymmetric encryption.
1.1 **Focus**

This Master’s thesis focuses on the theory of authentication of data using a device. The device is being developed as a part of the OffPAD project. The OffPAD concept is an Offline Personal Authentication Device. We intend to work on the process of combining OffPAD through a browser-based system for the client. The SHA-256 hash function is the basis for this data authentication. Beyond this major focus, the thesis also focuses on securing the transmission of data, which is encrypted using RSA asymmetric encryption. Therefore, this project delivers a concept of data authentication using SHA-256 hash function as well as data security during transmission using RSA asymmetric encryption.

1.2 **Motivation**

This project is an extension of previous research focused on the development of a new protocol for data authentication to be commercially used on existing online banking systems. This research is an extension of the OffPAD project funded by the Eurostars program, following the LUCIDMAN project funded by The Franco-Norwegian Foundation.

Data authentication is an integral part of the entire authentication process. It inherently is different from user authentication, as user authentication normally takes place at the initial stage before any transaction has occurred. Data authentication is the process of revealing original or corrupted data (Sabaa & Panda, 2007). Tampering with data or alterations are some of the biggest threats to data authentication. Therefore, when it comes to data authentication, the process of data transmission and transaction must also be secured. Storage of data can be either remote or local and modes of accessing such stored data can be numerous. This further complicates the process of securing the data. Transactions as mentioned above, can be either online or offline. Encryption of data is one means of securing it.
There are different types of attacks or malware such as Trojans, viruses, and phishing attacks, worms and advanced persistent threats / spear-phishing.

Advanced persistent threats or spear-phishing is one of the most common methods of attacking enterprises or organizations by masquerading a Trojan in emails to a large number of recipients, which is bound to be opened up by some of them. The Trojan can then alter data or user information, which causes corruption of data.

1.3 Research Question

There are three research questions, which we will answer in this thesis. The research questions focus on the development of a data authentication protocol or mechanism, which can be commercially used by existing online, banking systems.

- How to authenticate data using SHA-256 hash value function?
- How to secure data transactions using RSA asymmetric encryption?
- How to secure online banking transactions and apply data security?

1.4 Related Work and Projects

1.4.1 Secure Online Banking Companion (SOBC) Device:

The SOBC device is a smart card based device developed by Peng, Chen, Chang and Guan (Peng, Chen, Chang, & Guan, 2010), which can provide security for users to conduct online banking over insecure computers. The SOBC device consists of a smart card and a portable USB (Universal Serial Bus) sized smart card reader. This device consists of a USB port, small screen and two control buttons (Yes/No). The issuer initializes this device before handing it over to the customer. The device is based upon RSA (Rivest-Shamir-Adleman) asymmetric encryption and an X.509 certificate is generated for the device as well. This device creates a mutual TLS (Transport Layer Security) connection and it offers encryption/decryption capabilities.
1.4.2 User Specific Personal Images (USPI):
A user selects a specific image at one of the branches of the bank. This image is stored in the bank’s server. When a user initializes an online transaction, the bank server responds with a USPI image along with the confirmation details. The USPI consists of the image that the user has selected as well as the account number of the recipient embedded in the image. The USPI image is the backbone of this process which is limited to the confidentiality of it in the bank’s server (Goyal, Bansal, & Gupta, 2013).

1.4.3 The Cronto System
The Cronto system, which was a research project, done at the University of Cambridge, gives a transaction authentication solution for online banking. When the customer initiates a transaction, the Cronto server-based system accepts the request and stores the details given by the client application, after which a new visual cryptogram is generated. This visual cryptogram is then passed on to the customer, containing the transaction details. The user must possess a mobile with the Cronto client software installed in it, and a normal mobile camera. The user can use the client software to take a photograph of the visual cryptogram provided by the bank, and the transaction details are extracted from it. The user can then verify transaction details and confirm it by entering a code which is generated in the client software (Alzomai, Alfayyadh, & Jøsang, 2010).

1.4.4 SMS - Based Authentication
The SMS-based authentication process relies on two factors for its success. The first factor is the use of a cellular network, which is independent of the Internet, thus, integrity factor is checked. Another factor is the human user, who can verify the transaction details sent to his/her mobile device. The process works in a fairly simple manner, as many previous authentication or authorization system. The user initiates the transaction process by providing credentials and transaction details which are sent to the server. The server sends the transaction data including an authorization code to the user. The user tallies the two transaction details, and if the integrity is maintained, then authorizes the transaction by inserting the accompanying code. The one issue on this process is its complete dependence on the human user (Alzomai et al., 2010).
1.5 Online Banking and E-commerce

Online banking systems have enabled account holders to take control over their bank accounts and perform transactions as felt necessary by them (Guraau, 2002). This suggests that an account holder can access available banking services from anywhere at any time via the Internet. Therefore, almost all banks offer online banking services based upon virtual banking systems (Karim, Rezaul, & Hossain, 2009). Online banking transactions are a collection of financial and personal or enterprise related information. Therefore, there is a further need to secure such information. Different financial institutions are under immense pressure from rival competitors to provide Internet-based customer services including online banking to attract customers. This also puts pressure on such financial institutions to safeguard crucial information regarding their customers. A small glitch in the security can have severe repercussions on the business and customers as well. According to a 2015 global report on online fraud panorama by Ingenico Payment services, it was estimated that 60% of card fraud in Europe is linked to online payments. This leads to loss of revenue for businesses, which can range from 0.3% to 3%.

E-commerce is another contributing factor in the rise of online banking and online transactions. Almost four percent of total retail sales in the United States of America is attributed to e-commerce sales, which translates to normally around 40 billion US dollars annually (C. Bureau “Statistics abstract of US 2010). Many e-commerce transactions are not directly dependent upon banks or financial institutions, but rather rely on third-party payment processing systems such as PayPal. Such an involvement of third party payment systems adds extra challenges to securing transactions. One of which is the challenge of cash out fraud; for example, someone can create a fake transaction and make the payment using the credit card, which will be paid to its own recipient account. This payment can then be cashed out from the recipient account without paying any extra fees to the bank or payment service. Therefore, online banking and e-commerce transactions have numerous complexities related to authentication. However, this is a growing industry that has an immense effect on the global economy, and although it has problems, there must be new means to secure such transactions in the future.
Chapter 2

Background

In this section, concepts related to authentication, online banking processes, and other relevant information security issues are discussed further. Literature relevant to these factors is also considered throughout this process of reflection.

2.1 Authentication

Authentication, in general, can be divided into two categories: Entity Authentication and Data Authentication. Although, we often consider these two categories to be quite similar, they are very different when it comes to why and how these authentications are carried out. While entity authentication always begins the process, data authentication comes into play after the user authentication is completed.

With this said, it is important to understand that although entity authentication and data authentication are two major modes of authentication, there are different methods used to employ such authentication tactics.

2.1.1 Entity Authentication

Entity authentication is the process of identification of correct users by the service providers and vice-versa. Therefore, an identity of a person or an organization must remain a unique factor, which is used to distinguish between different persons or organizations in particular domains. Entity authentication typically takes place at the beginning of a session or interaction. Entity authentication can be divided into three modalities: syntactic, semantic and cognitive authentication (Jøsang et al., 2015) described further below.

1. Syntactic authentication: The verification process between two entities where a source entity transmits its authentication credentials such as passwords to a recipient entity. The
recipient can then validate the authentication credentials. The recipient is indifferent to the authenticated entity, and does not apply any policy.

2. Semantic authentication: The verification process between two entities where a source entity transmits its authentication credentials to a recipient entity that validates it. However, in semantic authentication, an additional step of compliance testing with a security policy is also done (Jøsang et al., 2015).

3. Cognitive authentication: The verification process between a cognitive source entity and a recipient entity by transmission of authentication credentials which is validated along with a compliance testing to a security policy. In addition, this is done through, a user-friendly representation of identity attributes which enables the cognitive relying party to recognize and reason about policy compliance of source entity (Jøsang et al., 2015).

There are various means of authenticating human users, such as username/password combinations, and biometric testing. However, user authentication alone is insufficient to secure online transactions; therefore, data authentication is also required.

The concept of identity management is a central issue in the current context of rapidly increasing online services. There is a huge increase in the number of online services provided through the Internet.

With this increase of services, there is also a greater number of users who use them. Identity management under such a context becomes a challenge for the service provider, as well as for the user. The service provider must issue credentials and identifiers to the users, and when service operations are initialized by a request from the user, the service provider must validate user credentials and provide allocated functionalities and services for users. Most of the identity management systems are service provider centric (Jøsang et al., 2015). In this sense, a need for a user-centric identity management system was discovered.

The first thing necessary to understand in a user-centric identity management system is the use of identifiers and credentials by a user. A user depends on a more manual means of managing all the credentials for different service providers. Memorization of all these credentials and identifiers for different service providers is a daunting task indeed. Therefore, a new approach towards management of identity from the user’s perspective is required. This particular concept gave rise to a concept of Personal Authentication device (PAD). The PAD can validate a user with a PIN, after this process the PAD can be activated and used by a user for personal purpose, which makes it user-centric. The PAD can have cross-platform authentication abilities that can accommodate different authentication protocols. It could be connected via
communication channels such as Bluetooth or Wireless Local Area Network (WLAN). However, a dedicated device such as PAD can be difficult to carry in comparison to a mobile phone, which is already carried by almost everyone. Mobile phones have greater capacities in terms of computing, processing, transmission and security. However, with the tough competition in the mobile market, where usability and ease of access of mobile devices outweigh the need for security features, it becomes difficult for manufacturers to make normal mobile phones highly secured. The French company Taz-Tag has introduced a mobile phone (TPH-ONE) based on Android which also integrates a secure element which can be accessed in a secure state (Varmedal et al., 2013). Unlike other mobile phones including Taz-Tag, The OffPAD does not provide similar services as mobile phones such as web browsing, music or making phone calls, therefore, it will not serve any another purpose than authentication. In this sense, the OffPAD is targeted towards commercial adoption by those who are in need of high-security measures, however, some limitations in the form of economic viability are of some concern. The OffPAD can be comparatively more expensive than OTP (One Time Password), but the overall security provided by OffPAD and its services can outweigh this disadvantage.

One solution to the problem of memorizing passwords for different browsers and browser-enabled websites are the storage of such passwords within the browser’s security umbrella. This creates a dependency on the security provided by the service provider for authentication, whereas the OffPAD enforces the user to remain in control of all his/her credentials.

While user authentication is just one part of the solution, another part lies in the process of service-provider authentication. With this said, it is necessary to understand that unlike user authentication, service provider authentication consists of several complexities by nature. These complexities come in the forms of the scope of service providers, which can be global or local, stability and reliability.

Identity theft is a common problem in today’s world, where most of the transactions are done online. The identity of the user can be manipulated either by malware in the client platform, using a Man-in-the-Middle within the Internet or at the client. Therefore, to verify the identity of the user, the authentication process must start from the user-side itself regardless of the presence
of any malware. In the client, this technology is based on a device called OffPAD (Offline Personal Authentication Device) used as authentication device to perform trusted transactions. This project follows the LUCIDMAN project, which was a collaborative research project, funded by the Franco-Norwegian Foundation in the period 2011-2013.

The Authentication process has to gradually evolve along with the evolution of the entire computer system. With this fact, authentication is not just concerned with single or multiple human users, but is also related to automated systems. Therefore, there are two different types of entities, first is the human entity where there are users or organizations, and then there is another group called system entities which consist of clients or server systems (Varmedal et al., 2013).

