Using Mobile Phones as Software Token for Generating Digital Signature Code – Digitally Signing an Online Banking Transaction

Abin Oommen Philip
M-Tech Computer Science Specialization in Data Security
TocH Institute of Science & Technology
Arakkunam, Kerala, India

ABSTRACT
Nowadays, Online banking security mechanisms focus on safe authentication mechanisms, but all these mechanisms are rendered useless if we are unable to ensure the integrity of the transactions made. Of late a new threat has emerged known as Man In The Browser attack, its capable of modifying a transaction in real time without the users notice, after the user has successfully logged in using safe authentication mechanisms.

In this paper we analyze the Man In the Browser attack and propose a solution based upon Digitally signing a transaction and using the mobile phones as a software token for Digital Signature code generation.

General Terms
Online Banking, Security, Digital Signature.

Keywords
Online Transactions, Man In Browser Attack, Digital Signature, Mobile Phones as Software Token.

1. INTRODUCTION
Online banking grew by 50 percent in 2009. As impressive as this statement is, it is dwarfed by the growth rate of banking Trojans and password-stealing malware, which was estimated to grow by over 180 percent in the same time period. In fact, intelligence agencies report that the speed and sophistication of such malware is outpacing most anti-virus and firewall updates.

These Trojans can infect a user's PC, and then launch man-in-the-browser attacks that can completely circumvent even the strongest user authentication measures.

Once inside, fraudsters can do significant damage to both consumer and corporate online banking accounts – for example by wiring money externally, or transferring funds via automated clearing house (ACH) or bill payment systems. The success of the Man In The Browser (MITB) and Man In The Middle (MITM) attacks [1] highlight the false sense of security that many types of authentication solutions can give IT/Security teams within organizations. In the case of MITM, deploying advanced two factor authentication solutions [2] like smartcards, hardware tokens, One Time Password's [3] or PKI have long been considered sufficient protection against identity theft techniques. However, since the MITB attack piggybacks on authenticated sessions rather than trying to steal or impersonate an identity, most authentication technologies are incapable of preventing its success.

In this paper we take a brief look into how the MITB attack takes place how it is capable of modifying an online transaction. We propose a solution based on using mobile phones as software token for Digital signature code generation. Digital signature is known to ensure the authenticity and integrity of a transaction. Mobile phones have become a daily part of our life, thus we can use the mobile phone as software token to generate Digital Signature code.

2. MAN IN THE BROWSER ATTACK SCENARIO
A new threat is emerging that attacks browsers by means of Trojan horses. The new breed of new Trojan horses can modify the transactions on-the-fly, as they are formed in browsers, and still display the user's intended transaction to him. Fraudster writes malicious code (often hidden in e-mail spam scams etc.), which infects account holders' computers with a Trojan capable of executing man-in-the-browser attacks. Structurally they are a man-in-the-middle attack between the user and the security mechanisms of the browser. Distinct from Phishing attacks [4] which rely upon similar but fraudulent websites, these new attacks cannot be detected by the user at all, as they are using real services, the user is correctly logged-in as normal, and there is no difference to be seen.

The MITB threat utilizes a malware Trojan [5] on a victim's computer that is able to modify Web transactions as they occur in real time. The Trojan does not intervene until after a user has authenticated himself with his financial institution using any authentication technology, including OTP tokens, smartcards and PKI. Once connected to the legitimate site it 'piggybacks' on a legitimate authenticated session between the user and the financial institution, the MITB attack alters the appearance of

transactions in the user's browser. The MITB attack modifies the transaction even before the transaction details are passed through the https link between the users website interface and banks server. As the alteration occurs in real-time, the MITB prevents the user from detecting the fraudulent activity. For example, the user thinks he is transferring funds between accounts to pay bills, and the browser displays the transfer, when in fact the MITB attacker is actually transferring the user's funds into the account of a third party. The user views and confirms what he thinks are his intended transactions, only to become an unknowing accomplice to raiding his own account.

An example of how an MITM attack would succeed:

1. Alice requests transfer of $1000 to Bob
2. MITB alters transfer request to transfer $21000 to Abe
3. MITB submits fraudulent request to bank
4. Bank requests confirmation of transfer of $21000 to Abe
5. MITB alters confirmation page to present user with original request
6. Alice reviews the transaction details and confirms request
7. Bank transfers $21000 to Abe

3. DIGITALLY SIGNING A TRANSACTION

A digital signature ensures integrity and authenticity of a transaction [6]. By digitally signing a transaction we mean taking the Hash [7] of the transaction details and encrypting the Hash using user's Private key. The beauty of the Hash function is that even if there is slight modification made in the transaction details it drastically changes the Hash result. This encrypted Hash code is known as Digital Signature code (DS).

Represented as,

$$DS = E(PR_{user}, H(\text{Transaction Details}))$$

Where,

E = RSA encryption function
PR_{user} = private key of user
H= SHA 256/384/512 hash function

The Digital Signature code can be decrypted to obtain back the Hash result using the user's corresponding Public key pair.

