Two Way Mobile Authentication System

Harish Dinne
&
Karthik Mandava

Blekinge Institute of Technology,
School of Computing,
SE-371 79, Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 457 38 50 00
Fax : +46 455 38 50 57
This thesis is submitted to the School of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

**Contact Information:**
Author(s): Harish Dinne & Karthik Mandava

E-mail: hari_d38@yahoo.com & karthikmandava@gmail.com

**University advisor:**

Dr. David Erman, david.erman@bth.se,
Ph.D., Telecommunication Systems
Blekinge Institute of Technology, School of Computing
SE-371 79, Karlskrona, Sweden
tel:+46-455-385658, tel:+46-709-285756
Web: http://www.bth.se/com/der.nsf

**University Examiner:**

Dr. Patrik Arlos,
Blekinge Institute of Technology, School of Computing,
SE-371 79, Karlskrona, Sweden
Mobil: +46 73 38 00 312
Email: patrik.arlos@bth.se

Blekinge Institute of Technology, School of Computing,
SE-371 79, Karlskrona, Sweden
Internet: www.bth.se
Phone: +46 457 38 50 00
Fax: +46 455 38 50 57
ABSTRACT

The ever increasing use of internet around the world has without doubt increased the usage of internet based services, e-business models, easier ways of communication and information sharing. Such drastic increase in usage of network based systems has made the current cyber security systems old dated as the hackers and attackers of networked systems is on the rise with new and modern attack methodologies. This has necessitated the need of more secure ways of communications. The issues of Confidentiality, Integrity and the Availability of systems are of prime importance and more research towards these issues has been called for around the world.

One of the major areas of security improvement is the way in which authentication of users is carried out. Even though many organizations still rely on static ID and password authentication system, this method is getting old and there is a requirement for a better way of authentication which is required. One of the solutions for this issue is the two factor authentication technique as a fundamental security function. Our thesis proposal explores the two factor authentication technique and implementation issues which can be used for the two factor authentication technique.

Two-factor authentication method is implemented in two main phases. In the first phase, the authenticator gets a request generated by the application to authenticate a specified user. When the request is received, it generates a one-time password and sends it through a SMS to a GSM cell phone registered for that specified user. The one-time password has a default timeout 5 minutes which is configurable. In the second phase of the authentication, a request is sent with the user id and a hash of the one-time password. If both the one-time and user specified password is valid then the user will be authenticated. [1] It proposes a secure, convenient and user friendly two factor authentication scheme and discusses its applications to online banking.

Keywords: Two factor authentication, One Time Password (OTP), Demo online Banking application, computer security, mobile device for authentication.
ACKNOWLEDGEMENT

We would like to convey our sincere and substantial gratitude to honorable supervisor David Erman for his intense efforts and precious guidance to improve the quality of our thesis.

We are especially grateful to our parents, who had always provided the courage, strength, encouragement, endless love, moral and financial support during our whole career.

We are thankful to our friends for their timely co-operation and assistance during tough times.

We also have best regards for BTH faculty including Mikeal Åsman and Lena Magnusson for their generous support and motivation in hard times.
### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OTP</td>
<td>One Time Password</td>
</tr>
<tr>
<td>2WAMS</td>
<td>2 Way Mobile Authentication Systems</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Messaging Service</td>
</tr>
<tr>
<td>PHP</td>
<td>Hypertext Preprocessor</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SIM</td>
<td>Subscriber Identity Module</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communication</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>FIPS</td>
<td>Federal Information Processing Standard</td>
</tr>
<tr>
<td>SHA1</td>
<td>Secure Hash Algorithm</td>
</tr>
<tr>
<td>DSA</td>
<td>Digital Signature Algorithm</td>
</tr>
<tr>
<td>DSS</td>
<td>Digital Signature Standard</td>
</tr>
</tbody>
</table>
# Table of Contents

Abstract .................................................................................................................. 3  
Acknowledgement ................................................................................................... 5  
Acronyms .................................................................................................................. 6  

1. Introduction  
1.1 Brief Introduction .......................................................................................... 9  
1.2 Motivation ....................................................................................................... 10  
1.3 Objective ....................................................................................................... 11  
1.4 Scope of the thesis ......................................................................................... 12  
1.5 Research questions ....................................................................................... 12  

2. Background and Related work ........................................................................... 14  

3. Existing Systems  
3.1 Types of existing systems ............................................................................ 15  
  3.1.1 Tokens .................................................................................................. 15  
  3.1.2 Biometrics ............................................................................................ 15  
  3.1.3 MobileID ............................................................................................... 16  
3.2 OTP implementation versus other existing methods .................................. 16  
3.3 OTPs versus other methods of securing data .............................................. 17  

4. Overview of Two Way Mobile Authentication System  
4.1 Analysis .......................................................................................................... 19  
  4.1.1 How Traditional passwords are compromised ...................................... 19  
  4.1.2 Vulnerabilities of a mobile device ......................................................... 20  
4.2 Features of Two Way Mobile Authentication System .................................. 21  
4.3 Limitations .................................................................................................... 22  

5. Technologies and Requirements  
5.1 Software Requirements .................................................................................. 23  
5.2 Hardware Requirements ............................................................................... 23  
5.3 Clickatell SMS Gateway ................................................................................ 23
6. **System Design and Implementation**
   6.1 Programming process ......................................................... 25
   6.2 Programming maintenance .................................................. 25
   6.3 Dataflow design ..................................................................... 26
   6.4 UML diagrams ....................................................................... 27
   6.5 Database design ...................................................................... 28
   6.6 Implementation ....................................................................... 29
   6.7 Testing .................................................................................... 34
   6.8 Procedure for the work done ................................................... 39
   6.9 Final Results .......................................................................... 40

7. **Conclusion and Future work** .................................................... 43

8. **Appendix** ............................................................................... 48

9. **Bibliography** .......................................................................... 56
1. Introduction

1.1 Brief Introduction:

Dynamic password (namely, One-Time-Password) technology is a sequence password system and is the only password system proved non-decripted in theory \cite{15}. Its basic idea is to add uncertain factor in authentication so that users need to provide different messages for authentication each time. By this way, the applications themselves can obtain higher security guarantee than those use static password technology. The typical implementation methods of OTP include Time Synchronization and Challenge/Response \cite{16}. No matter what methods are used to realize dynamic property of password for each authentication, the core is to ensure the randomness of factors added into the authentication. Many current OTP applications use mathematic methods like Hash function for dynamic passwords but still will suffer potential attacked risks Using static passwords for authentication, as it is commonly done, has quite a few security drawbacks: passwords can be guessed, forgotten, written down and stolen, eavesdropped or deliberately being told to other people. A better, more secure way of authentication is the so called "two-factor" or "strong authentication" based on one time passwords, instead of authenticating with a simple password \cite{12}. Strong authentication solutions using two identification factors require often an additional device, which could be inconvenient for the user and costly for the service providers. To avoid the usage of additional device, the mobile phone is used to receive the onetime password \cite{12}. Our project describes how effectively we can achieve the strong authentication using mobile phone without the need to carry the extra hardware for the onetime password.
1.2 Motivation:

In the past year, customers of some world’s largest banks have fallen victim to "man-in-the-middle," or MITM, identity theft schemes that have diminished the level of confidence that a customer have in regardence with online banking and battered bank reputations. As the term implies, the attackers stood in the middle of customers and banks and listen to all the communication between them, in order to steal account and other personal information. “In one MITM scheme last year involving a large U.S. banking company, the thieves sent seemingly authentic e-mails asking customers to verify their account information. The e-mails directed customers to a spoofed bank Web site that seemed legitimate but actually redirected the customers to a fake Web site set up by a hacker in Russia.” Due to this, consumers who do not feel safe online are increasingly steering clear of Internet banking sites and shutting out an important channel for financial services providers to expand their customer relationships. The industry research firm Gartner Inc. estimates that almost nine million adults in the United States have stopped banking online and that another 23.7 million decline to start out of security concerns. The continual spread of online scams and the reality that people are increasingly wary of online banking channels, raise the stakes for banks to protect customers and themselves from increasingly sophisticated cyber attacks.