OffPAD relies on the concept of a combination of a client system and a human user on the client side and the server system and a service provider organization on the other side as illustrated in Figure 1. There are multiple classes of authentication between these entities as illustrated in Figure 1, and explained in Figure 2 and 3. There are different modes of authentication, such as one-factor authentication, which relies on one authentication medium for example: a password. Another authentication mode is a two-factor authentication, which consists of a device such as One-Time Password (OTP) generator that can add an additional level of security. This two-factor method is one of the most relied-upon method of user authentication used by various financial institutions, and is commonly called strong authentication.

![Figure 1 Trusted Interactions in malicious environment](image)

---

Unlike the previous systems used for authentication, which relies on a computer or mobile client system, OffPAD is a dedicated device that is used as for identity management that can help to authenticate users as well as servers. It remains offline and adds a secure element to protect contents as well as the privacy of users (Varmedal et al., 2013).

2.1.2 Data Authentication

Although there is already considerable amount of research on user authentication, however, data authentication cannot be neglected. There are two types of authentication services according to International Telecommunication Union (ITU-T):

1. **Data origin authentication**

   Data origin authentication is described as the corroboration that the source of data received is as claimed (ITU-T, 1991). Data origin authentication does not provide any protection against duplication or modification of data. Its authentication remains limited to assurance of the source of the data is legitimate.

2. **Peer entity authentication**

   Peer entity authentication is described as the corroboration that the peer entity in an association is the one claimed (ITU-T, 1991). Its services provide assurance regarding the legitimacy of the involved entities in network sessions. Its use is specific to the confirmation of identities of involved entities from initialization of transaction or during the data transmission phases. It is specific to avoiding unauthorized access.
2.2 X.800 Recommendation- ITU-T

X.800 is recommendation security architecture from ITU-T for OSI (Open Systems Interconnection). It is one of the sought after standard by international security vendors. It focuses on three factors:

2.2.1 Services

Security service ensures the security of system and data. It is a service dependent on a protocol layer of the open communication system. X.800 services consists of five categories:

a. Authentication:
   Authentication is the service provided for authentication of a peer-entity or a data source.
   i. Peer Entity authentication services are provided by the (N)-layer, which ensures to the (N+1) entity that peer-entity is the claimed (N+1) entity (ITU-T, 1991).

   ii. Data origin authentication is provided by the (N)-layer, which ensures to the (N+1) entity that source of data is the claimed (N+1) entity (ITU-T, 1991).

b. Access Control:
   In OSI, this particular service is targeted towards unauthorized access of resources. These resources may or may not be OSI dependent resources. This service is implemented at the early stage where secure entities are identified, and unidentified entities are denied access.

c. Data Confidentiality:
   It consists of four types of confidentiality parameters, which protect data from unauthorized access and disclosure.

   i. Connection confidentiality:
      It provides security to data specific to users during specific connections.
   ii. Connectionless confidentiality:
      It provides security to data specific to users during single connectionless service data units.
   iii. Selective field confidentiality:
      It provides security to selective portions of users data specific to connections or on single connectionless service data units.
iv. Traffic flow confidentiality:
   It provides security to data, which might be extracted from the flow of traffic.

d. Data integrity:
   It is the security of data from origin to transmission, which means it provides assurance regarding the source of the data as well as the accuracy of data throughout the transmission process. It consists of five different forms of security aspects:

   i. Connection integrity with recovery:
      It provides integrity services for all user data on specific connections by detecting any sorts of modifications such as insertions, deletions, updates or reiteration of any data.

   ii. Connection integrity without recovery:
      It is similar to connection integrity with recovery besides the fact that it does not provide recovery services.

   iii. Connectionless integrity:
      It provides integrity services for a single connectionless single data unit which checks for any modifications and reiterations along the transmission process.

   iv. Selective field connection integrity:
      It provides selective integrity on specific fields of the user data over specific connection and checks for any modifications such as inserts, deletes or reiterations.

   v. Selective field connectionless integrity:
      It provides selective integrity on specific fields of the user data over a single connectionless single data unit and checks for any sorts of modifications.


e. Non-repudiation:
   It consists of any of two forms proof of origin or proof of delivery, which is described below:

   i. Non-repudiation with proof of origin:
      The proof of origin of data is provided to the recipient, which protects against any false claims of denial of sending data.

   ii. Non-repudiation with proof of delivery:
      The proof of delivery of data is provided to the sender, which protects against any false claims of denial of receiving data.

2.2.2 Mechanisms

a. Encryption
   It provides security through confidentiality of data or traffic. It works on the principle of use of ciphers which are basically codes used to encrypt and decrypt messages.
   These ciphers are of two types:
1. Symmetric (Shared-Key Cryptography)

In symmetric cipher, the same keys are used to encrypt and decrypt by sender and receiver. The user is in possession of a secret key, which remains unique as well. The user can then authenticate him by sharing this secret key to the authentication server. The user sends the username and a randomly generated number or code, which is encrypted using a secret key to the server. The server consists of the shared secret key, which is used to match the received encrypted code, and if the match is absolute, the user is authenticated.

OTP (One Time Password) tokens are a tangible implementation of symmetric-key.

2. Asymmetric:

Asymmetric cipher, which is also known as public-private key cryptography, uses two sets of keys. Unlike symmetric cipher where the same keys are used, here the public keys are known but the private key remains secret to the specific user. This creates a higher level of security.

b. Digital Signatures

It consists of two procedures:

1. Signing of data unit
   It is done using two methods:
   i. Encryption of data
   ii. Creation of a cryptographic key or check value for the data

2. Verification of data unit
   It is done using a publicly known process to validate the signature, which remain private to its owner.

c. Data integrity mechanisms

The integrity of data is related to either single data unit or a stream of data. The integrity of single data is maintained by placing checkpoints at sending entity and receiving entity. The sending checkpoint assigns a check-value or key to the data while the receiving entity produces exact check-value, which ensures the integrity of data. However, for connection-mode data transmission, additional forms of integrity checks are required such as time-stamping, sequence-numbering or secondary encryption. This creates a secondary layer of integrity check to protect against several threats such as replay, loss, disorder or mismatch.
d. Authentication exchange

This authentication mechanism consists of three different techniques:

i. Use of authentication information such as passwords

ii. Cryptographic techniques

iii. Use of characteristics or possessions of entity

e. Traffic-padding mechanism

This mechanism is used to provide protection against traffic pattern analysis. Using confidentiality mechanisms can protect it further.

a. Routing control mechanism

This mechanism works by pre-arranging a secure route for data transmission, detection of unwanted data forbidden under security policy and re-routing such data or deny access of data to specific networks.

b. Notarization mechanism

This mechanism utilizes a third party notary service to validate the authenticity of the transaction. All involved parties trust the notary service. Integrity mechanisms such as digital signatures, encipherment etc. as directed by the notary is used.

2.2.3 Attack

According to X.800, in general, there are two different types of attacks:

a. Active attacks

These are attacks targeted at changing the state or operation of the system. Some active threats are:

i. Masquerade
   A false entity presents itself as the claimed entity. It usually consists of other malicious attacks such as replay or modification of data.

ii. Re-play
   Re-play occurs when the same message is repeatedly used by different entities for unauthorized access

iii. Modification of messages
Modification of message is an unauthorized alteration of a transmitted message, which gets changed on the way.

iv. Denial of Service
The process of suppression or generation of messages transmitted from source to destination in such a manner that it disrupts the services on which the system depends is called denial of service.

b. Passive attacks

X.800 defines passive attacks as a threat of unauthorized access to information without changing the state of the system (ITU-T, 1991).

If these threats are discovered on time, then they cannot cause problems.

i. Wire-tapping or eavesdropping
Wiretaps are used to perform passive observation of data on different channels used for transmission, which can cause a leak of information sensitive in nature such as financial and personal data.

ii. Traffic Analysis
The observation of flow of traffic to understand patterns and inference of data from such patterns can be damaging. Traffic analysis can cause threats to the cryptographic system as well which depends on specific trends or patterns.

2.3 Threats specific to Online banking / E-commerce

Online banking is a feature provided by almost every bank to provide its customers with essential services. With the advent of the Internet, like every other industry, banks also quickly to accept its advantages. Online banking quickly became an integral part of the banking industry as well. The reason behind such a strong response to online banking is because of the customers. Customers of the banking industry became the true owners of their bank accounts (Gounaris & Akamavi, 2005). However, with the increasing use of cellphones and the expansion of 3G and 4G wireless services, a new front on the battle for customer satisfaction is opened. Online banking is now moving towards mobile online banking.

These changes have brought significant effects in our personal and professional lives as well. But, like every coin has two sides, there are several problems as well. One of the biggest issues when dealing with online banking is security. In one study done in Singapore, Internet banking
users accepted it because it seemed convenient, simple and more compatible (Gerrard & Barton Cunningham, 2003). Banks are a large source of personal information. However, in a 2005 study in the United Kingdom (U.K.), the banks were found to provide insufficient security mechanisms for its customers (Sarel & Marmorstein, 2006). Banks have however tried several solutions to solve problems related to information security in different ways. Banks such as HSBC gave away antivirus software to its clients for free. Banks have also come up with some high tech solutions of confirmation of information by using a cellular network such as SMS or through email confirmations. They have used two-factor authentication techniques such as PIN code generators. Now, the trend is growing towards the use of multi-factor authentication techniques such as biometric devices, pattern based encryptions, image capture etc.

2.3.1 Phishing – Identity theft attacks

In one study (Karim et al., 2009), which was conducted in London, UK, suggested that online banking users are aware of these various threats. Therefore, 24% of these users suggested that the best way to deal with these threats is through awareness and advanced technology. Most of these users are aware of phishing, which was found as the preliminary source of online banking problems.

This particular problem of phishing is highlighted with the fact that this is one of the most employed techniques by fraudsters to obtain credentials from online banking customers to steal financial information (Ståhlberg, 2008). Some infamous malwares which are used for phishing attacks are ZeuS and Spyeye (Coogan, 2010).

Phishing works through the concept of deception and infection. They portray themselves as genuine messages or warnings, which deceive the user to believe that they are in genuine danger of some sort. Then they ask the user to download some executable file hidden under genuine software to infect the device and provide them with their credentials.

2.3.2 Man in the Middle – Channel Breaking attacks

Online banking under present conditions, utilize two-factor authentication as a solution to a number of security problems. Two-factor authentication depends on the concept of improbability to estimate the randomization process, where a random password is generated to a specific user. However, the user remains the static element in this transaction. The two-factor authentication
works very well for passive attacks such as wiretapping, dictionary attacks, password guessing. However, the change of attacks from passive to active attacks adds a different degree of vulnerability to use of two-factor authentication techniques such as OTP (One Time Password) or PGT (Password Generating Token). Man in the middle attack is an example of such an attack, where a fake website which has almost similar web address as the original address can be used to fool potential customers to provide their credentials. These credentials can be used to access the original website. The attacker can misuse these credentials in a variety of ways. The attacker depends on the user to complete the authentication process, including two-factor authentication such as SMS verification or OTP verification. After the completion of two-factor authentication, the attacker can start the attack. One way to mitigate issues of man in the middle attack is through the use of session ids. However, attackers have discovered a new way of attack called “Man in the Browser” discussed further.

