Software Tokens Based Two Factor Authentication Scheme

Manav Singhal and Shashikala Tapaswi

Abstract—**This paper describes a method of implementation of Two Factor Authentication using Mobile handsets. This two factor authentication is based on Time Synchronous Authentication using the RFC1321 MD5 Message Digest Algorithm of Epoch Time, Personal Identity Number (PIN) and Init- Secret. The password generated would be One Time Password (OTP) which would be valid for 60 seconds only after which it expires and the user can not login through that password. The proposed method has been implemented and tested using J2ME OS based mobile Handsets. It can easily be further extended to Android based Handsets also.**

Index Terms—**Mobile security, one time password, two factor authentication**

I. Introduction

Today Security Risks are of great concerns. Many companies are comfortable with protecting their confidential information and transactions with a password. A simple password security may be effective for protecting non-critical data. But memorizing passwords, administrative issues and password hacking tools render a password-only authentication policy inadequate for protecting confidential information.

Authentication [1] is the act of establishing or confirming something (or someone) as *authentic*, i.e. that claims made by or about the subject are true. This might involve confirming the identity of a person, tracing the origins of an artifact, ensuring that a product is what it's packaging and labeling claims to be, or assuring that a computer program is a trusted one.

There have been different strategies proposed for making authentication mechanism more and more secure. There are different ways by which the secure passwords can be hacked such as Hashing, Guessing, Default Passwords, Brute Force and Hashing. Generally, a password containing both uppercase & lowercase characters, numbers and special characters too; is a strong password and can never be guessed. But still is not much secure way of authentication.

One way to strengthen your authentication policy is by adding factors such as tokens, smart cards, digital certificates and biometrics. The most common form of multi-factor authentication is two-factor authentication using a token or smart card as the second form of identification.

For the Two Factor Authentication (T-FA), the user has to carry a token/smart card which is really not feasible as for different organizations then he has to carry many Cards. So Mobile Phones can be a good option as every person carries a mobile handset these days. Mobile Phones now have higher resolution cameras and near high definition video with huge amounts of memory to enable storage of images and music. Now the Internet can be browsed through your handset and 3G and Wireless LAN connectivity is also available.

In this paper, an attempt has been made to implement Two Factor Authentication scheme using Software Tokens so that the user can validate and authenticate his identity using his Mobile Phones. In the next Section, background about Two Factor Authentications is given, different types of T-FA. Section III describe about Design Implementation. And Section IV explains System Testing and then Conclusion.

Manuscript received February 25, 2012; revised April 26, 2012.

The authors are from ABV-Indian Institute of Information Technology and Management, Gwalior (e-mail: singhal.manav@gmail.com; stapaswi@iiitm.ac.in)

II. Back Ground

Authentication is generally a process which is required to access the secure and confidential data. So the user is required to establish his identity based on the different factors. Here are the three mechanisms by which the user authentication can be done are as given below: [2]

- *Knowledge Factors*

The first one is regarding what the requestor individually knows as a secret.

For e.g. - Password or a Personal Identity Number (PIN)

- *Ownership Factors*

The second is regarding what the requesting owner uniquely has.

For e.g. - Passport or an ID-card

- *Inheritance Factors*

The third one is regarding what the requesting bearer individually is.

For e.g. - Biometric data, like a fingerprint or the face geometry or the eye retina

Now the Two-factor authentication (T-FA) means using any independent two of these authentication methods (passport + Personal Identity Number (PIN)) to increase the assurance that the bearer has been authorized to access secure systems. And Multi-factor authentication hence means two or more of the authentication factors are required for being authenticated to access the data.

Two-factor authentication is based on the concept of "something you have" and "something you know". Now a days, the most common example in real life is the typical ATM Banking Scenario – which combines something you know (your password) and something you have (your ATM Card) to prove your identity that who you are. Two –Factor Authentication is vital for effective network security. Hence

three types of T-FA are used generally in networks [3]:

1) Challenge Response Authentication

In this method, there are five steps defined in which the user authenticates himself and proves his identity.