In some cases the attacker sets up a trap and waits for the victim. The unaware victim goes about his work in a normal way unknowing that his data is being hacked. Then the hacker was able to intercept user password/account information and he can use the records in fraudulent transactions or as goods for sale to other criminals. Criminals also use Web sites to read, insert, and change messages between the bank and its customers.

These attacks highlight the shortcomings of secure socket layer protocol and multifactor authentication security measures that many financial institutions have adopted. These security measures are limited because they only require that the bank and customer trust one another and do not provide the added assurances required to thwart MITM or related schemes.

In the present day environment, there is a need to make sure that only the authorized clients can be able to access the secured data or sensible information. With the present tools available in the web, there is a high possibility of cracking the entire system or stealing the credentials which is barely insufficient for the authentication process. Hence one-factor authentication is not adequate in terms of security with the high risks involved in e-business.
To overcome all the above mentioned attacks we need a system which will solve all the above mentioned problems. So the solution for this is “Two-factor authentication”. [1] Two-factor authentication provides shielding for banks and their customers from MITM attacks. The two-factor authentication model uses an online password and an additional form of authentication (such as using one time password) for online security. This approach authenticates users but does not enable them to confirm that they are communicating with legitimate online site. So we planned to develop a system which will generate a onetime password that any bank customer can use it as a second factor authentication, so that user is authenticated strongly and also it will free all customers from above mentioned all attacks.

1.3 Objective

One of the major shortfalls of the one factor authentication systems is the ease with which they can be cracked. This threat has increased over time as new and modern methods can be used for guessing and cracking authentication systems. The need for better and more secure systems has given rise to the concept of the two factor authentication system. In this new system, first factor is just usual password that every one creates while registering or creating an account. The second factor is the one time password, that we generate using some secured functions and sent through SMS gate way to a GSM device usually a mobile phone.

The implementation and usage of a two factor authentication system is more complex and difficult to crack by an attacker. In the scope of this thesis, we will be designing a simple bank application. The authentication of a user happens in two steps and by the user providing the system with two passwords. One password is the general password that he/she has to enter every time a user wants to login. The second password is the One Time Password (OTP) that we generate in our application and send it to the [1]end user via SMS on his mobile phone which he must then type in to be granted access to the system, this password will expire automatically after some time. To generate this one time password we are using SHA1 algorithm, because the output generated in this algorithm is irreversible.

The implementation phase of the thesis include designing the website using PHP, HTML, CSS and Java Script, generation of a onetime password and delivery of the generated password to the GSM device user. The password delivery system is implemented by [3] Clickatell, SMS gateway service provider.

Finally we will design a demo bank application, which will authenticate a user with two different passwords. One password is of user choice and the second
password is that we generate with our application. One-time password systems attempt to alleviate the problem of "sniffed" passwords by making the replay of a password useless.  

1.4 Scope of the thesis

The two way mobile authentication system is an innovative technology used to solve the existing problems of the present one factor authentication which is a simple username and a password. The two way mobile authentication solves this problem by using a strong authentication with the combination of “something you know”, “something you have” and “something you are”. When compared the above three methods individually, all the methods have some vulnerabilities. Something you know—may be shared, something you have –may be stolen and something you are stronger but it is expensive to use in all the cases. So the combination provides a stronger authentication.

The thesis aimed towards the realization of a strong two factor authentication using mobile device to

- Provides with a cost effective and user friendly authentication.
- Avoids the use of a simple username and password system which is not secure anymore.
- Using the mobile as your authentication token.
- Ease to use any existing applications on web
- No additional use of hardware
- Easy to deploy
1.5. Research questions

1. How Secure is our generated OTP (One Time Password)?
2. Why mobile device in this application?
3. Why did we use PHP as our Programming Language?
4. Why SHA-1 algorithm and why not SHA-2?
5. How two factor authentications is achieved using OTP?
6. Where we implement this system?
7. What are the advantages of using this system?
8. What are the limitations on this application?
2. Background and Related Work:

By definition, authentication means using one or more mechanisms to prove that the person is who he claims to be \[^{13}\]. Once the identity of the human or machine is validated, the access can be granted. There are three universally recognized factors for authentication exist today are: what you know (e.g. passwords, PIN’s), what you have (e.g. smart cards or tokens), and what you are (e.g. figure prints, face recognition, biometrics, etc). “Two factor authentications is a mechanism which implements two of the above mentioned factors and is therefore considered stronger and more secure than the traditionally implemented one factor authentication system”. One of the examples of two factor authentication includes withdrawing money from an ATM machine. When someone wants to draw money from the ATM, first he\(\text{she}\) has to input his\(\text{her}\) ATM card i.e. what you have and again he\(\text{she}\) has to enter the pin number i.e. what you know in order to access his\(\text{her}\) account.

Recent work has been done in trying alternative factors such as a fourth factor, e.g. somebody you know, which is based on the notion of vouching \[^{13}\]. Only recently, two-factor authentication systems based on mobile devices have started to gather some interest within the research community. In \[^{1}\] an authentication mechanism is presented which requires both a Web and a GPRS connection. The end user enters userid/password details using a web-based interface and gets an OTP via short message service on his mobile phone, which he must then type in to be granted access to the system. The General Packet Radio Service (GPRS) connection is not convenient for the user since it can be very costly and network quality of service (including availability of network coverage) is not always satisfactory. In addition, security of the scheme relies on information (image) related to the user, but the underlying rationale needs to be expanded with further arguments. The work of \[^{18}\] and \[^{19}\] are contributions are related to mobile payments and give some of the same approaches to mobile user authentication. In \[^{18}\], messages are routed to the mobile device via a GSM-SMS service and depend on the phone number as a means of authentication. In \[^{19}\] the authors develop a stronger authentication mechanism based on information stored in a Subscriber Identity Module (SIM) card and the Authentication Centre of the subscriber's carrier. One drawback of this approach is the necessity for the financial service provider to enter into a prior agreement with the network carrier.
3. Existing Systems

3.1 Types of Existing systems:
There are several systems for dealing with two way mobile authentication. They may differ in delivering the password to the authorized user or a different entity based on the security constraints. Some of them are as follows

3.1.1 Tokens
A token is a device used to authorize the user with the services. A token may be software or hardware. Software tokens are used to identify the person electronically, i.e. it may be used as a password to access something. Hardware tokens are small hand held devices which carry the information which stores cryptographic keys, digital signatures or even bio-metric data by which we can send generated key number to a client system. Mostly all the hardware tokens have a display capability. The hardware tokens include a USB, digital pass etc.

Drawbacks
- A token shall be carried all the time.
- Special software is required to read the token.
- Anyone can access the information that has the token i.e. in case of theft.

3.1.2 Biometrics
A biometric authentication is the advanced form of authentication. A biometric authentication is nothing but it scans the user’s characteristics such as finger print and eye retina and stores in the form of a string. When the user tries to authenticate it matches with the stored data and then gives access when a commonality is achieved and when the user has gained access he can enter the password to view the required information[20].