### 2.3.3 Man in the Browser – Content Manipulation

MitB (Man in the Browser) attacks are possible through several techniques such as Browser Rootkit. This is a popular technique employed for the Man in the Browser attacks. This attack usually occurs between a user and the browser. The browser is compromised using various malicious browser extensions, which can alter functionalities within browsers. These extensions attach themselves to various parts within the browser including the installation files located locally within the hard drives. Man in the Browser attacks are more sophisticated in the sense that they give the user the impression of security, as everything seems normal for the users. MitB attacks employ browser extensions as these extensions have different privileged capabilities to enhance or alter services provided through operating systems. These attacks utilize browser helper objects, JavaScript, AJAX, Browser API and DOM (Document Object Models). The attacker can remain in control of different sessions and these connections can run on TLS/SSL connections as well.
2.4 Server Authentication

These above-mentioned techniques could apply for users as well as servers. Some scenarios for such an attack can be:

i. **Error while typing the domain names of the financial institution**

   An error while typing the domain names of the financial institution can be used as an opportunity by these phishing websites. Such an error can redirect the user to a cloned website.

ii. **Security of servers or websites**

   A lack of security for servers or websites can severely be misused by the phishing messages as well as hosting purposes of illegitimate websites.

iii. **Offline dictionary attacks:**

   A list of passwords can be used to attack in offline mode.

iv. **Online dictionary attacks:**

   An Online dictionary attack is the same as an offline dictionary attack, however, it varies in the context of verification of passwords in online mode to find out if it is right or not.

v. **Online identity impersonation:**

   An Online identity impersonation is one of the issues with current social media. Facebook, for example can be used to create a fake account by establishing a fake user. There is no particular possibility for Facebook to link the user to the corresponding user profile which means it cannot verify whether the user is as claimed. Anyone can upload a photo of someone else on Facebook and there are no means to verify if that photo is linked to the real user or not. Public Key Infrastructures and Pretty Good Privacy are in some sense solutions to provide a link between online entities to an individual. However, these solutions lag behind, in terms of issuing personal certificates to every individual. Certification Authorities cannot provide individual certificates to all users.

vi. **Compromise of Certification Authorities:**

   Certificates are a means of assurance that the information is true. Certificates are mostly distributed by various third party elements that are known as Certification Authorities (CAs). The validity of such certificates rely on a cryptographic watermark however, valid certificates can be used for phishing attacks. So the validity of certification does not always constitute in the valid website itself. So this creates vulnerability for CAs and their certificates as well.
Server Certification is widely done using WTCA (WebTrust for CA). This particular dependency upon CA creates a security assurance limited to the capabilities of the CA. Therefore; the server will only be as secure as its strongest CA.

**vii. Information variation at Servers:**

In order to manage information within the security procedure of an organization, the information is classified according to its sensitivity, confidentiality, and integrity. This information is then stored in a relevant server, which consists of the required security procedures for access. This is done for the availability of information.

Server authentication is becoming more troublesome as the Internet gets broader and the number of users increases rapidly. The most implemented process of authentication of online entities during online transactions is TLS (Transport Layer Security) previously known as SSL (Secure Socket Layer). TLS is responsible for the security of transaction and encryption of transaction related information. However, the password used for user authentication can be a point of vulnerability created by the use of phishing websites, which is one of the biggest threat for online banking users. Some of the other threats mentioned above, such as dictionary attacks, are also possible in such scenarios.

In early days, passwords, in general used to be simple and short but however, with present security needs, users are recommended by service providers to create long and complex passwords with alphanumeric characters, which makes them difficult to remember unless someone has an exceptional memory. Public Key Infrastructure (PKI) is another solution to such a problem; however using such an elaborate method of security has its set of limitations in terms of offline attacks on the system, where the system is limited to the user credential storage. User is limited to using a system which has the keys installed. Such a limitation makes it easier for attackers to focus their attacks on a limited number of systems.

A few years back, online banking relied heavily on the username password combination, however, with new threats, the use of one-time password along with username, password combination has increased (Nagaraju & Parthiban, 2015). Therefore, the trend is moving towards multi-factor authentication, where biometric authentication techniques are employed. Almost 52 percent of financial organizations in Asia use fingerprint security systems (Tripathi, 2011).


2.4.1 Types of Server Authentication

There are two popular cryptographic transport protocols called TLS and SSH which use RSA or DSA as primary authentication techniques.

For server authentication, two of the most important cryptographic transport protocols are:

1. TLS – Transport Layer Security

   In the case of a TLS encrypted with RSA, the server sends a TLS certificate and the public key to provide credentials for a handshake. An attack can occur at this point if the attacker is in possession of the server’s private key and the message containing valuable information including session-controlled information can be decrypted.

2. SSH – Secure Shell

   SSH consists of two different versions of handshake:
   
   a. SSH - 1

      In SSH - 1, the client can encrypt the session key using the public key of the server.
   
   b. SSH – 2

      In SSH - 2, a session key can be generated from Diffie-Hellman key exchange.

        In SSH - 2, a session key can be generated from Diffie-Hellman key exchange.

        Similar to TLS, an attack can occur when an eavesdropper with the private key of the server can decrypt the SSH - 1 messages. While SSH -2 provides a greater level of security using Diffie-Hellman, but it is still open to active attacks.

        SSH enables authentication of the server to a client using signature verification while the handshake takes place. The channel is encrypted however, the password is in plain text and if the attacker already has a private key of the server, then the password is compromised.


2.4.2 Domain Name System

DNS (Domain Name System) can be considered as one of the important factors for the success of the Internet in general. Let us consider us remembering all the Internet addresses by memorizing the digits, which would impossible. DNS is a convenient solution to this problem. DNS is a link between a human user and the server.
DNS is a hierarchical distributed system spread around the world (Wang, Hu, Li, & Yan, 2006). It consists of TLDs (Top Level Domains), which remain at the highest level followed by SLDs (Second Level Domains), which give names to domains. Various Internet applications, including emails and surfing the web, depend on DNS. The number of host records has grown from 20,000 in 1987 to 1,033,836,245 in July 2015 i.e. more than 1 billion (Internet Systems Consortium, 2015).

Although DNS has significantly increased in size to exponential size, there are security glitches in DNS. These errors are concerned with integrity and authenticity (Mockapetris, 2004). As described above, DNS is hierarchical in nature and there are a large number of domain names registered to different organizations including various people. The primary use of DNS is its ability to distinguish one website from another. ICANN (Internet Corporation for Assigned Names and Numbers) is an organization, which governs the policies related to DNS and implementations. Thus, it has established seven new generic top-level domains (gTLDs) (ICANN, 2015). An extensive study of DNS by Danzig et al. was performed on the DNS traffic on ISI (Information Sciences Institute) root name server in 1992 (Danzig, Obraczka, & Kumar, 1992). This study suggested an array of problems related to recursion loops, poor failure detection algorithms. Another study by Brownlee et al. suggested passive forms of attacks directed to F-root server such as queries being repeated to private address space, or invalid top-level domains, source port zero and requests used to update root servers (Jung, Sit, Balakrishnan, & Morris, 2002).

Root servers have a relatively higher level of security. However, there is an issue of authoritative servers. These servers are more vulnerable in terms of security with regards to their location, usage and performance issues.

**2.4.2.1 How does the DNS work?**

DNS is a distributed set of addresses. DNS is normally a database of these addresses, which consists of mappings from a domain to different IP addresses. Domain names are hierarchical in nature, for example, a domain name “A” and another label or domain “B” exists then “A.B” is considered a sub-domain of ‘A’. This sub-domain is considered as a part of the ‘A’ domain name space as well.
Mappings related to respective domains are hierarchical as well. NS (Name Server) provides mapping of each domain. Name servers of respective domain are identified by a DNS mapping of type NS from the domain name to the corresponding domain name of the name server. Mappings of any domain name are considered legitimate only if they are received from a name server of that domain or its parent domain.

Clients, on the other hand, use resolvers in order to find resource records for a certain domain. These resolvers interact or query the name servers to locate resource records. The name server then responds with a lookup and finds the corresponding resource records, or if it does not find a match, then a corresponding message is retrieved. Resolvers cache the DNS responses as per the time to live (TTL) field of response, which is normally represented in t seconds.

**2.4.2.2 Different types of DNS-based attacks:**

1. **DDoS (Distributed Denial of Service) attacks:**
   
   Denial of Service attack is targeted normally towards a single victim, where the attacker can take up all the resources of the network. DDoS attacks are normally accompanied by fake IP (Internet Protocols) addresses. This makes the attack harder to trace. One particular method of this kind of attack is the use of flood attack. In an SYN flood attack, the TCP (Transmission Control Protocol) is involved in a half connection. A random source address is created for TCP segment and TCP SYN flag is set for each TCP and requested to the server. The server then receives TCP SYN segments; the server then stores all the state information for each TCP connection by creating a TCB (Transmission Control Block). After this, the server sends a TCP SYN + ACK segment to the source address and waits for the ACK segment. However, instead of sending ACK segment, the attacker sends multiple SYN segments to the server. This will eventually create a large amount of TCB and creates a denial of service.

2. **DNS hijacking:**

   The accuracy of DNS is one of the preliminary factors to affect the proficient use of the web. DNS hijacking is a real threat to allow users to accurately navigate the web and this inaccuracy leads to security issues. One such example is cache spoofing, where an attacker can insert the false address in a cache of DNS within a server or even a single browser on a client system. The most important we need to understand is that because of logistical issues, we depend on multiple DNS servers to surf the Internet. If we are looking for particular website, we will receive multiple suggestions from servers. DNS spoofing which is another name used for DNS hijacking is also transferrable through this process which makes it, even more, dangerous. An attacker can come in control of the DNS server and this can severely affect security.
After DNS hijacking, legitimate users of legitimate websites might be redirected unknowingly into a fake website. Among several mechanisms used to tackle the issue of DNS hijacking, challenge-response mechanism seems to be employed more. However, this mechanism becomes ineffective against man in the middle attacks.

2.4.2.3 Countermeasures against DNS attacks:

a. SYN flooding attacks

As a countermeasure to SYN flooding attacks, an approach to reduction of the TCP connection to its hash is done. This approach of reduction, reduces the size, however, it does not have the intended effect since the size is not reduced to zero. The effect is thus limited. SYN cookies embed the data instead of storing the hash for any TCP SYN + ACK segment and forwards it to the attacker. This technique has the intended effect of zero state.

Although the intended effect is achieved, there are several drawbacks. One of the side effects of using this solution is that the ACK segment including the legitimate connection is received and the TCB data is regenerated which can be instantiated. TCB data contains 32-bit sequence number field, so some TCP options might not fit. TCP SYN + ACKs might not be retransmitted in response to this and the server cannot detect legitimate client ACK segment.

2.5 DNSSEC (Domain Name System Security Extensions):

It is considered to be one of the most effective means to protection against threats towards DNS. It achieves the protection using digital signatures. The DNSSEC ensures the integrity of response through cryptography. The reason to use DNSSEC to sign DNS records is to assure that the records are valid to any authenticating DNS resolver.

DNSSEC provides authority of origin, data integrity and authenticated denial of existence. It is proven effective against one particular type of DNS attack called DNS spoofing. DNSSEC can secure an entire zone through zone signing. This signing of a zone with DNSSEC creates validation to a zone without affecting any mechanism for DNS response and query.
When a client sends a request or a query towards a DNS server for the DNSSEC signed zone, the DNS server sends the DNSSEC records and an attempt to validate response with the records.