- Firstly the user enters his username and password.
- The Server sends an 8 digit challenge.
- Now the user enters the 8 digit challenge.
- 8-digit response is displayed on the token.
- Then the user enters the 8 digit response and thus validated to access the data.

Challenge Response Method proceeds through a laborious five steps process and it is much prone to user error.

2) Event Synchronous Authentication

In this method, there are only three steps in which the token code is based on the next number in the sequence, not the random number generation scheme which makes it much prone to the hacking.

- User activates the next token code by pushing the button the token.
- User enters the username and passcode (the passcode is an event produced token code and the user's PIN).
- Then the server authenticates by matching the user passcode with the server passcode (Server Passcode is based on the next event in the sequence).

3) Time Synchronous Authentication

In this method also, we have three steps for the authentication but here the difference is that both the user and the server have the internal clocks that are synchronized hence they are called time synchronous. And they also have the identical seeds.

A seed is the starting values used by the random number generation to create a pseudo random number.

- The user enters the username and Passcode (the passcode is a 4 to 8 digit random token code and the User's PIN).
- The Server and the token create the token code by combining seed record and current Greenwich Mean Time.
- The Server authenticates the user passcode with the server passcode and thus validated if found correct.

There are various advantages of the Time Synchronous Authentication over the Event Synchronous and Challenge Response Authentication.

- *Security Concern*

The Time Synchronous is much secured than the other two because it is based on the token's secret seed which we can say is virtually hacker proof. The other two are less sophisticated and prone to attacks.

- *Easy Use*

The time synchronous events steps are less while event synchronous consists of three steps and challenge response consists of five steps.

- *Portability*

Time Synchronous hardware tokens are extremely portable because they are in not tied to the user's desktop. We can also choose from any number of factors that can be easily integrated into the Palm devices and mobile phones.

RSA Tokens are based on the Time Synchronous and hence are much safe and efficient and less vulnerable to hacking attacks.

A security token may be a device that an authorized user of computer services is given to ease authentication. Security tokens are used to prove one's identity electronically .We have two types of Tokens- Hardware Token and Software Tokens.

The token is used in addition to or in place of a password to prove that the customer is who they claim to be. The token acts like an electronic key to access something. Hardware tokens are typically small enough to be carried in a pocket or purse and often are designed to attach to the user's keychain. Software Tokens are type of security device that may be used to authorize the use of computer services. Software tokens are stored on a general-purpose electronic device such as a desktop computer, laptop, PDA, or mobile phone.

A new category of T-FA tools transforms the PC user's mobile phone into a token device using SMS messaging, an interactive telephone call, or via downloadable application to a smart phone, which can display a six-digit security code that changes every 30 seconds (assigned to a unique credential ID for user registration).Since the user now communicates over two channels, the mobile phone becomes a two-factor, two-channel authentication mechanism.

There are different commercial products available in the market today such as RSA SecurID [4], Vasco's DigiPass [5], and PhoneFactor [6].

RSA SecurID two-factor authentication is based on something you know (a password or PIN) and something you have (an authenticator) providing a much more reliable level of user authentication than reusable passwords. RSA offers the enterprises a wide range of user authentication options to help in positively identifying the users before they interact with mission-critical data and applications.

VASCO's Two-Factor Authentication technology replaces insecure static passwords with strong, dynamic OTP that can never be reused at future logins. This enhancement increases access security without compromising user convenience. Once activated, the end-user enters and confirms their PIN each time a one-time password is generated. VASCO VACMAN Controller performs all OTP and signature validation functions for the complete DIGIPASS product line, including DIGIPASS for Mobile.

PhoneFactor adds a second factor of authentication to any existing corporate or website login. The username and password has to be entered first. Instantly, PhoneFactor sends you a text message with a passcode. Simply reply to the text message with the passcode to authenticate. This is also a good Two Factor Authentication Product.

Here an attempt has been made to develop the software tokens for mobile phones which are based on Time Synchronous Authentication which would be efficient and less vulnerable to hacking attacks.