Drawbacks
- Biometric authentication is convenient only for limited applications, since the system becomes very slow for a large number of users.
- Finger prints can be taken on a small tape and can be provided for the hardware.
- Additional hardware is required to detect the fingerprints and eye retinas.
3.1.3 Mobile ID

Mobile ID offers a strong two way authentication by authenticating the user to the service and service to the user. The mobile id works such a way that the user is required to send the code generated by the application after which the Mobile id generates a code to identify the user with the service[21].

Drawbacks

- Mobile phones with 2.5 G and third generation only are supported.
- Software is to be installed into the mobile device.

3.2 OTP implementation versus other existing methods:

One of the method[2] used in generation of OTP (One Time Passwords) is by using a mathematical algorithm to generate a new password based on the previously generated password (i.e., OTP (One Time Passwords) are, effectively a chain and must be used in a predefined order). This is not secure because once if hacker finds what sequence of passwords is using by the user, he can easily trace out the future OTP (One Time Passwords).

The cheapest way would be generating a[2] One-time password and then delivering it on a piece of paper which is already known by someone who generates the OTPs on a device. The reason for this is these systems avoids the costs of SMS messaging. Even though delivering the OTPS by this way is cheap, it is not feasible because of the time to deliver the password to the user is too long.[15] “Dynamic password (namely, One-Time-Password) technology is a sequence password system and is the only password system proved non-decrypted in theory[1]. Its basic idea is to add uncertain factor in authentication so that users need to provide different messages for authentication each time. By this way, the applications themselves can obtain higher security guarantee than those use static password technology”.

The other systems rely on electronic tokens which are algorithm-based. The OTP generators must handle the situation when a token is not properly synchronized with the server when the system requires the OTP to be entered on a default timeout which leads to additional development costs. Time-synchronized systems, avoid this at the cost of having to maintain a clock in the electronic tokens. Whether or not OTPs are time-synchronized is basically irrelevant for the degree of vulnerability, but avoids the need of re-entering the passwords if the server is expecting the last or next code that the token should have, because the server and token have drifted out-of-sync.

Compared to hardware tokens, as far as one has a phone or a mobile device, we can eliminate the need of carrying an extra item which would be of no use other
than generating the one time passwords. In terms of costs also using a phone as a token provides with the best convenience that it is not necessary to deliver devices to each end-user. For most users, a mobile phone can be trickle-charged to save the charging for at least some time in a day, but most proprietary tokens cannot be trickle-charged. However, most proprietary tokens have tamper-proof features.

3.3 OTPs versus other methods of securing data:

One-time passwords increases the vulnerability to social engineering. The attacks in which the phishers attempts to find the already used OTPs that they used in the past. In the year 2005 and 2006 these kind of attacks was used in Sweden and US. Also the time-synchronized one time passwords are vulnerable to phishing in two methods: The password can be used by the attacker as the original user must use the OTP, if the attacker gets the one time password in plaintext. The other kind of attack -- which may be compromised if the OTP system implements using the hash chain as above -- is that after the phisher uses the social engineering, the phisher should then use the past OTP codes to predict the OTP codes which may be used in the future.

Even though OTPs are most secured than the passwords we usually remember, The users of OTP systems are still vulnerable to MIM attacks, the OTPs shouldn’t be shared with the others and the use of an OTP in layered security is more safer instead of using the OTP alone; we can achieve layered security by using an OTP in combination with a password that is memorable to the user.

The benefits of using layered security is that a single sign-on in combination with one master password is safer than using only one layer of security during the sign-on, and thus the inconvenience of password fatigue can be avoided if we have long sessions with many passwords that needs to be entered during the mid-session. However, the drawbacks of using different kinds of security during a single sign-on is that one have the problems with security precautions every time they log in even if one is logging into the computer to access data which doesn't need as much security as some other sensible transactions that computer is used for. The following table shows how

2WMAS performs better than the other existing systems in terms of cost, complexity and protection.
<table>
<thead>
<tr>
<th>Method</th>
<th>Password</th>
<th>OTP + Password</th>
<th>Digital Certificates/PKI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advantages</td>
<td>Widely used and supported by the largest number of applications</td>
<td>Two-factor authentication compatible with password based infrastructure: zero client footprint option</td>
<td>Bi-directional authentication</td>
</tr>
<tr>
<td></td>
<td>Technology easily understood by users</td>
<td></td>
<td>Can provide two-factor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Non-repudiation</td>
</tr>
<tr>
<td>Disadvantages</td>
<td>Relies on human protection and management of the secret.</td>
<td>Requires possession of OTP generation software/hardware or access to a secondary channel for OTP transmission</td>
<td>Certificate management cost can be prohibitive for large user base.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Heavy footprint to manage on client.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Not compatible with small devices.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Requires distribution of certificate/smart card to client.</td>
</tr>
<tr>
<td>Key Vulnerabilities</td>
<td>Brute force</td>
<td>Man-in-the-middle/client insertion</td>
<td>User override of warnings</td>
</tr>
<tr>
<td></td>
<td>Man-in-the-middle/client insertion</td>
<td>Phishing (reduced to one time action)</td>
<td>Client insertion (reduced)</td>
</tr>
<tr>
<td></td>
<td>Phishing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Over the shoulder</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Keystroke loggers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applicability</td>
<td>Lower risk environments</td>
<td>B2C Commerce</td>
<td>Highly secure environments</td>
</tr>
<tr>
<td></td>
<td>Legacy environments</td>
<td>Enterprise Security (VPN)</td>
<td>Monetary or legal transactions where non-repudiation is a required feature</td>
</tr>
<tr>
<td></td>
<td>No network usage or protected network usage</td>
<td>Environments not suited for PKI (e.g. password based application infrastructure)</td>
<td>Environments where mutual authentication is required.</td>
</tr>
</tbody>
</table>
4.1 ANALYSIS

2 Way Mobile Authentication System (2WMAS) is an innovative authentication system that provides access to Web-based resources by using a two-way user authentication through the existing personal mobile phones. It is used to solve the security flaws of the web based Internet and Intranet, by involving the users to authenticate themselves using their personal mobile phones. The registration of the users has to be done in a secured manner before he can actually use the system.

It is designed to provide security to Web-based Internet and Intranet applications, and requires users to authenticate themselves with two unique criterion - a username and password, and a code which they get only during authentication [2] (a one-time password OTP sent to their mobile phone) before they are permitted to access a secured web resource. With 2WMAS, we can positively identify users and deliver services easily and in a most secured way to users, without having the need of an additional security system. End users can have the advantages of a very simple process that omits the need to remember multiple passwords.

As the Web-based Internet becomes the most important tool for financial transactions, the level of security becomes a major concern in an organization's transaction system. Transactions in these days are secured using passwords. Institutions spend huge amounts of money on secure SSL solutions to make sure the passwords are not tracked. But, in majority of cases security violations occurs above the reach of PKI and SSL solutions.

4.1.1 How the traditional password system can be compromised

1. Passwords can be captured while they are being entered into the browser. For instance, by 'trojan- horse' applications which most of users might have unknowingly installed, while installing shareware applications, while reading a malicious email, or while even visiting a dubious website. Some 'trojan- horse' applications like the infamous 'Back Orifice' can even control PCs, and allow hackers to view their screens the same way as a remote software application like PC anywhere!

2. Some users may store their passwords carelessly on their PCs, which can be seen by anyone with access to the terminals, including their PC repairman, or if they are using a cyber cafe machine, anyone who uses that terminal after them.
3. An inefficient programming APIs of a system might give clues to the hackers to bypass the entire system.