A DNSKEY resource record is used by a recursive DNS server to validate responses from an authoritative DNS server. This validation occurs through decryption of digital signatures, which are contained in DNSSEC-related resource records. The hash values obtained from them are computed and compared. The result of the hash values must match in order for the request to be materialized into specific data as requested, otherwise, no further response is provided except for a service failure message.

Table 1: Resource record types for DNSSEC

<table>
<thead>
<tr>
<th>Resource record Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RRSIG</strong></td>
<td>RRSIG (Resource Record Signature) records contain the signatures generated with DNSSEC. A query issued by a resolver for a name is returned with a response containing RRSIG</td>
</tr>
<tr>
<td><strong>NSEC</strong></td>
<td>NSEC (Next Secure) records prove non-existence of DNS names. It is targeted towards spoofing attacks which are targeted to mislead clients to believe that the DNS that they search do not exist.</td>
</tr>
<tr>
<td><strong>NSEC 3</strong></td>
<td>NSEC 3(Next Secure) is an update over NSEC which adds an additional benefit of avoiding recurrent NSEC queries. This phenomenon of recurrent NSEC queries called “Zone Walking” is prevented with NSEC 3. However, either NSEC or NSEC 3 can be used but not simultaneously.</td>
</tr>
<tr>
<td><strong>NSEC 3 PARAM</strong></td>
<td>NSEC3PARAM (Next Secure 3 Parameter) determines the NSEC 3 records to be included in the responses to non-existent DNS names.</td>
</tr>
<tr>
<td><strong>DNS key</strong></td>
<td>DNSKEY record is used for validation process by DNS server</td>
</tr>
</tbody>
</table>
2.5.1 How does DNSSEC work?

![Diagram of DNSSEC validation process](image)

**Table 2: Steps in a DNS query and response with optional DNSSEC data**

<table>
<thead>
<tr>
<th>Step</th>
<th>Query-response</th>
<th>Optional DNSSEC data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A DNS client sends a DNS query to a recursive DNS server.</td>
<td>The DNS client can indicate that it is DNSSEC-aware (DO=1).</td>
</tr>
<tr>
<td>2</td>
<td>The recursive DNS server sends a DNS query to the root and top-level domain (TLD) DNS servers.</td>
<td>The recursive DNS server can indicate that it is DNSSEC-aware (DO=1).</td>
</tr>
</tbody>
</table>

---


which consists of a public cryptographic key stored in a resource record which is used to verify signature.

**DS** DS (Delegation Signer) creates authentication chains through a secure delegation especially to child zones.
The root and TLD servers return a DNS response to the recursive DNS server providing the IP address of the authoritative DNS server for the zone. Authoritative servers for the parent zone can indicate that the child zone is signed using DNSSEC and include a secure delegation (DS record).

The recursive DNS server sends a DNS query to the authoritative DNS server for the zone. The recursive DNS server can indicate that it is DNSSEC-aware (DO=1) and capable of validating signed resource records (CD=1) to be sent in the response.

The authoritative DNS server returns a DNS response to the recursive DNS server, providing the resource record data. The authoritative DNS server can include DNSSEC signatures in the form of RRSIG records in the DNS response, for use in validation.

The recursive DNS server returns a DNS response to the DNS client, providing the resource record data. The recursive DNS server can indicate whether or not the DNS response was validated (AD=1) using DNSSEC.

### 2.5.2 Other techniques for countermeasure

#### 2.5.2.1 Ingress Filtering

It is a means by which service providers can trace the source networks for any relevant traffic. This countermeasure is especially targeted towards spoofing attacks, where the source IP address is used through a proxy or a spoofed IP address. Distributed Denial of Service (DDoS) attacks can also be prevented by using ingress filtering techniques. It can trace the source networks, and also protects itself against spoofed access to networking equipment. There are five different ways of implementing ingress filtering:

a. **Ingress Access Lists**

   It crosschecks the source address of every message transmitted over a network against a list of acceptable prefixes and if the packets do not match the filter they are removed.

b. **Strict Reverse Path Forwarding**

   It is similar to the concept of using access lists, however, in this case, the access lists, which are used, are of dynamic nature to avoid any issues with duplication using FIB (Forwarding Information Base).

c. **Feasible Reverse Path Forwarding**
It is an extension of Strict Reverse Path Forwarding, including the use of FIB however; it looks up all possible routes invoked with routing protocol specific methods. It is especially targeted towards asymmetric routing or multi-homing of the network.

d. Loose Reverse Path Forwarding
   It is a type of a “route presence check”. It has its similarities to strict RPF, however, it differs in the fact that it checks only if the route exists or not.

e. Loose Reverse Path Forwarding ignoring default routes
   It is like loose RPF except it does not care for default routes. The router searches the source address in the route table and the packet is preserved if the route is discovered. It varies in the essence as the default routes are excluded.

### 2.5.2.2 Unicast Reverse Path Forwarding (RPF):

The priority for any router is to forward the IP packets which it might receive. For this matter, it really does not care about the source IP address, but only cares about the destination IP address. However, this gives an immense opportunity for attackers to attack this particular vulnerability. Therefore, Unicast RPF is employed as a security feature, which can prevent IP spoofing attacks. It consists of a routing table for the source IP address and checks for the IP packet, which it received to verify with the table. If the packet does not match then it is discarded.

Unicast RPF works with two modes:

a. **Strict Mode**
   It means that the router will check for two particular agendas:

   1. It searches for the matching entry for the source IP packet in the routing table.
   2. It searches for the interface to reach the source

   The IP packet is tested for both of these agendas and if the IP packet passes both of them then it is accepted, otherwise, it will be discarded. It is most suitable for IGP routing protocols, which use the shortest path to the source of IP packets concept in which:

   Interface to reach the source = Interface from where the packets originate

b. **Loose Mode**
   Unlike the strict mode, in loose mode, only one agenda is checked:
   1. It searches for the matching entry for the source IP packet in the routing table
The IP packet must pass only this particular agenda, and then it is allowed. It is mostly useful when multiple ISPs are involved and asymmetric routing is chosen. However, one disadvantage is the null0 interface related sources are discarded.

2.5.2.3 Sender Policy Framework (SPF) Protocol:

SPF protocol works on the concept of proof of origin and verification method. It basically checks for integrity. SPF is basically a DNS record. The mail server, which is in-charge of the incoming mails, crosschecks the domain of their email address against the list of authorized hosts to send messages of the domain based upon SPF. This process then makes a decision upon whether the email passes the criteria or not, and if it does it is allowed, otherwise it is blocked. SPF protocols are simple enough to use as a user needs to have access to a DNS server and modify a set of parameter to control the email server responses to the email received with SPF and validation.

One of the biggest benefits is the prevention of SPAM and forged emails. Forged emails from unauthorized sources, that claim servers that are enabled with SPF verification check false identities. The IP addresses of servers, which generate spams, can be blacklisted using SPF. Therefore, traffic from such servers will be filtered out.

Specific SPF records:

There remains a divide between a pure SPF and TXT record with SPF syntax, where specific SPF varies in terms of TXT record. While most of the modern servers can easily interpret specific SPF, some servers still cannot interpret them.

In the case of modern servers, the identity of the message is verified using both records one after the other. First the specific SPF is checked followed by TXT. SPF, which is based upon TXT record in the DNS server, is normally located in the outgoing mail server. The TXT record consists of a list of legitimate IP addresses and host names for a particular domain from which the mail might originate. The incoming mail server can then verify against the TXT record.

2.6 Mobile Banking
In 2015, The GSMA Intelligence (GSMA, 2015), reported in the report “The Mobile Economy”, that there are almost 3.6 billion mobile phone users all around the world which is almost half of the world’s population. This reflects the huge increase in users of some sort of mobile phone in the world. However, in developed nations this increase is nominal since the new penetration rate has reached close to saturation. Similarly, the report also suggests that the penetration rate for the adoption of smartphones will also reach 70 to 80 percent ceiling, which is normally considered to be the plateau.

Smartphones use in the developed nation is at its peak; however this increase is mostly accredited to the young users of the smartphones. One of the studies suggested that the reason behind the popularity of mobile phones is the ease of access to information and constant connectivity (C. Ling, Hwang, & Salvendy, 2006). There is a difference between the purposes of use of mobile technologies for different age groups. Younger users are concerned with social interaction aspects, while the users who are adult or elderly, the purpose varies from job-related issues to security, safety and personal independence (Abascal & Civit, 2001).

The use of smartphone and mobile technologies, in general, depends upon motivation or purpose of use. This can be either intrinsic or extrinsic motivations and it can also vary in terms of importance, either utilitarian or hedonic (Conci, Pianesi, & Zancanaro, 2009). There are several other factors, which are associated with the use of mobile phones, which applies to all users irrespective of age or gender. These factors are accessibility to information, display characteristics, arranging appointments, and safety (R. Ling & Haddon, 2003). One study suggested the use of mobile phones by older people in Finland was highly influenced by security and communication factors (Oksman, 2010). Another study suggested that the purpose of use of mobile phones by the elderly was highly motivated by social integration and independent living (Kurniawan, 2007).

The sense of security obtained from the ownership of a mobile device is directly correlated to the omnipresent nature of communication medium to contact loved ones at points of distress. However, another added benefit is the ability to remain independent. This increase in the use of secure mobile devices is opening up opportunities for online mobile banking, mobile payment etc. The convenience of doing transactions anywhere and anytime and the increasing level of
security on mobile devices are two important factors for this high increase in mobile banking technologies.

2.6.1 Mobile Transaction Procedure

A mobile transaction consists of three entities:
1. A User
2. A Device (mobile)
3. A Transaction operator (Bank, mobile operator, or both)

For security of mobile transactions there are three specific processes:
1. Identification
2. Authentication
3. Secure performance

A mobile user is normally identified in a cellular network through the unique phone number and PIN. The user in a mobile banking environment can be identified by three procedures:
   i. What the user has
   ii. What the user is
   iii. What the user knows

This means the user can be identified with the device that he possesses or through biometric means of different means ranging from fingerprints, vocal analysis etc. A mobile provider can authenticate the requests from the users through the subscriber information or through various specialized mechanisms such as digital signatures or security protocols such as WTLS (Wireless Transport Layer Security Specification (Herzberg, 2003). The transaction process consists of a transaction operator; secure payment protocols such as SET (Secure Electronic Transactions) or iKP (Internet Keyed Payments). The mobile transaction operator can use mobile gateways to support different communication and access control mechanisms (Herzberg, 2003).

When a transaction request is sent, the device using a private key, which remains unknown to the provider, must sign it digitally. The mutual agreement between a user and the operator sets a public key based upon DSA or RSA. The user must be able to view each individual transaction details for every request using markup languages for signing documents, which are pre-verified with a trusted authority through certificate or signatures.
User authentication is another important aspect. A mobile system is only as strong as its weakest link, which is, of course, a user. With the new smartphones, there are various possibilities to infiltrate devices especially those with open source operating systems.

Another important entity involved in online transactions is the merchant. The merchant is same as on the online shopping terms. This merchant account receives payments from a third party such as a financial institution. These transactions normally take place with an establishing a secure communication channel, followed by identification of customer to merchant through a subscriber identification or secure signature from customer’s device, which is a guarantee of payment.

The security of such a transaction is maintained through a secure communication channel. The user authorizes the transaction request, the payment transaction request from the user’s mobile device is matched with the respective offer, then the payment request contains state information within a cookie or a URL parameter containing MAC (Message Authentication Code) which ensures authenticity through a key which is known only to the operator.