III. Experimental Details and Implementation

In this paper, we propose a mobile based software token that is supposed to replace the existing hardware based

tokens and computer-based tokens.

The system consists of two parts –

- Software installed on Client's Mobile Phone.
- Server Software.

In this system, the One Time Password (OTP) would be generated without connecting the client to the server. Here Time Synchronous Authentication has been implemented.

The Server would be running separately and the user would be generating the password based on some of the factors which are unique to him. The Password so generated would be valid for only certain duration of time. After that, it will expire automatically. And after the expiry of the token no login from that password and to login again, the user has to generate another new password.

It is a free strong authentication solution for java capable mobile devices like phones or PDAs. The Client Mobile Phone must be java enabled (Nokia, Siemens, Motorola, Sony, BlackBerry, etc.) because the Client side software is developed in Java 2 Micro Edition (J2ME). And the Server is also implemented in a Java 2 Enterprise Edition (J2EE) using Servlets and Java Server Pages.

A. One Time Password

In order to make the system more and more secure, the One Time Password algorithm must be such that it cannot be guessed and should be safe from all sort of hacking attacks.

Several factors can be guessed in order to generate the OTP such as user's mobile number but these factors can be easily guessed and hence it would not be secure.

Secondly these factors must be common to both the server and the client so that the password generated from the factors at the client side must be matched with the password generated by the server using those factors only. Since it is time synchronous so the following factors are chosen:

1) Epoch Time

Here the current epoch time is generated in a 10 second granularity. The Server and Client clock must be synchronized then only it would work. Epoch Time would ensure that the password generated would be unique every time.

2) PIN

The second factor is PIN (Personal Identity Number) which would be unique for every user. Hence it would also serve the purpose of two factor authentication. This PIN for each user would be stored in the database at the server side.

3) Init-Secret

The third factor is Initialization-Secret (Init-Secret). It is also unique to every user. It is a 16-hex-digit secret that has been created when the device was initialized. It would also be stored in the database at the server side for each user. The Init- Secret is not known to the user.

Authentication is based on two factors: a PIN known by the user and the Init-Secret stored on the mobile device. To compensate time differences, the server will accept passwords from 3 minutes in the past to 3 minutes in the future. The time should be synchronized with the client and the server to ensure the correct password generated each time. The above factors are taken together and they are hashed using RFC1321- the MD-5 Message Digest Algorithm [7].

B. Client Design

A J2ME program has been developed and installed on the mobile phone to generate the One Time Password. The program has an Interactive GUI so that the user can easily handle this. The Software has been developed using Eclipse as Integrated Development Environment (IDE) and it can run on any Java enabled handset.

Firstly Load the MIDlet on the devices you plan to use. Installation of the .jar and .jad file is vendor specific. Usually java enabled phones come with some kind of application installer for PCs that allows to install MIDlets over IrDA or serial cable.

When the MIDlet is installed, run it. Now enter the PIN to generate one time passwords, but to use them you will need to initialize the device first and write the Init-Secret into the appropriate user-record on the authentication server. To initialize the token, press 0000. Enter an arbitrary sequence of 25 keys as a random seed. The Init-Secret that will be shown is not to be written down anywhere else but the server itself.

It cannot be displayed again. If 0000 is pressed any time later, the initialization string will change, i.e. the device will be re-initialized. The initialization of a device should always be done by the administrator of the authentication server, not the user himself. A user does not need to know the Init-Secret.

Fig. 1. Mobile screen displaying 8 digit pass code and init- secret code

The user is allowed to enter only his PIN. If he tries to enter some other PIN, an error message would be displayed giving unauthorized access. When the user enters the PIN, the system concatenates the PIN, the Init-Secret and the current epoch time and hashes it with RFC1321 MD5 Message Digest Algorithm.