4.1.2 Vulnerabilities of mobile device:

In this section we will explore the vulnerabilities of mobile devices, in order to understand the security risks relating to these technologies. Human vulnerability factors, such as compromised PINs, are not taken into account.

**Vulnerability: Untrustworthy Interface** In many cases, the security of an application boils down to the question as to whether a really trustworthy user interface exists. Since Trojans or viruses might attack mobile devices which have become as complex as PCs, this assumption cannot always be made. Besides that, the full range of browser-related vulnerabilities, e.g., phishing applies to mobile devices as well. However, in some of the use-cases described below we assume that the mobile device in general is more secure than an ordinary.

**Vulnerability: Theft/Loss of the Device** An additional vulnerability, which is rather uncommon in the desktop world but very serious when it comes to mobile devices (and laptops), is theft or loss of the device. Because mobile devices are very small and are usually carried around by their owners they get stolen or lost very often. A mobile device could somehow be used to verify an individual's identity but it should not contain the individual's identity because mobile phones are frequently lost or stolen.

**Vulnerability: Man-in-the-middle-attacks** even the best protection against physical attacks and an unbreakable encryption scheme do not help when the smart card cannot identify the party on the other side. Especially in the case of online authentication, where the communication is tunneled via many hops, this vulnerability becomes a serious issue. The attacker inserts himself between the server and the smart card. Even if the channel is encrypted, both sides believe they are talking to each other, and the attacker can intercept, delete or modify the communications. The usual protection mechanism against this kind of attacks is mutual authentication.

**Vulnerability: Cryptanalytic attacks** these attacks directly target the cryptographic algorithms. Published algorithms are continuously being reviewed by the scientific
community. Successful cryptographic attacks will, over time, lead to a need for greater key lengths.

Vulnerability: Eavesdropping the communication between two contactless devices can be eavesdropped from a certain distance. In this case, the attacker would first record the communications from a close distance and later try to break the encryption (if any) with appropriate equipment. The most popular example for this vulnerability is a well-known weakness in the basic access control (BAC) mechanism deployed in electronic passports [13][17].

Vulnerability: Tracking In many cases, NFC/contactless devices have a unique identification number (UID) which they send on request in order to establish a communication. This enables the movements of individuals to be tracked. Whenever the owner of a contactless device (e.g., an NFC-enabled mobile phone) passes by an NFC reader (e.g., integrated into the entrance gates of a big shopping mall), the monitoring system could store the UID in a database. Over time, a specific profile of the movements of the person would be created [1]. Of course, location tracking of mobile devices via the GSM interface is an important issue anyway, but is outside the scope of this paper.

4.2 Features of 2 Way Mobile Authentication Systems

Double-criterion to check the identity of the User:

It provides a cost-effective solution to provide the web resources with a double-criterion authentication system. [1] Through a browser, a user requests permission to access a Web resource which needs an additional authentication code required for the Web Application. It then generates a [12] one-time access code and sends it to the mobile phone registered to the user by an SMS text message. The user has to enter the access code into the Web-browser to finish the authentication. After the user enters the authentication information, the system determines if the information submitted is valid or not. If valid it goes ahead with the Web Application thereby allowing the user to perform the necessary transactions, otherwise not. By separating Web Application with Authentication server, we can also divide the responsibilities to decrease the internal fraud.

Protecting the existing authentication system:

The 2WMAS could not [2] replace the existing authentication system, but instead serves as an added layer of security that protects and enriches the existing authentication system, either software or hardware.
Protecting against Internal fraud:

The system’s core authentication and messaging engine is such that it provides with a good level of security to safeguard from reverse engineering and program transformation of the software. A security platform is never secured if someone has the access to the parts of the application and its security algorithm, modify the content of code to reveal security flaws or even create a backdoor entry.

4.3 Limitations on 2 Way Mobile Authentication Systems

1. 2 Way Mobile Authentication cannot solve the problem of phishing (phishing is defined as a process of gathering personal data such as credentials, information of the credit cards and other sensitive data by impersonating as a trusted party through electronic communication).

2. A user cannot login to the system if the GSM gateway service provider’s servers are down where he could not receive the OTP even though he is a genuine user.

3. This system cannot be used when a user’s mobile network service provider terminates the connection due to the delay in bill payments and also poor signal of the network.
5. Technologies and Requirements

5.1 Software Requirements

Application Language: HTML / CSS / Java Script and PHP

Operating System: Linux / Windows

Protocols: HTTP

Web Server: Apache

5.2 Hardware Requirements

- Any mobile device which is capable of communicating with GSM networks.
- Any computer which have an Intel p4 or above processor, minimum server configuration of 512 MB RAM and 10 GB free hard disk space.
- Web server capable of uploading PHP Scripts.

5.3 SMS Gateway Service Provider (www.clickatell.com):

[SMS has shown significant resilience in market that is bombarded with media that all add to the clutter of daily Communications. SMS is a form of highly personal, immediate communication with high reach capability, low cost and high retention levels. With communications media converging, SMS is now accessible in many ways as a business tool. [3] Clickatell is an SMS Gateway provider, which provides an interface between an existing systems and the SMS Messaging Gateway. It is a lower level connectivity option, but offers the very good functionality and flexibility for the end user. With the API clickatell can set up alert-based SMS delivery from Clickatell’s server. Depending on the messaging requirements, clickatell may find one or more of Clickatell's products to suit clickatell needs, out of which they have opted HTTP API which is gives us the easy ways in order to connect to the Clickatell API for sending SMS.

HTTP/HTTPS API is one of the easiest server-based ways of communicating Clickatell’s gateway. We can use it either in the form of a HTTP-POST or as a URL that uses GET method for sending SMS. It is recommended to POST for larger data transfer, due to the size limitations of GET. Communication to the API can be done either via HTTPS on port 443 or HTTP on port 80. All calls we made to the API...
must be URL-encoded. The parameter names used here are case-sensitive. Batch messaging done in a variety of ways. For using clickatell’s API, we need to register at: www.clickatell.com and sign-up there and buy the SMS credits, so that we can send SMS.

Below is the procedure to send an SMS using clickatell (mentioned below is manual, we need to program it accordingly to integrate)

1. First we need to sign up for an HTTP/S account; we will be given a username, password and api_id (application id).
2. Once the registration has been activated we will receive 10 free credits with which to test our service.
3. Have the number we wish to send to ready in international format e.g. +448311234567.
4. Open the browser (e.g. Internet Explorer), and type the code in following sequence.
   http://api.clickatell.com/http/sendmsg?user=xxxxx&password=xxxxx&api_id=xxxxx&to=448311234567&text=Meet+me+at+home
5. The text of the message should be in above mentioned format, and the ‘+’ signs replace the spaces that are present in between words as above.
6. Press 'Enter' on the keyboard and the message will be sent the mobile phone of specified user.
6. System Design and Implementation

6.1 Programming Process

Below are the steps to be followed to demonstrate two way mobile authentication systems:

1. Create a project flow design.
2. Design good looking and attracting web pages and site flow with Hypertext markup language (HTML) and Cascading Style sheets (CSS).
3. Decide the database structure to store the registered user’s contact, log-in and account details.
4. Develop the code needful to navigate the application dynamically.
5. Integrate the Clickatell's HTTP/HTTPS API to our system to send SMS (which include the one-time password (OTP) generated by the SHA-1 algorithm)
6. Test the application to locate and remove any bugs.
7. Compile the tested application and deploy the files in the server.
8. Buy a domain name and point it to our server to make it live.