After this payment is made, the operator confirms payment to the merchant through a signed authenticated message. The merchant then replies back with a receipt of the confirmation of the transaction.

### 2.6.2 SMS Banking

SMS banking was initially targeted towards the transmission of non-confidential information through GSM (Global System for Mobile communication) network. The SMS message is quite vulnerable as the messages remain in plaintext and an algorithm called A5 does the encryption. The encryption is done during the transmission process; therefore end-to-end encryption algorithm is not available.

This process has several vulnerabilities, first of which is the sending of a request from the user to the banking server using USSD (Unstructured Supplementary Services Data) string containing the user’s PIN code. This sensitive information is sent on the GSM network in plain text. The second vulnerability is the encryption algorithm A5, which is not considered to be secure.

Different versions of A5 are used for encryption, A5/1 is the standard used in North America and Europe and A5/3 is used by 3G services under UMTS (Universal Mobile Telecommunications
3G services using A5/3 are considered to be secure, however, 3G/UMTS and GSM operate in different frequencies. 3G technologies include two factors:

1. Encryption on the air interface
2. Mutual authentication between the user and the network using HLR (Home Location Register) and USIM (Universal Subscriber Identity Module).

The UMTS frequency can be blocked funneling the user to use GSM. Therefore, the vulnerability still remains in some forms as DCH (Dedicated Channel) starvation attack. DCH is a finite resource divided among the subscribers. When multiple devices connect to a network for DCH services, they soon get depleted and DoS (Denial of Service) attack is instigated.

### 2.6.3 USSD Banking

USSD (Unstructured Supplementary Services Data) consists of two different modes:

i. USSD 1

ii. USSD 2

USSD 1 allows one-way communication to a network while USSD 2 allows for two-way communication between a user and network. End-to-end transaction security is provided throughout the communication layer with a subversion of the identity of the subscriber. However, within the communication layer, the data is not encrypted therefore, the security can be compromised if the encryption is broken.

The sensitive data regarding customers is encrypted and stored on a server in USSD2. The handset only consists of authentication information of customers such as PIN (Personal Identification Number) and banking instructions and the user does not need to provide other credentials such as account number details. However, this system has similar security threat as an ATM card, if the PIN number of the ATM card is known then anyone can use it. Similarly, if the mobile device is stolen then anyone who possesses the device has the device, the SIM card, and the authentication information.

### 2.6.4 Mobile Banking in GPRS

The early stage of Mobile banking made use of WAP (Wireless Application Protocol) where a user can use the banking service by simply using the URL (Uniform Resource Locator). The
WAP enabled mobile banking systems were considered to be secure. However, there is no end to end encryption, between the client and the Gateway, and also between the Gateway and the Server (Narendiran, Rabara, & Rajendran, 2008).

GPRS (General Packet Radio System) also known as 2.5G is another technology used for data transmission over mobile networks; however, it is inefficient for continuous data transmission. The data transmission services under GPRS are packet-based. GPRS is based upon the penetration rate of GSM. There are 4 major types of data, which have to be secured in a GSM/GPRS network:

a. User Data: This is voice or non-voice data sent or received by users.

b. Charging Information: Information used to bill for non-voice services collected from SGSN (Serving GPRS Support Node) and GGSN (Gateway GPRS Support Node).

c. Subscribe Information: Information about customers recorded in HLR (Home Location Register) and VLR (Visitor Location Register).

d. Technical information: Information about GSM/GPRS network architecture and configuration.
Attackers can target any one of these four data types. One particular form of attack is the DoS, which targets GGSN by overloading it with requests for service. This DoS attack causes prevention of other mobile stations to gain access to the network.

GSM and GPRS provide three layers of security:

1. **Anonymity**

   TMSI (Temporary Mobile Subscribe Identities) are used to provide anonymity to subscribers on the cellular network using IMSI (International Mobile Subscriber Identity). The IMSI contains personal subscriber number, home network name and country code of subscription number.

2. **Authentication**

---

In the case of GPRS, authentication is handled by SGSN, through a randomly generated 128-bit number. This number is sent to the mobile station, which generates a 32-bit response using its private authentication key. This key remains unique to the subscriber implemented through SIM (Subscriber Identity Module) and A3 algorithm.

3. User Data Protection

To provide security against interception and eavesdropping, encryption methods are employed. The random 128-bit number generated in the authentication process and the private key from HLR is combined with the A8 algorithm to produce an encryption key. Therefore, the data transmission process between the GPRS network and mobile station is encrypted using GPRS-A5.
Chapter 3

Research Method

The focus of this project is to further apply the OffPAD authentication concept to incorporate this technology into platform independent mobile platforms such as cellphones and other handheld devices. In this particular field, there have been prior studies. The research method consists of five elements.

- The first stage was therefore to do a literature study, where we came across different types of articles regarding authentication, online banking, mobile technologies, threats relevant for online banking environments etc. Articles that were considered relevant to this thesis have been included in Chapters 1 and 2 respectively. As this builds on both the Lucidman project\(^4\), and on the OffPAD project, we have also included relevant articles related to those projects.

- An important element of the research method was to identify limitations, weaknesses and needs in existing solutions. Some of these previous technologies and their limitations are described further in Sections 1.4, 2.1, 2.2, 2.4 and 2.5. Some legacy mobile banking technologies such as SMS-banking, WAP-based banking, GPRS-based banking, 3G-based banking are also discussed further in Section 2.6 to give an overview of the evolution of mobile banking environments.

- The next element in the research method was to design new solutions to compensate for the existing limitations in previous solutions.

- The subsequent element in the research method was to implement a prototype of the new design as a proof of concept. This prototype is targeted towards authentication of data using the concept from OffPAD project. However, it differs in the sense that it does not require a dedicated device, which means it uses the concept of OffPAD as a mobile application.

- The last element in the research method was to test and evaluate the new design and the implemented prototype. There are still some limitations to this prototype, which we have

---

\(^4\) http://www.lucidman.org/
mentioned in Chapter 8. These limitations can generally be viewed from two standpoints: time constraints and scope of this thesis.
Chapter 4

System Overview

In this chapter, we will discuss the overall process of development of the proposed authentication system for online banking and its prospect for integrating with OffPAD. First of all, the requirements are listed; also the assumptions, in which the projects are based, are listed, after which the technical description of the project is explained. This chapter ends with its future use with the OffPAD project itself.

4.1 Requirement study

Since there are three main entities involved in the system, each of them has some requirements, to be able to be used in the proposed system. The requirements for each entity are listed below.

4.1.1 OffPAD device
1. Android device with API level more than 11
2. Bluetooth connectivity
3. On-screen keyboard

4.1.2 Client / Browser System
1. Java Runtime Environment (JRE)
2. Tomcat Server
3. MySQL Server
4. JavaScript enabled

4.1.3 Server System
1. Java Runtime Environment (JRE)
2. Tomcat Server

Above mentioned requirements are specifically for running the proposed system, based on the development environment.
4.2 Assumptions

Since the primary focus of this proposed solution is data authentication, therefore, there are assumptions made on the side of encryption management system. The assumptions can be basically categorized into two groups. First, assumptions made in the initial decision on the cryptographic protocols between the OffPAD and client/browser. And secondly, we assume that the communication channel between the client/browser and server/bank is secured using TLS/SSL.

As per the first assumption, the client and OffPAD device will initially decide to follow any of the three cases presented below.

4.2.1 Shared Key

Shared key encryption consists of one key, which is able to encrypt and decrypt messages. Therefore, both the parties involved in the process of sending and receiving must possess the same key. This means the process is dependent on keeping the key secret from other intentional intruders. However, this creates vulnerability in the system, as anyone in the possession of such key is able to decrypt messages.

HMAC is a cryptographic function using a secret key. It is a hash function.

Notations:

- H: Hash function
- B: Byte-length
- K: Secret Key
- L: Byte-length of Hash

Parameters:

1. K is restricted to the length of B.
2. K longer than B is hashed using H.
3. K is not supposed to be shorter than L, for security reasons.
4.2.2 OffPAD as a dedicated device

OffPAD can be a dedicated device distributed by the concerned bank to their customers. This device will have the users public key pre-installed in it and will be unique as per the user. For the communication with the client browser, this device will have the Bluetooth capability including pairing and communicating with the client. This device can have the encryption mechanisms pre-installed in it for secure communication between the client browser and itself.

Using this device the user needs to provide only destination account number and amount, which are both numeric, so it will have a display and numeric keypad in it. It would be similar to the devices that are distributed by the Nordea banks to their customer.

---

5 Retrieved from: http://2.bp.blogspot.com/-6ZBjHc3kyx8/Uhh8oRbBzcI/AAAAAAAAFEA/FV8MOuobeTk/s1600/IC168364.gif, 18/10/2015
Such devices are used for one-time code purposes and, are distributed by the bank to their users. They provide a one-time code, which is used with your social security number and password for accessing the bank services. It is most prevalent in online banking environments. There are different types of one-time password generators; some of them are PIN protected while some are not. It provides an additional level of security, however, users must carry multiple devices corresponding to different institutions because different banks will use a variety of such devices.

---

7 Retrieved from: http://www.safenet-inc.com/uploadedImages/Products/Enterprise_Data_Protection/Multi-Factor_Authentication/OTP_Authenticators/eT_PASS.jpg?n=2413, 19/10/2015
8 Retrieved from: http://www.cib.hu/%5Eupload/Kepek/token_small, 19/10/2015
4.2.3 OffPAD solution as an application

As for the third possibility, OffPAD solution can be used as an application for mobile devices. This is the scenario that our proposed project is based on. The android application can be installed in any android devices with the requirements listed above in Section 4.1.1. In this scenario, the users need to download the application from the respective download store and use it with the client browser for online transactions. The public key used for encrypting the data comes within the package of the mobile application. And there would be a login mechanism for user authentication.

There are also some assumptions that are made for PKI factors for various entities of our system. Assumptions are as following:

1. We assume the reliability and ability of a CA (certification authority).
2. We assume certificates are provided to clients and all of them possess such certificates in their browsers including servers.
3. We assume all relevant protocols such as handshake protocol are followed.
4. We assume that CA has the ability to revoke certificates.

We use TLS/SSL as the transmission mechanism between client and server. However, the details about the transmission between client and server are not a part of this thesis. The thesis is relevant to the context of data authentication mechanism generation only. Similarly, we assume that the user will be informed about any threat, which is discovered using our solution through different means such as SMS, phone or email. This communication of information regarding threat sent from the server to the user is not the focus of this study therefore, we shall assume on this context.
4.3 Prototype design and development

As explained above, the primary focus of this system is data authentication. So that is also kept in mind when the prototype and designs were made for this system. With the focus on the user and the frequency of online transactions on a daily basis, the system needs to be less time consuming and user-friendly. An extra layer of security needs to be added without increasing the overall transaction period.

With this in mind, I have tried to keep the design as simple as possible with the use of minimum unique fields to fill out for the user.

4.3.1 Device

The original OffPAD device is based on the android OS. For the purpose of implementing and testing my application, I have used Samsung Galaxy s5 as OffPAD device to insert the details of destination account number and amount to be transferred into the client/browser securely. An apple device, MacBook Pro, is used as a client device, which opens the java servlet on its browser as the banks website. The OffPAD device and client device are connected via Bluetooth channel. The Bluetooth pairing between these devices is done beforehand the transaction is initiated.

The existing online banking system requires a small device, which generates a one-time code for the online transaction. Sometimes when I forget to carry the device, I am unable to perform any transaction. However, OffPAD technology is independent of any code-generating device as the web app can be downloaded anytime with any android device and synced with the browser for carrying out any transaction.