Represented as,

$$HASH = D(PU_{user}, DS)$$

Where,

D = decryption using RSA
PU_{user} = Public Key of user

The Hash of the transaction details as obtained at the receiver's side is computed and compared with the decrypted hash. If both the Hash results match, we can ensure integrity and authenticity of the transaction and be sure that the transaction was not modified before or during the transit. RSA algorithm [8] is used for encryption and decryption purpose.

4. PROPOSED SYSTEM

In this paper, we propose using mobile phones as a software token system that is capable of generating a Digital Signature. This requires designing a software application that can be installed onto customer's mobile phones i.e. capable of generating a unique Digital Signature code for each particular transaction details entered onto the mobile device. It uses user's unique Private key embedded into the application in order to sign the transaction details and generate the Digital Signature code. The application can either be directly installed onto the customer's mobile device or downloaded from the banks site as a .jar file after proper authentication.

Steps in Digitally Signing a Transaction using Mobile Phone as Software token [see Figure 1]

1. The user enters the banks website after proper authentication using Two Factor or any other safe authentication mechanisms

2. When the user wants to initiate a transaction to another account he selects the Make Payment option on the banks website.

3. The user then enters the payees account no and the amount to be transferred onto the banks website. The bank provides a unique reference code for this particular transaction.

4. The user then runs the Digital Signature code generation application on his mobile phone by authenticating himself using his Account no and password.

5. The user enters the transaction parameters such as the reference code, payee account no and amount as on the banks website onto his mobile phone and selects the digital signature code generation option.

6. This Digital Signature code generated on the users mobile phone is then entered in addition to the transaction details onto the banks website to proceed the transaction.

The proposed system is secure and consists of two parts:

1. Software installed on the client's mobile phone to generate the Digital Signature code and

2. Server side verification software to verify the authenticity of the user and integrity of the transaction based upon the Digital signature code.

In the following sections we will see how the Digital Signature Code is generated on the user's mobile phone and verified at the server side.

4.1 Digital Signature Generation

The Digital Signature code is generated on the Mobile phone by entering the following factors from the banks website onto the mobile phone interface.

- **Ref Code**: A unique Reference code provided by the bank for each transaction, unique for each user and each transaction.
- **Payee Acc No**: It indicates the account to which transaction has to be made. It's entered by the user.

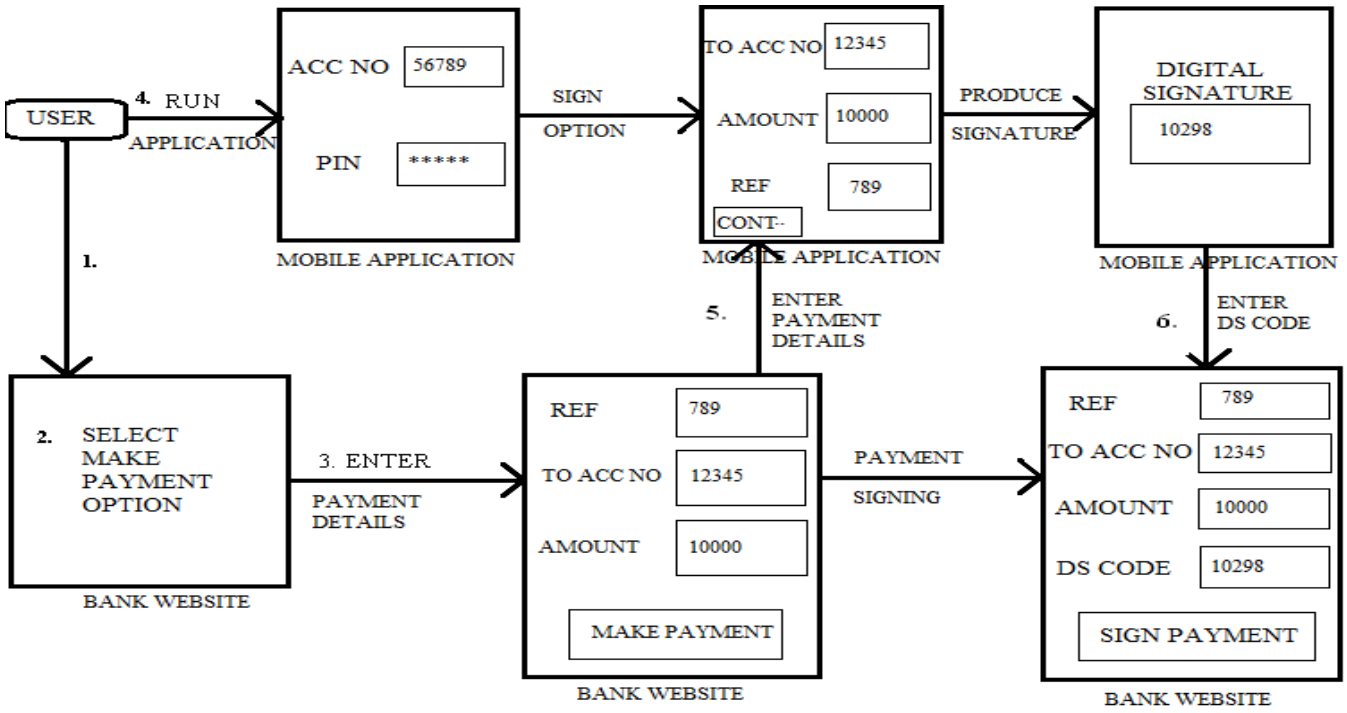

Figure 1:Block Diagram Representing user Interface with Mobile Application and Banks Website

- **Amount:** It indicates the amount the user wishes to transfer.