6.2 Programming Maintenance:

Apart from the bugs, we have to maintain the program we created. Program Maintenance is a term used for the updating of a program after the program is put into use. This updating may be a result of the users request or a change in the way the program needs to operate.
SYSTEM DESIGN

Data Flow Diagram
**UML Diagram**

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

Use case Diagrams depict the control flow of a functional system from a user’s point of view. Use cases are used during requirements gathering and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are the users that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

**Use Case:**

- Registers with the bank for a new account
- login the system using 2 Way mobile authentication
- Adds money to his account
- Transfers money to same or other bank accounts
- View the corresponding transaction details
DATABASE DESIGN

Database: bthbanken
Username: bthbanken
Password: Qua4rigadb
6.6 Implementation

index.php

This is the home page of the site and includes links to login, new user sign up and forgot password.

register.php

This is the registration page of a new user to the bank, where the user enters his login and contact details. This program performs basic client side (JavaScript) validations of the details entered. On submitting this page will redirect to loginform.php

loginform.php

This page checks whether the username submitted in register.php already exists in the database, if exists then, it will again redirect to register.php throwing an error “username already exist, please try another one...” and if does not exist, It will perform the following steps:

1. Inserts a new user login details in user_login table namely, username and password. The user_id is an auto incremented field and increments its value by 1.

2. Based on the last inserted user_id in user_login table, it will insert a new record in user_details table updating all the contact details of the user.

3. Inserts a new record in user_balance table by a default value of zero in acc_balance (account balance)

4. Displays the login form and on submitting this form will redirect to login_home.php
**login_home.php**

It checks whether the posted login details are valid or not. If they are invalid then, it redirects to index.php throwing an error message “invalid username / password”

Else if valid then, it performs the following steps:

1. Connects to the database and fetch the user_id of the username and password submitted and stores in a session variable.

2. Creates a one-time password (OTP) based on the SHA1 algorithm and stores it in a session variable.

3. Retrieves the contact number of the user, based on the user_id from the table user_details.

4. Creates a query string to send an SMS based on Clickatell’s API procedure and sends an SMS to the contact number retrieved.

5. After sending the SMS it will redirect the page to authenticate_password.php

**authenticate_password.php**

This page asks the user to enter the OTP sent to his mobile via SMS and on submitting this page will redirect to verify_password.php

**verify_password.php**

This page checks whether the entered OTP is equal to the already stored session password variable and if it is correct will redirect the page to transaction.php else it will again redirect the page to authenticate_password.php throwing an error “you have entered wrong password, please try again...”
**transaction.php**

This page performs the following:

1. Displays the account id which is same as user_id.

2. Retrieves the account balance of the user from user_balance and displays it.

3. Displays the credited and debited details from the table user_transfer and user_received accordingly.

**add_money.php**

This page displays a form asking the user an amount to add to his account balance. On submitting this page will post the values to verify_add.php.

**verify_add.php**

It will update the acc_balance field in the user_balance by adding the existing and the posted amount and redirect to transaction.php

**bank_selection.php**

It shows two options for the money transfer namely, “same bank” or “other bank”. On selection of the options will submit the page to redirect.php

**redirect.php**

It will redirect the page to samebank.php or otherbank.php based on the selection of money transfer in bank_selection.php
**samebank.php**

It shows the existing account numbers – names of the bank and asks for the amount to be transferred. On submitting this form will redirect the page to same_auth.php

**same_auth.php**

This page performs the following:

1. Stores the amount and account number posted from the samebank.php in a session variable.

2. Creates a onetime password (OTP) based on the SHA1 algorithm and stores it in a session variable.

3. Retrieves the contact number of the user, based on the user_id from the table user_details.

4. Creates a query string to send an SMS based on Clickatell’s API procedure and sends an SMS to the contact number retrieved.

5. asks the user to enter the OTP sent to his mobile via SMS and on submitting this page will redirect to aftersame_auth.php

**aftersame_auth.php**

This page checks whether the entered OTP is equal to the already stored session password variable and if it is not correct, will redirect the page to same_auth.php throwing an error “you have entered wrong password, please try again...” else if correct, it performs the following:

1. Deducts the transferred amount from the existing balance in user_balance.

2. Adds the same amount to the transferred account number.

3. Inserts a new record in user_transfer updating the details of the transferred
bank, account number and the transaction date. (credited details)

4. Adds the transferred amount to the account number (i.e. user_id) to which the money is transferred.

5. Inserts a new record in user_received updating the details of the transferred bank, account number and the transaction date. (debited details)

6. Redirects the page to success.html showing a message “your transaction is successfully completed”.

otherbank.php

It asks the user to enter the account number, bank name and the amount to be transferred. On submitting this form will redirect the page to other_auth.php

other_auth.php

This page performs the following:

1. Stores the amount, account number and the bank name posted from the otherbank.php in a session variable.

2. Creates a onetime password (OTP) based on the SHA1 algorithm and stores it in a session variable.

3. Retrieves the contact number of the user, based on the user_id from the table user_details.

4. Creates a query string to send an SMS based on Clickatell’s API procedure and sends an SMS to the contact number retrieved.

5. Asks the user to enter the OTP sent to his mobile via SMS and on submitting this page will redirect to afterother_auth.php
afterother_auth.php

This page checks whether the entered OTP is equal to the already stored session password variable and if it is not correct, will redirect the page to same_auth.php throwing an error “you have entered wrong password, please try again...” else if correct, it performs the following:

1. Deducts the transferred amount from the existing balance in user_balance.

2. Inserts a new record in user_transfer updating the details of the transferred bank, account number and the transaction date. (credited details)

3. Redirects the page to success.html showing a message “your transaction is successfully completed”.

6.7 TESTING

Testing Objectives

In light of the diversity of existing software testing, it is advantageous to consider the types of tests as they become available to a designer. This will also help identify the scope of a particular test and clarify its main advantages and disadvantages as well as make the developer aware about the limitations of this test.

Functional Tests are used to exercise the code with nominal inputs (input values) for which the expected values are available. We also know the boundary conditions for these inputs. For instance, functional testing of matrix multiplication can involve some data (matrices) for which the results are known in advance.

Performance Tests are utilized in order to determine the widely defined performance of the software system such as an execution time associated with various parts of the code, response time(in case of embedded systems),and device utilization. The intent of this type of testing is to identify weak points of a software system and quantifying its shortcomings, leading to further improvements.

Black Box-White (Glass) Box Testing: As the name suggests, the criterion leading to this type of discrimination specifies whether the internal (logical) structure of the system is available for testing purposes. If so, we are concerned with white box testing. If the internal structure is not available or exercised when developing the test suite, we confine ourselves to black box testing. Depending which way was selected,
the points of view on testing are also radically different. In black box testing we are interested to test what the system is supposed to do. The testing is worked out from input data perspective; subsequently we see if the outputs (actions) of the software match the expected values. Functional, stress, and performance tests fall under this general category. In white box testing, testing concentrates on what the system does. Essentially, using detailed knowledge of code, one creates a battery of tests in such a way that they exercise all components of the code (say, statements, branches, paths). Structural testing sub schemes white box testing.

The server was implemented using PHP. A Clickatell SMS gateway was used for sending SMS messages from the server side. The Clickatell API [3] was used to send the messages and the SHA 1 [4] algorithm was used to hash the password. [9] MYSQL was used as a database. There is no specific requirement for the client, just any mobile device that is capable of communicating with a GSM network.