4.3.2 Implementation

The implementation of this project is basically focused on the data authentication. The overall time for the transaction to be verified in this project is not greater than the time required in existing online banking process. The online banking system in Norway generally requires a one-time code-generating device. The code from the device is entered into the browser, which provides user authentication. Using the social security number, the one-time code generated from
the device and a personal password, a user is able to access their bank account in their browser. Then, they have to furthermore verify the transaction using the one-time code and password. In the context of this project, it is focusing more on data authentication. User authentication can, however, be provided assuming that we use a dedicated OffPAD device provided by the bank. In such device, the user credentials are pre-stored and is unique for every user. While, if we assume that the user uses OffPAD solution as a mobile application, then the user authentication would be done by entering the user credentials in the application. Also, as we know data is the main motive behind any attack, it needs to be securely transferred between the entities.

The implementation process can be briefly explained as follows:
First of all, the user opens the browser and logs into the bank website. The user then initiates the transfer function by selecting the transfer option from the menu in the browser. The client browser then refers to the OffPAD device to input transaction details. The user then opens the OffPAD app, selects transfer option and fills in the necessary information. The user then clicks the submit button and forwards the details to the browser. These fields are injected into the client browser automatically. Before injecting the destination account and amount in the client browser, the OffPAD app generates a hash value (hash_app) from the combination of the destination account and the amount fields. This hash value is generated using the SHA-256 hash encoding technique.

\[
hash_{app} = SHA-256 \text{ hash (destination account+ amount)}
\]

This hash value is transmitted along with other fields from the device to the client browser. The transferring of data from the device to client browser is via the Bluetooth channel. Bluetooth channel can be eavesdropped, and different attacks can be made to the Bluetooth channel using different techniques like MAC spoofing, which is a type of passive attack that takes place at the instance of the pairing of devices just before encryption is started through the use of special hardware devices. After such an attack, the attacker can remain in control of data transmission process leading to failure in connections of legitimate connections, capture, and manipulation of data. To counter such issues, our solution provides stronger encryption for Bluetooth transmission of data. The data that is being sent by device gets encrypted using the public key of
the client browser. And when the data is received, the client browser decrypts it and injects the destination account and amount in the respective field in the transfer form.

\[ data = RSAAsymmetricEncryption(destination\ account + amount + hash) \]

The other fields like KID (Kundeidentifikasjon) and remarks are filled in by the user and submit the form. The transfer forms data are transmitted along with the hash_app as a hidden field. This field is not visible to the users. The communication between the client browser and the server is through the TLS/SSL.

The server then receives the transfer request sent in by the client browser. Also with all the necessary fields, it also receives the hash_app from the client browser. The server then generates the hash value (hash_server) from the combination of received destination account and amount fields from the client browser. This value is also generated using the same mechanism that the OffPAD device uses for hash generation.

\[ hash_server = SHA-356\ hash(\ received\ destination\ account + received\ amount) \]

The server then compares the two hash values, i.e. hash_app with hash_server, if they matches, then the transaction request is accepted and necessary steps are carried out. If the values mismatch, then the server will know that there has been an alteration of values on the client side. So the transaction request is denied and the user is notified by SMS, phone or by email, about the inconsistency of the transaction.

The mismatch of the two hash values occurs only when there is an alteration of data in the client side. The client side might be infected with Trojans, malware and other kind of malicious software. These might alter the values in the transfer form fields. This alteration may or may not be known to the users. So any alteration that occurred in the client side by any means will lead to hash mismatch, by which the server would infer about the inconsistency in the transaction. This solution works even though the client is infected with malicious software. This way, the data is authenticated through the web in any online transaction.
The whole process can be broken down into vital steps as follows:

1. The process starts with the installation of the mobile application in android phones. After the installation, it is paired with the client system via Bluetooth. The mobile application is then ready to use with the web browser. The user can turn the Bluetooth on/off according to their need using the application. A Bluetooth activation button is also available in the application itself for easy access. There needs to be a Bluetooth connection between the OffPAD device and the client system for completing an online transaction.

2. The user logs in into the client browser using the credentials and initiates the transfer function by clicking on the transfer operation.

3. The web browser then directs the user to the OffPAD device for entering the details of the transaction namely the destination account number and the amount to be transferred.

4. After entering the destination account number and amount, the user then clicks on transfer button. At the same time, the secure device generates a hash value from these two inputs and sends the generated hash value to the web browser along with other fields via Bluetooth.

5. The web browser then directs the user to a web applet where the destination account number and the amount get injected from the OffPAD device. Other fields like KID (Kundeidentifikasjon) and remarks need to be entered by the user in the web browser itself.

6. The user then clicks the transfer button. The web browser forwards the hash value generated by the OffPAD device to the server. After receiving the transaction details, the server also generates its hash value from the received destination account number and the amount.

7. The two hash values are then compared. If they match, the transaction is carried out successfully.

8. If there is a mismatch between the two hash values, the transaction is rejected and the user is notified of the inconsistency.
Figure 10: The basic working diagram of proposed system.

1. The customer is notified.
2. The transaction is halted, and process 
   is valid.
3. If matches, the transaction is
   complete.
4. The hash server
   (received destination + received account) 
   generates SHA 256 hash from
   5. Transfers destination account
   and amount with hash + app
   6. Instructs the received value from app to
   7. Completes the transfer function
   8. Logs on to the bank website
   9. Sends the value to the bank website
   10. Generates hash, app and hash server.
In Figure 10, there are four components:

1. User: U
2. Client/Browser: C
3. OffPAD: O
4. Server/Bank: S

Steps described on this diagram:

1. U initiates User authentication by log in credentials.
2. U initiates transfer functionality through C.
3. U enters destination information and amount in O.
4. The connection between O and C is established through Bluetooth.
6. C sends transfer information to S using TLS.
7. S receives information, generates a hash key, which consists of destination information and amount.
8. S compares the two hash keys and either accepts or declines the transaction.

This is a brief overview of the diagram shown above. Based on this structure, the practical implementation details are explained further in this section:

### 4.3.3 Application Prototype

The application is developed in java on the android sdk platform. The functionality is kept fairly simple for better user interaction. The different screenshots of the application are presented below.
Figure 11 shows the initial screen of the user app, which is shown once the user opens the application.

Figure 12 shows the main menu page of the application. It consists of four options:

1. Turn Bluetooth On
2. Transfer
3. History
4. About

Users can select any one of these options at a time and view information or initiate transactions.
Figure 13 shows the application accessing the permission to access the Bluetooth channel of the device. The user can directly turn on the Bluetooth from the application itself.

Figure 14 shows the status of the Bluetooth. The status information serves its purpose of informing the user about an open/closed channel.

Figure 15 shows the transfer information page. In this page, the user can insert the information relevant to particular transfer, such as account number of the destination as well as the amount. The user clicks on the “Send to Browser” button to commit the transfer request, this is then injected into the client browser automatically.

Figure 16 shows the information received by the user when the successful transaction is completed. The user acknowledges the information by accepting it through the “OK” button.
Figure 15: Transfer Information

Figure 16: Transfer success information

Figure 17: Transfer failure

Figure 18 Transfer history in ascending order of date
Figure 17 shows the transfer failure information shown to the user. The user again acknowledges this information by pressing the “OK” button. The transfer failure can occur in various reasons, such as, if the client has not turned on Bluetooth or if the client is not listening to the Bluetooth port for receiving data.

Figure 18 shows the transactions information page, which shows a list of transactions, with its details and status. It shows the destination account numbers, amounts including corresponding date-time stamps and the status, whether it was a success or failed to send the data to the client browser. All of such information is arranged in descending order according to the time they occur.

```
<table>
<thead>
<tr>
<th>Id</th>
<th>Date</th>
<th>Sent To</th>
<th>Amount</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 Jan 2016 11:53:57</td>
<td>6030.2548.363</td>
<td>120000</td>
<td>success</td>
</tr>
<tr>
<td>2</td>
<td>10 Jan 2016 11:59:12</td>
<td>222</td>
<td>111</td>
<td>success</td>
</tr>
<tr>
<td>3</td>
<td>10 Jan 2016 11:59:33</td>
<td>2525</td>
<td>1212</td>
<td>success</td>
</tr>
<tr>
<td>4</td>
<td>10 Jan 2016 15:24:42</td>
<td>55500000</td>
<td>25000</td>
<td>success</td>
</tr>
<tr>
<td>5</td>
<td>10 Jan 2016 12:29:35</td>
<td>55225522</td>
<td>1212500</td>
<td>success</td>
</tr>
<tr>
<td>6</td>
<td>10 Jan 2016 12:28:49</td>
<td>2525</td>
<td>111</td>
<td>success</td>
</tr>
<tr>
<td>7</td>
<td>11 Jan 2016 13:12:46</td>
<td>6030.12.1414</td>
<td>25000</td>
<td>success</td>
</tr>
<tr>
<td>8</td>
<td>11 Jan 2016 13:14:13</td>
<td>222222</td>
<td>111111</td>
<td>success</td>
</tr>
<tr>
<td>9</td>
<td>11 Jan 2016 13:15:07</td>
<td>333333</td>
<td>55555</td>
<td>success</td>
</tr>
<tr>
<td>10</td>
<td>11 Jan 2016 13:19:26</td>
<td>3332222</td>
<td>1111220</td>
<td>success</td>
</tr>
<tr>
<td>12</td>
<td>11 Jan 2016 13:34:43</td>
<td>6039.12.1414</td>
<td>25000</td>
<td>success</td>
</tr>
<tr>
<td>13</td>
<td>11 Jan 2016 13:39:45</td>
<td>2525</td>
<td>12100</td>
<td>success</td>
</tr>
<tr>
<td>14</td>
<td>12 Jan 2016 11:59:58</td>
<td>2522252</td>
<td>12000</td>
<td>failed</td>
</tr>
<tr>
<td>15</td>
<td>12 Jan 2016 17:00:30</td>
<td>2525123456</td>
<td>123456</td>
<td>failed</td>
</tr>
</tbody>
</table>
```

Figure 19: Transaction history in descending order of date

Figure 20: About information

Figure 19 shows all the transaction information arranged in ascending order of the date. The user can toggle the sorting order by pressing the date header, or the icon beside the header.

Figure 20 shows the information about the application to the user. This information is presented when the user selects “About” option from the main menu.
4.3.4 Client Server Prototype

The client and server are both running a website that is hosted in tomcat server and user access it via any browser.

In figure 21, OffPAD client overview shows basic information about the system. The different functions are aligned in the left navigation bar, where the user can invoke different functions by clicking on them. The user can invoke a transfer function, or see the history of his/her transaction details by just clicking on the respective navigation bar item.
In the proposed system, not all functions of the standard bank are present. We used the functions that seem relevant to the project only. So the options presented for the users are limited to the following:

1. Overview
2. Transfer Amount
3. History
4. About

Figure 22: Instructions for transfer operation

Figure 22 shows the information provided by the user when he/she selects the transfer operation. The process and steps of using the transfer operation are shown in the dialog window. The user then opens the application on the device and selects the transfer option. The user then enters the
destination account and the amount he/she wishes to transfer in the application. When the user presses the send to browser button, the data with the hash key is sent via Bluetooth to the client. When the application sends the details, this dialog window will automatically disappear and the user will be presented with the transfer form with the destination account number and amount automatically filled.