After entering the following factors onto the Mobile Interface the user selects the sign option.

The application concatenates the above factors and the result is hashed using SHA-256/384/512 which returns a 256/384/512 bit result depending on the Hash used. The message is then shrunk to specified length of 32 or 40 bits (i.e.: 4 or 5 digits) by breaking it into two halves and XOR-ing the two halves repeatedly and

padding extra bits if necessary. The result is then encrypted using user's Private Key to obtain the Digital Signature code (DS). RSA encryption scheme is used to encrypt the Hash of the Transaction details.

Represented as,

$$DS = E (PR_{user} , H(acc\ no, ref, amt))$$

Where,

E = RSA encryption function
PR_{user} = private key of user
H = SHA 256/384/512 hash function
Acc no = receivers account no
Ref =reference no for the particular transaction
Amt = amount selected for transaction

5.2 Client Interface

A J2ME program is developed and can be installed onto the user's mobile phone as a .jar file. The .jar file is run and the application gets installed onto the mobile phone. The application is capable of generating unique Digital Signature codes for each transaction. The program has an easy to use GUI that is developed using the Net Beans drag and drop interface. The program can be run on any J2ME-enabled mobile phone. The private key corresponding to each user is embedded into the particular application software code on the users mobile.

In order for the user to run the Digital Signature Code generation application, the user must enter his username and PIN on the mobile phone interface and authenticate himself and select the Digital Signature generation option. The user then enters the transaction details from the banks website onto the mobile phone

and the application generates a Digital signature code corresponding to the particular transaction. The username, PIN, and generated Digital Signature Code is never stored on the mobile phone. Even if the mobile device is stolen a third party cannot run the application as proper authentication is required to run the application

5.3 Sever Verification

Initially the banks server generates unique (Private, Public) Key pairs. The private key is installed into the application code, making each application unique for each user. The corresponding Public key is stored in the Database against the user's data. The application can either be installed directly onto the users mobile from the banks office or downloaded from the banks website as a .jar file after proper authentication.

In order to make a transaction, the user enters the banks site through his existing authentication means. When the bank server receives a transaction request along with transaction details and Digital Signature code, the Public Key corresponding to the

particular user is applied onto the Digital Signature code provided by the user to obtain back the hash result of the transaction signed by the user. This ensures authenticity of the transaction as only the user can sign the transaction using the corresponding private key pair.

The Hash is obtained back as follows,

$$HASH = D(PU_{user}, DS)$$

Where,

D = decryption using RSA

PU_{user} = Public Key of user

The banks server re-computes the hash of the transaction details the bank has received from the user and compares it with the decrypted hash. If both the hash results match, integrity of the transaction can be ensured and we can be sure that the transaction was not modified. If the Hashes do not match, it means that some modifications have been made in the transaction and the transaction request cannot be proceeded and a corresponding message is send to the user.

6. CONCLUSION

Today much focus is on ensuring safer authentication mechanisms in online banking process, but all authentication mechanisms are rendered useless, unless we ensure the integrity of the transactions. The Man-In-the Browser attacks are capable of modifying a transaction even if we ensure a safe authentication mechanism.

The paper proposes, a solution using mobile phones as software token to generate Digital Signature code in-order to digitally sign a transaction, thus ensuring authenticity, integrity and non-repudiation of an online banking transaction. The application software has been designed for java supporting mobile phones, it can be suitably modified to support other mobile OS.

7. REFERENCES

[1] Protecting Online Customers from Man-in the-Browser and Man-in-the-Middle Attacks, Whitepaper by ARCOT

[2] Two Factor Authentication Using Mobile Phones Fadi Aloul, Syed Zahidi And Wassim El-Hajj.

[3] Implementation of OTP Two Factor Authentication System From Visolve For Banks, Whitepaper.

[4] nisr-wp-phishing.pdf, The Phishing guide NGS Ltd.

[5] Online_Banking_Fraud_Prevention_WP_US_4301[1].pdf, Whitepaper by ACI

[6] An introduction to cryptography and digital signature by Ian Curry.

[7] Descriptions of SHA-256, SHA-384, and SHA-512, sha 256-384-512 pdf.

[8] RSA Encryption, http://www.geometer.org/mathcircles/RSA .pdf

[9] Web based Digital Signature Verification solution by Maestro

[10] How to increase the Online banking users confidence by Mr.Gilbert