First the system was tested by sending an SMS from the web server to the mobile phone and proved to be working with the GSM device. Next we have created some fake user accounts for testing purpose on the server. Once the client requests an OTP via an SMS, the server would check the user credentials, generate the OTP, and send it back instantly. The complete process of receiving the request, checking the factors, generating and sending the OTP was done in less than a second from the server’s side. This proves the efficiency and effectiveness of the system when dealing with many customers. A test was done to check if there are any chances of generating two identical hashes for two different users, i.e. generating a hash collision. In this experiment, the database was filled with information for ten fake users. The generated OTP was set to be 6 characters long. The OTP can consist of upper/lower-case characters and digits, yielding a total of 4.9E+91 different possible combinations. This is almost impossible to brute force with the existing computing facilities. For each of the ten users, 100,000 OTPs were generated and compared. A total of 1 million OTPs and all million OTPs were unique. We performed lot of tests to this application some of the test cases are as follows.

<table>
<thead>
<tr>
<th>Case 1</th>
<th>To ensure correct username and password.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected Results</td>
<td>It should be login when provided the correct username and password and the page should be redirected to another page asking for a one time password or access should be denied if they are wrong</td>
</tr>
<tr>
<td>Prerequisite</td>
<td>The user should have a valid username and password.</td>
</tr>
</tbody>
</table>
### Steps

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Steps</strong></td>
<td>Visit the website <a href="http://www.bthbanken.com/">http://www.bthbanken.com/</a>.</td>
</tr>
<tr>
<td></td>
<td>1. Enter the correct username and password in the fields provided.</td>
</tr>
<tr>
<td></td>
<td>2. Click on the login button</td>
</tr>
<tr>
<td></td>
<td>3. Perform the above steps above with entering wrong username and correct password and correct username and wrong password</td>
</tr>
</tbody>
</table>

### Results

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Results</strong></td>
<td>1. When provided the correct username and password the page gets redirected to a page asking to enter your one time password.</td>
</tr>
<tr>
<td></td>
<td>2. When provided the correct username and wrong password or wrong username and correct password, the access is denied asking to enter the correct username and password.</td>
</tr>
</tbody>
</table>

### Case 2

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Case 2</strong></td>
<td>To check whether the one time password is delivered to mobile phone and when entered it redirects to transaction page</td>
</tr>
<tr>
<td><strong>Expected Results</strong></td>
<td>After a successful login the page should be redirect to a transaction page.</td>
</tr>
<tr>
<td><strong>Prerequisite</strong></td>
<td>The user should have a mobile phone registered with the username and password.</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Steps</strong></td>
<td>Visit the website <a href="http://www.bthbanken.com/">http://www.bthbanken.com/</a>.</td>
</tr>
<tr>
<td></td>
<td>1. Check for a field asking to enter a onetime password.</td>
</tr>
<tr>
<td></td>
<td>2. Wait for a one time password sent to your mobile and enter the password code into the field.</td>
</tr>
<tr>
<td></td>
<td>3. Follow the above steps and enter a random password.</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Results</strong></td>
<td>1. The page is redirected to a transaction page when entered the correct password sent to a mobile phone.</td>
</tr>
<tr>
<td></td>
<td>2. The page displays an error message stating that the login is failed.</td>
</tr>
</tbody>
</table>

### Case 3

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Case 3</strong></td>
<td>Checking the application whether it will asks the password while transferring the money to another account.</td>
</tr>
<tr>
<td><strong>Expected Results</strong></td>
<td>The application should generate the one time password and send it to the mobile phone.</td>
</tr>
<tr>
<td><strong>Prerequisite</strong></td>
<td>The user should have a valid account and a GSM Device (mobile phone) with which the phone number registered with his bank account while registration.</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Steps</strong></td>
<td>Visit the website <a href="http://www.bthbanken.com/">http://www.bthbanken.com/</a>.</td>
</tr>
<tr>
<td></td>
<td>1. Enter your username &amp; password and log-in. If the details you entered are correct, you’ll be shown an authentication page.</td>
</tr>
<tr>
<td></td>
<td>2. You will receive an SMS to your mobile with an OTP. Then enter the password in the authentication page to authenticate yourself.</td>
</tr>
<tr>
<td></td>
<td>3. After this you will be considered as a valid user and you'll be shown your account details page.</td>
</tr>
</tbody>
</table>
4. Since, it is demo bank you need to add money to your account if you want to transfer funds to some other account. So, click on "Add Money" and add funds.
5. Now, you have some funds in your account. Then, click on "Money Transfer" to transfer funds to an account of same bank or other bank.
6. Once you select the bank option, you need to enter the bank details and funds and submit the page.
7. Then, again you will receive an SMS to your mobile for authentication and once it is through, the funds will be deducted from your account and added to the other account.

Results

Received OTP when the user wants to transfer the money to another bank.

<table>
<thead>
<tr>
<th>Case 4</th>
<th>Checking the application whether the session cookies were expired or not.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected Results</td>
<td>When the user presses the logout button and logout from the application, he is not able to login again without entering the regular password and OTP.</td>
</tr>
<tr>
<td>Perquisite</td>
<td>The user should have a valid account and a GSM Device (mobile phone) with which the phone number registered with his bank account while registration.</td>
</tr>
</tbody>
</table>

| Steps | Visit the website http://www.bthbanken.com/.
1. Enter your username & password and log-in. If the details you entered are correct, you'll be shown an authentication page.
2. You will receive an SMS to your mobile with an OTP, just enter the password in the authentication page to authenticate yourself.
3. After this you will be considered as a valid user and you'll be shown your account details page.
4. Then press the logout button.
5. Press the home button it should ask for password |
| Results | The application asked for the password, by this we can make sure that the session cookies were expired. |

Test Activities

In software testing we encounter a number of key activities:

Test plans, Test design, Test Cases, Test procedure, Test execution, Test report

A test plan indicates the scope, approach, resources and the schedule of testing activity. At this stage, one indicates what is to be or not to be tested and which tasks to perform. In addition, it is necessary to identify the sources and levels of risk in testing. Software testers are also identified. Test planning may begin as soon as the requirements are completed.

It is difficult to determine when to stop testing or when a reasonable number of faults have been detected. For these reasons, criteria should be provided as a
guideline for test completion. A test design refines the approach in a test plan. Test design also identifies specific features to be tested by the design, and define the associated test cases. It is strongly suggested that tests should be designed for regression testing (test previously executed can be repeated at a later point in development and maintenance). The test cases and test procedures are constructed in an implementation phase. One should strive for the most compact (smallest) collection of test cases (batteries) that still meet the goal. Good test cases have a high probability of detecting undiscovered errors. A test procedure identifies all steps required to operate the system and exercise the specified test cases to implement the already defined test design. Test execution is performing the test procedures. Test execution starts from the component level and moves up to the integration, system and acceptance level. A test report summarizes all outcomes of testing and highlights the discrepancies detected. Sting activities are distributed across the entire software life-cycle as shown in the figure.
6.8 Procedure for evaluating the work:

Now-a-days one of the most common problems now-a-days some applications are facing is authenticating a user to a specific application and some other problems as we mentioned in motivation. So to overcome those problems one has to use the two-factor authentication. So this model basically uses an online password and an additional form of authentication (such as using one time password) for online security. This approach authenticates users but does not enable them to confirm that they are communicating with legitimate online site. So we planned to develop a system which will generate a onetime password that any bank customer can use it as a second factor authentication, so that user is authenticated strongly and also it will free all customers from above mentioned all attacks.

We have studied about “how traditional passwords are broken” and what should we have to do in order to avoid the password cracking. We did some research on this and found a way so that is very hard (almost impossible) to crack the password we generate and deliver it to the user in a secure way.

To achieve the above said we need a demo bank application. So we started creating the bank application. [8] We used PHP, HTML, CSS and JavaScript for creating the application [7] since capability of PHP as a web service engine in both qualitative and quantitative aspects while comparing it with other web service engines implemented in java and C. To store the user data we used SQL. The details about how the information is passing from one page to another page were explained in the implementation section.