Figure 23 shows the information that is injected from the OffPAD device, particularly, the destination account number and the amount. After this, the user can fill in other information for the transfer form like KID and remarks.

Figure 23: Injected Information

Figure 24 shows the user providing additional information for the transfer operation. Since the destination account number and amount are already injected, the users just have to provide other information.

Figure 24: Adding additional fields
Figure 25: Transfer request processing in server

Figure 25 shows the transfer request received from the client with the transaction details. Upon receiving the request, the bank checks if the transaction is authentic or not, by comparing the hash values. If authentication succeeds, then the transfer is carried out, if not, it is halted and the user is informed.

Figure 26 shows the history of all the transactions. It displays the information about the transaction like destination account, amount, KID and remarks. It also shows success if the transaction was authentic and failed if not. If the transaction has not been processed then it will show not_processed.

Figure 26: History showing transaction details
Figure 27 presents a scenario, where the client is infected. The Trojan in the client alters the value in the amount field from 4000 to 4001. This is normally not noticeable by the users.

Figure 27: Value altered by Trojan

Figure 28 shows the transaction authentication failing, due to inconsistency in the data. Since the data have been altered in the client side, the hash generated by the app and hash generated by the server will mismatch, by which the bank will deny the transaction. This information shows that
the Trojan has tampered the data; therefore, the transaction process is terminated, and the user is informed about it.

<table>
<thead>
<tr>
<th>ID</th>
<th>Destination</th>
<th>Amount</th>
<th>KID</th>
<th>Remarks</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>65028.2548.36366</td>
<td>17000</td>
<td>testing true</td>
<td>should be failed</td>
<td>failed</td>
</tr>
<tr>
<td>29</td>
<td>222</td>
<td>111</td>
<td>unknown</td>
<td>unknown</td>
<td>not_processed</td>
</tr>
<tr>
<td>30</td>
<td>222</td>
<td>1212</td>
<td>unknown</td>
<td></td>
<td>success</td>
</tr>
<tr>
<td>31</td>
<td>555020300</td>
<td>25000</td>
<td></td>
<td></td>
<td>failed</td>
</tr>
<tr>
<td>32</td>
<td>552253522</td>
<td>1212500</td>
<td>unknown</td>
<td>unknown</td>
<td>not_processed</td>
</tr>
<tr>
<td>32</td>
<td>2525</td>
<td>111</td>
<td>1212</td>
<td>54545</td>
<td>failed</td>
</tr>
<tr>
<td>34</td>
<td>65028.12.14.14</td>
<td>23000</td>
<td>unknown</td>
<td>unknown</td>
<td>not_processed</td>
</tr>
<tr>
<td>35</td>
<td>22222222</td>
<td>111111</td>
<td>unknown</td>
<td>unknown</td>
<td>not_processed</td>
</tr>
<tr>
<td>36</td>
<td>553333333</td>
<td>5555</td>
<td>unknown</td>
<td>unknown</td>
<td>not_processed</td>
</tr>
<tr>
<td>37</td>
<td>553333222</td>
<td>111</td>
<td>1212</td>
<td>54545</td>
<td>failed</td>
</tr>
<tr>
<td>38</td>
<td>3333</td>
<td>1111</td>
<td></td>
<td></td>
<td>success</td>
</tr>
<tr>
<td>40</td>
<td>2825</td>
<td>12100</td>
<td></td>
<td></td>
<td>success</td>
</tr>
<tr>
<td>41</td>
<td>9812.25.5256</td>
<td>40000</td>
<td>21414124314</td>
<td>Value changed by Trojan</td>
<td>failed</td>
</tr>
</tbody>
</table>

Figure 29: History showing failed status

Figure 30: Information about the client

Figure 29 shows the history report to the user. It shows the failed status for the transactions, which was infected with the Trojan.

Figure 30 shows information about the client and author. This can be used as a page to describe the bank details and give information to the users.
Chapter 5

Technical Description

In this chapter, we describe the technical aspect of Data authentication in OffPAD using the authenticated Hash system. The choice of APIs and protocols used in this system are described according to the entity. This begins with the explanation of choices of technologies used and later explains the devices used on the process of development.

5.1 Choice of technology

The technologies used were selected to be compatible with the cross-platform scenario. Since java is used for android as well as client/server application, we chose java as the programming language to develop this system. Since we have a common platform for all the entities, the libraries and the APIs were quite simple to understand and implement. Each API and libraries are described according to the function they were used for.

5.1.1 RSA-Asymmetric cryptography

For the implementation of RSA-Asymmetric cryptography, we have used Bouncy Castle API \(^9\) for java servlet for implementation of client and Spongy Castle API \(^{10}\)(repackage of the Bouncy Castle API for the android platform) for android application in the device. The Bouncy Castle Crypto package is a java implementation of cryptographic algorithms. It was developed by the Legion of the Bouncy Castle. Also, the Spongy Castle is a repackage of Bouncy Castle for android. For the RSA Asymmetric cryptographic methods, we use 1024 bit key pair system. All the methods are implemented using the APIs. The 1024 bit key pair is shown in figures below.

---

\(^9\) https://www.bouncycastle.org/, accessed on 11/03/2015
\(^{10}\) https://github.com/rtyley/spongycastle, accessed on 11/03/2015
The most important package that was used for the encryption / decryption is the java.security package. This package includes all the methods that are needed to implement network security features in the java platform.

5.1.2 SHA-256 Hash

The SHA-256 hash is generated on both OffPAD device and the server system. They both use the same mechanisms for encoding the text into the SHA-256 hash value. The hash value on both sides is generated by the combination of destination account and amount. On the device side, the message is encoded using the MessageDigest class in the java.security package and then converted into base 16 hex values. The same happens on the server system for generation of the SHA-256 hash value.
5.2 Devices and Development

For the development of the system, we use various devices to represent each entity. For the OffPAD device, we use the phone with android operating system, and for the client and the server side we used MacBook pro, early 2011 model. The technical specifications of the entities are presented below.

5.2.1 Android Device
- Samsung Galaxy S5 with Android 5.1.0
- 1GB Ram and 16GB SD Card
- Quad-core 2.5 GHz Krait 400 processor
- Bluetooth v4.0, A2DP, EDR, LE, apt-x

5.2.2 Client and Server System
- MacBook pro 15-inch with early 2011 model
- OS X El Caption 10.11.2
- Processor 2.3Ghz with Intel core i7
- 8GB DDR3 memory

For the proposed system, we have developed an android application to be used in OffPAD device and for both client and server we have developed java servlets. Java servlets are running in tomcat server with MySQL as the database.

The android application uses dedicated Bluetooth adapter for communication in Bluetooth channel. It also uses MySQL as the database to store the data. The interface for data handling is done by ActiveAndroid library. It uses simple java objects to represent the data in the database and also provides various methods to manipulate those data. It also uses Spongy Castle API for
implementing RSA Asymmetric cryptography, which is repackaged android version of the Bouncy Castle API for consistency and methods compatibility.

The communication between the device and client is done via Bluetooth using bluecove API (http://bluecove.org/bluecove/). We use Bluetooth sockets for sending and receiving the data. For Bluetooth connection, the pairing is done previously and the addresses are hard coded for both device and the client. There are two addresses needed to have a Bluetooth connection, MAC address for the device and UUID for the service protocol. Bluetooth adapter and the Bluetooth socket are used for data transmission.

For the Bluetooth channel to be secured, we have encrypted the message flowing through the device to the client using RSA Asymmetric cryptography using 1024 key pair. The hash created in both android application and server servlet is a SHA-256 hash.

The client and server system are java web servlets with MySQL as database and tomcat v8.0 as hosting server. The client also uses bluecove API for Bluetooth operations. It uses the Bouncy Castle API methods for implementing RSA Asymmetric cryptographic functions. The private key used for decryption is stored in preference file in java servlet.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
</tr>
<tr>
<td>destination</td>
<td>varchar(60)</td>
</tr>
<tr>
<td>amount</td>
<td>varchar(15)</td>
</tr>
<tr>
<td>KID</td>
<td>varchar(60)</td>
</tr>
<tr>
<td>remarks</td>
<td>varchar(150)</td>
</tr>
<tr>
<td>hash_received</td>
<td>varchar(200)</td>
</tr>
<tr>
<td>hash_generated</td>
<td>varchar(200)</td>
</tr>
<tr>
<td>status</td>
<td>varchar(20)</td>
</tr>
</tbody>
</table>

Figure 34: Transaction table schema for MySQL

Since all the systems were implemented in Java, common methods that are defined in the java packages can be easily implemented. The android application was also developed in java language, as well as both of the servlets and all the libraries and API used. This made easy and effective implementation of the transfer function in a cross-platform scenario.
Chapter 6

Discussion

6.1 Threats to Online-banking:

6.1.1 Phishing:
One of the biggest threats to online banking is a phishing attack. As described before, one of the effective means to tackle this problem is the use of cognitive entity authentication. In Chapter 2, section 2.3.1, phishing and its effects on online banking are explained further. This explanation suggests that phishing attacks are targeted towards users who remain ignorant to such attacks.

6.1.2 Man in the Middle:
Man in the middle (MITM) attacks are also considered to be channel-breaking attacks. Such channel breaking attacks can penetrate two-factor authentication procedures as well. An example of this kind of attack is a communication between client and server. When the attack occurs, the attacker places itself between the client and server in the communication channels such as TCP (Transmission Control Protocol) connection. The TCP connection is split into two new connections: first connection is established between client and attacker and the second one is between attacker and server. This gives the attacker ample capacity to make changes to the data that flows through this intercepted channel. In our case of online banking, inside the application such as banking application, the attacker can change fields such as name and account numbers of recipients or amount.

6.1.3 Man in the Browser:
Man in the Browser attacks are also called content manipulation attacks. Unlike MITM attacks, in this attack, a Trojan is installed prior to the attack. The browser is compromised through the pre-installed Trojan. The attacker can gain easy access to transaction details and the users normally does not get any indication of an attack since all the normal functionalities remain undisturbed. These attacks are mostly browser-based attacks. However, the Trojan can infect the
The Trojan remains inactive until the trigger event such as a button click event is initialized. Once the trigger event occurs, the Trojan becomes active and modifies contents right before transactions occur. The server receives the modified content and there is no possibility of finding out whether the content is tampered with or not. The server completes this transaction and provides messages in the form of receipt or feedback to the user. The attacker then detects this generated receipt and the content is replaced with original content expected by the user. The receipt is then delivered to the user. Therefore, a victim never really discovers inconsistencies until later. Therefore, there is a need of a mechanism, which can maintain the security of data throughout the transaction process.

**6.2 Entity Authentication:**

In Chapter 2, Background, I have explained some aspects of entity authentication. In general “Entity Authentication” is divided into three categories: syntactic, semantic and cognitive-authentication.

The main focus of entity authentication is to validate the identity of a certain user. Although the above-mentioned three categories identify users according to different techniques, their task remains the same.

Syntactic authentication is somewhat similar to the X.800 peer-entity authentication, which is explained further in Chapter 2 as well. However, syntactic authentication has poor security prospects, especially against phishing attacks. While syntactic authentication is considered as a poor security mechanism, semantic authentication establishes the legitimacy of identity through a secondary process. The claim made by the entity is crosschecked on a database or list to check its validity. However, cognitive entity authentication provides a capacity to reason like human beings or advanced artificial intelligence systems. This capacity makes it possible to provide a sound judgment. Although I have explained some aspects of entity authentication, it is not our main focus for this thesis. The main focus of this thesis is data authentication.
6.3 **Focus of Thesis:**

This thesis is focused on the data authentication techniques. We wanted to understand and develop a system, which can provide data authentication throughout a transaction to guarantee that attacks such as manipulation of data can be prevented.