Now the website is ready and we need something to take the data from the web server to the user. There are different ways we can send the generated one time password to the user. We can print the password on a paper and give it to user or we can email the password and some other ways to achieve this. The easiest and cheapest way of doing this is sending the password to users GSM device i.e., usually a mobile phone. The reasons for why we have chosen the mobile phone were explained in comparison of other technologies section. Now the problem is to deliver the password from the web server to the user’s mobile device.

We have taken the help of Click-a-Tell SMS gateway’s API. The reason for using this gateway and description about “how to use this API” was explained in the “SMS Gateway Service Provider (www.clickatell.com)” section. Now the question is “how secure is our password?”
6.9 Final Results:

Security of the generated OTP (One Time Password):

We are using SHA1 algorithm to generate the one time password. SHA1 is basically a secure hash algorithm that will give a 160bit fixed output for any arbitrary input of data. So basically we are generating a random number by using the inbuilt random function that PHP has and giving the output of that random function as input to the SHA1 algorithm and just taking the first 6 bits as the onetime password.

Now the next question arises is the “how secure is the password if we use SHA1 algorithm?” For instance, SHA1 creates hash values of 160 bits. There are therefore different hash values, and although some data records may have the same hash value, it is only a remote possibility.

So, If we obtain a hash value and somehow manage to try out random messages, we are very likely to get one with the same hash value. However, this process would take far more than 100 million years with the hardware currently in use. Recent statistics says that the 160 bit hash value can be cracked even in less time. If this is the case, it can be broken if the attacker knows the 160bit hash value. In our case we are not using the whole 160bit value we are just using the first 6 bits so it is impossible to crack the password and find what the input was given. So by this explanation we can say that our OTP was 100% secure.

Use of the mobile device in this application:

ANS: In day to day life mobile phones have become the most important and cheapest means of communication. There are many applications using this technology to simplify human life in terms of cost and time. The growth of the mobile phones in the present generation is astonishing.

The system we designed generates an OTP which is used for the authentication for the user. The generated OTP can be delivered to the user by phone, e-mail, post etc, but delivering the OTP to the user or client is the easiest way.

So the 2WMAS uses the mobile phones as a device for authentication system which makes the human avoiding the need for carrying an external mobile device by using the mobile phone as an authentication device for secure transactions or access secured data. It also increases the assurance that only the bearer of the mobile device has been authorized to access.
Cryptanalysis of OTP:

SHA-1 algorithm abbreviates for secure hash algorithm. It takes any arbitrary input and generates a fixed output. SHA-2 is a family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-bit words where SHA-512 uses 64-bit words. There are also truncated versions of each standardized, known as SHA-224 and SHA-384. From the generated 160 bit hash value we are using the only first six characters.

So there is no need to go for SHA-2 because it is not possible for an attacker to find the similar hash value if he knows all the 160bit hash value. Even if there is one bit change in the input then the sha1 will provide a different output.

E.g.

SHA1 ("The quick brown fox jumps over the lazy dog")
= 2fd4e1c6 7a2d28fc ed849ee1 bb76e739 1b93eb12

SHA1 ("The quick brown fox jumps over the lazy cog")
= 2fd4e1c6 d25e1b3a fad3e85a 0bd17d9b 100db4b3

By seeing the above example we can conclude that even if the first six characters are same but the input to the SHA-1 algorithm are different. Even if the attacker finds a collision then also he cannot make use of it because the input to the SHA-1 function is a random function. The input is not same every time, it will be different every time. So it will generate a different hash value. Another important thing is that it is impossible to guess what will be the next value that is generated by that random function.

Applications: There are a lot of places where we can implement our system. Here are some of the examples described.

Online banking system: In this project we have developed a demo banking application where we can use this system for authenticating a user to bank that he himself is the right person to access his account.

Mobile number verification: Now-a-days buying or selling in online has become the common thing. In order to do that a user has to create an account before using that service. At the time of registration some websites has the phone field as a mandatory field. So the website won’t allow the user to access its services until the user’s phone number is verified. So we can use this application directly to validate the user and verify his phone number.
This system can also be used in Enterprise solutions which include secure remote access, Enterprise authentication and B2B Transactions. For consumer: Online banking, e-commerce and common authentication.
7. Conclusion:

Our thesis goal was to study and implement the two way authentication method and its advantages over the one way authentication system. Our first step was analysis where we studied the traditional authentication systems and how passwords are compromised in such systems and what can be done to negate the comprising factors. This was followed with the study of the limitations of the two way mobile authentication systems. Once the above were completed, the focus was shifted to the implementation of the two way authentication method. The algorithm selected is SHA-1 Algorithm, and then the implementation of the design for the password generation was carried out in PHP. This was followed by an application development of a Bank application and testing our implementation of the two way authentication system with such an application. The One Time Password (OTP) was sent to the GSM user through Clickatell, a SMS gateway provider. During the testing of the implementation, it was found that the system was working fine and that our implementation of the two way authentication system was working and had better security compared to the conventional one way authentication system. [2] The OTP password generator ensured that the same password was not repeated and the OTP will be expired after a time lapse. Our thesis goal to study and implement a two way authentication method was successful and the functionality implemented by us was working satisfactorily.

Future Work:

Probing deeper, the demo application in this thesis also provide a strong foundation for future work in Two Factor authentication for security applications. Future developments include a more user friendly GUI and extending the OTP algorithm so that password can be generated based on different cryptographic functions. In addition to that we can add features such as giving as choice to the user to choose from different ways to authenticate him to the system to which he was supposed to authenticate.
SCREEN SHOTS
BTH BANKEN is not a Real Bank (Just a Demo). It is not related to BTH University.
8. Appendix:

Demo:

Here is the small demo of our project:

Visit the website http://www.bthbanken.com/.

8. Click on the link "New User? Sign Up", you'll be directed to registration page.
9. Fill your genuine details and submit the page, you'll be then redirected to login page.
10. Enter your username & password and log-in. If the details you entered are correct, you'll be shown an authentication page.
11. You will receive an SMS to your mobile with an OTP, just enter the password in the authentication page to authenticate yourself.
12. After this you will be considered as a valid user and you'll be shown your account details page.
13. Since, it is demo bank you need to add money to your account if you want to transfer funds to some other account. So, click on "Add Money" and add funds.
14. Now, you have some funds in your account. Then, click on "Money Transfer" to transfer funds to an account of same bank or other bank.
15. Once you select the bank option, you need to enter the bank details and funds and submit the page.
16. Then, again you will receive an SMS to your mobile for authentication and once it is through, the funds will be deducted from your account and added to the other account.
17. Click on the "Current Transaction", you can review the transactions you made recently.

FIPS 180 -1 Standard:

"The Federal Information Processing Standards Publication Series of the National Institute of Standards and Technology (NIST) is the Official series of publications related to standards and guidelines Adopted and promulgated under the provisions of Section 111(d) of the Federal Property and Administrative Services Act of 1949 as amended by The Computer Security Act of 1987, Public Law 100-235. These mandates have given the Secretary of Commerce and NIST important responsibilities for improving the utilization and management of computer and
related Telecommunications systems in the Federal Government. The National Institute of Standards and Technology (NIST), provides leadership, technical guidance, and coordination of Government efforts in the improvement of standards in these areas through the Computer Systems Laboratory.

Secure Hash Algorithm (SHA-1) was specified in FIPS 180-1 standard and can be used to generate a fixed length of output message called a message digest. The output of SHA-1(message digest) is given as input to the Digital Signature Algorithm (DSA) as mentioned in the Digital Signature Standard (DSS) which validates the signature of the message. The SHA-1 should be used by both the sender and intended receiver of a message to compute and verify the digital signature.”

**SHA-1 Algorithm:**

The SHA1 encryption algorithm specifies a Secure Hash Algorithm (SHA1), which can be used to generate a condensed representation of a message called a message digest. The SHA1 is required for use with the Digital Signature Algorithm (DSA) as specified in the Digital Signature Standard (DSS) and whenever a secure hash algorithm is required. Both the transmitter and intended receiver of a message in computing and verifying a digital signature uses the SHA1.

SHA1 is used for computing a condensed representation of a message or a data file. When a message of any length < 2^64 bits is input, the SHA1 produces a 160-bit output called a message digest. The message digest can then be input to the Digital Signature Algorithm (DSA), which generates or verifies the signature for the message. Signing the message digest rather than the message often improves the efficiency of the process because the message digest is usually much smaller in size than the message. The same hash algorithm must be used by the verifier of a digital signature as was used by the creator of the digital signature. The SHA1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify. SHA1 is a technical revision of SHA (FIPS 180). A circular left shift operation has been added to the SHA (FIPS 180). SHA1 improves the security provided by the SHA standard. The SHA1 is based on principles similar to those used by the MD4 message digest algorithm.
Features of SHA1 algorithm:

- The SHA1 is used to compute a message digest for a message or data file that is provided as input.
- The message or data file should be considered to be a bit string.
- The length of the message is the number of bits in the message (the empty message has length 0).
- If the number of bits in a message is a multiple of 8, for compactness we can represent the message in hex.
- The purpose of message padding is to make the total length of a padded message a multiple of 512.
- The SHA1 sequentially processes blocks of 512 bits when computing the message digest.
- As a summary, a "1" followed by m "0"s followed by a 64-bit integer are appended to the end of the message to produce a padded message of length 512 * n.
- The 64-bit integer is l, the length of the original message.
- The padded message is then processed by the SHA1 as n 512-bit blocks.

PHP:

PHP: Hypertext PRE Processor, is a widely used, general-purpose and server side scripting language that was originally designed for web development, to produce dynamic web pages. We can embed PHP into HTML and that usually runs on a web server, before that we have to configure and process PHP code used to create web page content from it. That configuration can be deployed on most web servers and also into almost every operating system and platform for free of charge. Statistics shows PHP was installed on over 20 million websites and 1 million web servers.

MYSQL:

MYSQL is a RDBMS, which can be abbreviated as relational database management system. The database management system that we used in this application was MySQL. MySQL database will allows users to create a relational database management system structure on the web-server that may be present in somewhere to store data like fields or records or user details. This program runs like a server providing multi-user access to a number of databases. MySQL is most commonly used with PHP scripts to make innovative powerful and flexible server side applications. We basically create forms by using HTML with fields in them, and
MYSQL holds tables and PHP is used to send queries to the database, by combining these entire things one can create truly efficient projects in the web.

**TABLE STRUCTURE**

CREATE TABLE `user_balance` (
    `user_id` int(10) NOT NULL default '0',
    `acc_balance` varchar(255) NOT NULL default '0',
    PRIMARY KEY (`user_id`)
);

CREATE TABLE `user_details` (
    `user_id` int(10) NOT NULL default '0',
    `firstname` varchar(255) NOT NULL default '',
    `lastname` varchar(255) NOT NULL default '',
    `email` varchar(255) NOT NULL default '',
    `address` varchar(255) NOT NULL default '',
    `contactnumber` varchar(255) NOT NULL default '',
    `alternatenumber` varchar(255) NOT NULL default '',
    `state` varchar(255) NOT NULL default '',
    `zip` int(20) NOT NULL default '0'
);

CREATE TABLE `user_login` (
    `user_id` int(10) NOT NULL auto_increment,
    `user_name` varchar(255) NOT NULL default '',
    `user_pass` varchar(255) NOT NULL default '',
)
CREATE TABLE `user_received` (  
    `user_id` int(10) NOT NULL default '0',
    `received_bank_name` varchar(255) NOT NULL default '',
    `received_acc_number` varchar(255) NOT NULL default '0',
    `amount_received` varchar(255) NOT NULL default '0',
    `transaction_date` varchar(255) NOT NULL default ''
);  

CREATE TABLE `user_transfer` (  
    `user_id` int(10) NOT NULL default '0',
    `destination_bank_name` varchar(255) NOT NULL default '',
    `destination_acc_number` varchar(255) NOT NULL default '0',
    `amount_transferred` varchar(255) NOT NULL default '0',
    `transaction_date` varchar(255) NOT NULL default ''
);  

user_login  
Attributes: user_id, user_name, user_pass  
  
user_id : It is an auto incremented field which track the number of users registered with the bank. It is the primary key for this table.  
  
user_name : It stores the username of a registered user which is unique.  
  
user_pass: It stores the password corresponding to a username.
user_details

Attributes: user_id, firstname, lastname, email, address, contactnumber, alternatenumber, state, zip

user_id: It is a field which corresponds to the user_id in user_login table.

Firstname: It stores the First name of the registered user.

lastname: stores the Last name of the registered user.

eemail: stores the email id of the registered user.

address: stores the address

contactnumber: stores the mobile number of the user to which the SMS has to be sent.

alternatenumber: this field is optional and stores the alternate number of the user.

state: stores the resident state.

zip: stores the zipcode of the resident.

user_balance

Attributes: user_id, acc_balance

user_id: It is a field which corresponds to the user_id in user_login table.

acc_balance: It stores the net balance of a account holder. By default it has value zero.

user_received

Attributes: user_id, received_bank_name, received_acc_number, amount_received, transaction_date

user_id: It is a field which corresponds to the user_id in user_login table.

received_bank_name: It stores the name of the bank from which a user receives money.

received_acc_number: It stores the account number of the user who transfers the money.
amount_received: It stores the amount received.

transaction_date: stores the date of the transaction.

user_transfer
Attributes: user_id, destination_bank_name, destination_acc_number, amount_transferred, transaction_date

user_id: It is a field which corresponds to the user_id in user_login table.

destination_bank_name: It stores the name of the bank to which the money has to be transferred.

destination_acc_number: It stores the account number of the user to transfer the money.

amount_transferred: It stores the amount transferred.

transaction_date: stores the date of the transaction.

Sample Coding:
/* code to generate OTP and storing it in a session variable for comparison in the next step */
    $otp = sha1(mt_rand());
    $otp_password = substr($otp, 0, 6);
    $_SESSION['OTP'] = $otp_password;
/* below is the code to send a request to clickatell’s API to send an SMS */
$clickatell_sms_url = "http://api.clickatell.com/http/sendmsg?";
$clickatell_username = "rajan_eetc";
$clickatell_password = "Cefuv123";
$clickatell_api_id = "3205489";
$clickatell_to = "46".$sms_mobile_number['contactnumber'];
$clickatell_message = "BTH+BANKEN+Authentication+Password:+".$otp_password."++Thank+you+for+registering+an+account+with+BTH+BANKEN";

$clickatell_query_string = $clickatell_sms_url."user=".$clickatell_username."&password=".$clickatell_password."&api_id=".$clickatell_api_id."&to=".$clickatell_to."&text=".$clickatell_message;

$send_sms = file_get_contents($clickatell_query_string);
9. Bibliography:


guide. Micro, IEEE.

and Wireless Security, Proceedings of IFIP 1-NetSec 2001: Advances in

GSM authentication, Technical Report KHUL-MA-2002-3 Information