One particular research related to OCR (Optical Character Recognition) -based technology, which was carried out prior to this thesis, was the point of inception of this thesis. In the OCR-based research, an OffPAD device takes a photo of the screen of the device such as a mobile screen or a laptop screen. This photo is taken by the user to record the transaction details such as account information, amount etc. After this process, the transaction takes place between client and server like any normal banking transaction.

The bank receives transaction details, through the normal transaction process and it sends back authentication token. We assume this token might be a hash value. We also assume that it is sent through a different communication channel such as SMS or email to add another level of security.

OffPAD now uses OCR to convert transaction details to character values and generate its own hash. The two hash values received from the server as well as OffPAD are compared. An acknowledgment of such comparison includes two possibilities either match or mismatch. If matched, the transaction goes forward otherwise different counter-measures are initiated.

Some issues from this OCR-based research were related to the OCR technology. OCR relied on several factors for proper functioning since the quality of pictures is an underlying factor to affect this technology:

1. **Light conditions:**
   
   The OCR technology relied heavily on the quality of the image, and the illumination factors linked to lighting conditions. Images taken under dimly lit conditions were not acceptable.

2. **Screen types:**
   
   The OCR technology was also dependent on the resolution of the screens as well as types of screens. There is a variety of screens such as LCD (Liquid Crystal Display), LED...
(Light Emitting Diode). Similarly, there are a number of resolution values for such different screen types. This variety of screens and resolutions creates another limiting factor for OCR technology.

3. Resolution of cameras:
   Different mobile devices have different types of cameras, which consist of different resolution types. One of the most important factors affecting the quality of images is the resolution of cameras. Therefore, this is another important limiting factor for OCR.

4. Environmental factors:
   Another important factor is the environmental factors. Under certain conditions the user might not be able to take a focused image and it might end up as a jittery image or distorted. Such images create limitations as well.

Besides issues related to OCR; another issue is related to persistence of Trojan on the client system. Therefore, in search of a secured system with few limitations, we came across a solution of using OffPAD as a non-dedicated application. We used SHA-256 to generate hash key for consistency of data in device and server. Hash value is generated from the data sent from the OffPAD enabled device to the client, using SHA-256. From the use of SHA-256 in this thesis, we found several benefits:

1. SHA-256 has not been broken till date, and is expected to remain secure for a long time, e.g. for at least 10 years.
2. Device and server uses the same JRE (Java Runtime Environment), therefore, it adds the possibility to use a common library.
3. SHA-256 has higher compatibility with different types of browsers, mobile devices and servers.

For similar reasons, we used RSA asymmetric encryption methods. We also implemented shared key mechanism in this thesis. However, we discovered that shared key mechanism can be easier and better to implement if we use OffPAD as a dedicated device; since each device is unique to the individual and the public key can be implemented on the device itself for every user. This creates a possibility for banking organization to relate users to their corresponding private keys. The main focus of this thesis is on the use of non-dedicated devices. We intend to use the concept provided by OffPAD on any mobile device. This is made possible by creating a mobile application targeted for any android device. During this thesis, we realized that vulnerability in
applications is dependent on several factors including the OS (Operating System). In our case, the mobile application is dependent on android, which is an open source OS. Android devices, in general, such as mobile phones host a number of other applications, which can cause vulnerabilities in the entire system. If the security of any one application is compromised then the entire OS can also be affected. This creates a security hole in the OS. Another issue is related to the OS itself, which can be easily rooted by the users. The rooted devices have lower security since the user has already gained access to the root level features. For example, DNB bank in Norway does not allow its users to run “Vipps” application on rooted devices for security reasons. A French company called TazTag developed a mobile phone based on Android. The mobile device was named TPH-ONE which also provided similar functionalities as an OffPAD does, by providing identity management and security services (Varmedal et al., 2013). This particular device bears similarities to our application.
Chapter 7

Conclusion

In this thesis, we have discussed different types of authentication methods, devices, and technologies. They vary in types of security they provide to the users. This thesis, however, is focused on data authentication aspects. In this thesis, we have tried to answer three main research questions:

1. How to authenticate data using SHA-256 hash value function?
2. How to secure data transactions using RSA asymmetric encryption?
3. How to secure online banking transactions and apply data security using OffPAD non-dedicated application?

We have discussed some of the common threats related to online banking environments including different security procedures, which are put in place to tackle such threats. However, these security procedures have some shortcomings, which are utilized by attackers. To provide a higher level of security, we intend to utilize OffPAD techniques on a mobile application environment. A prototype of this mobile application was developed in this project. Prior work on OffPAD project, such as the previously mentioned OCR – technology based OffPAD systems were developed for two-factor authentication purposes. However, such authentication procedure required a secondary channel for communication as described before in Section 6.3.

7.1 OffPAD as a mobile application:

The prototype developed as a part of this thesis is an extension of the OffPAD project. The concepts of OffPAD are the basis on which this particular prototype is developed. Our prototype
however uses an Android-based mobile device. Breaking the trend of using a dedicated OffPAD device, we intended to use a non-dedicated device to understand if such a device can provide a higher level of security.

In our prototype, we have tried to focus on using Bluetooth enabled devices such as a normal mobile phone. The two specific data fields “Account Number” and “Amount” are automatically injected from the OffPAD enabled device. This automatic injection is a new concept we have used for this prototype. But, we realized that the security provided by Bluetooth channel itself is not very reliable. Since we use multiple platforms for communication of data, we used SHA-256 hash data integrity between them. The main focus of our thesis was to ensure that the data manipulation attacks as mentioned above in Section 6.1 and especially 6.1.3 can be avoided using our prototype. RSA asymmetric cryptographic techniques are used to encrypt the data when it is transmitted through the Bluetooth channel. Previous prototypes developed for OffPAD project were normally based upon two-factor authentication techniques while our prototype does not employ such techniques making the authentication process faster as well.

However, as described above in Section 4.2, we assume some aspects of the design that are related to client-server communication and security procedures. Although we have made it clear that TLS/SSL is utilized to secure the client-server communication channel, we have not provided practical implementations on this part since the thesis is not focused on the client-server communication. Our prototype is developed for non-dedicated devices such as mobile phones. During this project, we discovered that there are many issues related to mobile devices. Especially in Android based devices which use an open source OS, it is easier for users to manipulate root level features. This access to root level features can make the device vulnerable. Therefore, the prototype still has some limitations when it comes to real world implementation.
Chapter 8

Limitations and Future work

This prototype is developed as a concept to apply OffPAD based technology in a mobile application environment as a non-dedicated device. However, there are still new scopes of the project, which need to be explored to provide a real world implementation model.

8.1 Diverse device support

The current version of the prototype is developed for Android devices. Although the development process has introduced some new concepts of using the OffPAD technology as mentioned before, in the future, the prototype must be developed for other major mobile developers as well, such as Apple or Windows. This diversification can bring about new changes to the project and make it more universal.

When developing the prototype for Android-based mobile devices, we came across several issues related to the level of security provided by Open Source OS (Operating Systems). Unlike Android, Apple iOS or Windows OS, is not Open Source; therefore, prototype development for such platforms might provide new insights.

8.2 Use of SHA-256 and RSA

SHA-1 and MD5 are considered to be insecure; therefore, we have opted to use SHA-256. Another reason is that a large number of devices support SHA-256. However, this large number of device support is directed towards newer devices, applications and operating systems. However, there are known compatibility problems with older systems such as Windows XP SP2 (Service Pack 2).
While in server-side context, there are different types of requirements listed for different types of servers. For example, Microsoft has provided a list of requirements for server side SHA 256. While Java-based servers support SHA-256 for Java SDK 1.4.2 or higher installations, Apache based servers support SHA-256 with Apache version 2.x or higher. As for RSA cryptographic techniques, there are some assumptions.

RSA assumptions are:

- $A$ encrypts a message in plaintext $M$ for $B$ using $B$’s public key $(n,e)$ where cipher text $C = M^e \pmod{n}$
  - Here, $n$ = modulus (product of two or more large primes)
  - $e$ = public exponent
  - $B$ knows corresponding RSA private key $(n,d)$ which is used to decrypt $M = C^d \pmod{n}$
  - An adversary can learn $C$ from eavesdropping and public key as well. However, the private key remains the unknown factor.

RSA works on the assumptions that the value of $n$ is large enough and generated randomly then the plaintext message $M$ is a random value between 0 and $n-1$.

An adversary can decrypt a cipher text $C$ if he obtains decryptions from a legitimate recipient who has received cipher text in previous occasions. Such attacks are called cipher text attacks. Attacks related to bit security is called selective-cipher text attack where only partial information is decrypted from a message. To prevent such attacks, one padding scheme has been proposed called OAEP (Optimal Asymmetric Encryption Padding).

### 8.3 Client-Server Security Mechanism

One of the limitations of this study is the ability to provide complete transaction security. The main focus of this thesis lies in the development of data transmission authentication mechanism for the OffPAD. However, client-server security is one very important issue in this context. Although, we have mentioned that the client-server security will be established with TLS/SSL, however, practical implementation has not been done for this part of the thesis. In the future, we would like to extend the design into this arena as well.
8.4 Shared Key

Another area that we have not covered in this thesis is the shared key that the client and device should use for encryption and decryption. For the RSA Asymmetric encryption and decryption, there should be a mechanism to decide upon a key pair between a client and the device. It means that the public key of the client should be pre-known to the device. This part of sharing the PKI values among the parties has not been implemented and discussed in this thesis. This part is pre-assumed and a shared key pair is used for encryption and decryption.

8.5 Future Work

OffPAD as a dedicated device needs further research as it has a higher potential to provide security measures. All the limitations of this thesis mentioned above should be resolved and a complete authentication mechanism can be developed.

8.5.1 OffPAD as a Separate Device

One of the improvements that can be done is to implement an OffPAD as a separate dedicated device with just Bluetooth capabilities. By doing this, the security would be distributed upon client and OffPAD device. It would be difficult to compromise the security of two separate devices at the same time. Also, this device would have a display and numeric keypad by which the user can enter the values. Also, the user authentication should also be implemented in this device. It can be biometrics such as fingerprint or face detection capabilities, or can be simple two-factor authentication.

Such separate devices can also be used for user authentication in the online banking scenario, by which, using such devices, the user can directly login the bank website. The user provides the user authentication to the device, and the device automatically logs on the user on the client browser with values injection in the form.
8.5.2 User Authentication
Since we have focused more on the data authentication part in this thesis, the user authentication needs also to be implemented to be a complete authentication system. The user authentication has to be on both device side and on the client side. The two-factor authentication can be used for the user authentication. One-time password, the certificate-based approach using PKI, PIN codes etc. can be used for checking the authenticity of the users.

8.5.3 Key Sharing / Signing Mechanisms
A key assumption in terms of RSA asymmetric was shared key assumption. Where we use an existing key pair for encryption and decryption methods. Hence for future work, a strong key sharing mechanism should be in place. It can be implemented by issuing a certificate, which can be verified and used. We can use digital certificates for distributing public keys to all the parties. We can also use public key cryptography, which has many mechanisms for trusting an entity, like PGP’s “web of trust” and HTTPS’s public key infrastructure (PKI) model.
References