Hence the hackers have to know both the PIN and Init Secret in order to steal and crack the password. And the One Time Password generated through the Software Token would be valid for 60 seconds after which it expires. That is, just after generation of password, the user must login within 60 seconds otherwise that password would expire and would be of no use. Then he has to again generate the password.

C. Database Design

Here the database should be maintained at the server side.

A database design is needed at the server side in order to store the information such as the Username, PIN and Init Secret corresponding to each user.

The database will not be used to store the password as it would not be secure but it would be used by the Server to generate the password as the request comes to the server. Hence the OTP algorithm would not be traced.

D. Server Design

A server would be implemented in the Organization in order to generate the One Time Password (OTP). The server consists of the database as described in the above section. The Server is implemented using Java 2 Enterprise Edition (J2EE) architecture. Initially the user has to register them at the Organization and the Administrator will provide the User his (PIN) and his Initialization Secret which he would store in the Database at the Server and secondly he would provide the mobile application to the user and the Init-Secret would also be initialized on his handset and the handset would be synchronized with the Server Clock.

So whenever the user would enter his PIN, the OTP would be generated with the MD5 hash of the PIN, Init Secret stored on the device and current epoch time. This password would be valid only for 60 seconds after which it expires. Now the user has to generate another password. Whenever the request comes for the Password, the Server generates the password for that particular PIN and Init-Secret for the previous 60 seconds. If the password is matched from one of those, the user is granted the access to the System. The Init-Secret is not known to the user that is just used by the user mobile handset to generate the OTP and at the server side to generate the password.

And the Administrator can also generate the 16 hex digits as the Init Secret (which is unique to every user) with this system from the server side.

And in case the mobile phone is lost, the user has to inform the Organization and then his account would be disabled and hence the hacker/attacker would not be able to break the security system. And if the hacker comes to know even about the Init Secret by any chance, then also he does not know the PIN of the user without which he cannot generate the OTP. Hence it would be secure.

IV. System Testing

The Server was implemented using Java 2 Enterprise Edition (J2EE) and the client side was developed using Java 2 Micro Edition and the application was installed on Nokia 5300 phone. And the SQL Server 2000 was used as the database on the Server Side.

The experiment was done to check the chances of getting same identical hash of two people. The database was filled with 10 fake users' information with the PIN unique to each of them and the 10 different Init-Secrets were also filled. The generated OTP was 8 characters long so that the brute force attacks cannot also guess the OTP which can consist of any letter or number.

The 10, 00,000 Passwords were generated for each user for 10, 00,000 seconds of time and none of the single password matched with each other. And the 10, 00,000 Passwords were also generated with different PINs of 10 fake users also and correspondingly no password matched with one other. All OTPs were unique.

V. Conclusion

Now a days, using static passwords, as it is commonly done for accessing the confidential data and information is no more considered secure and safe. Hence Two Factor Authentication becomes more and more popular. But on the other hand, there are disadvantages of T-FA also. The drawback of strong authentication is that every user has to be provided with a token device. This can be quite expensive. And hence the cost for the organization increases as it has to provide its every user with the hardware token for authentication which could be very expensive as to provide the million users the token is not really practical.

Fortunately mobile phones that are capable of running java applets are becoming more and more widely spread. It stands to reason to use your mobile phone as an authentication token. Hence Mobile Phones are a good means of software tokens for using two factor authentications.

This paper focuses on two factor authentication implementation using mobile phones. The method has been tested and found secure since it involves the factors that are difficult to guess and hack.

References

[1] E. Valente, 2009, Two-Factor Authentication [Online] http://www.sans.org/reading_room/whitepapers/authentication/two-factor-authentication-choose-one_33093Federal

[2] Financial Institutions Examination Council (2008). [Online] "Authentication in an Internet Banking Environment" Available:

[3] http://www.ffiec.gov/pdf/authentication_guidance.pdf

[4] RSA SecurID Authentication, [Online] http://www.opsec.com/solutions/partners/downloads/rsa_securid_roi.pdf

[5] Mudge and Kingpin (January 2001), Initial Cryptanalysis of the RSA SecurID Algorithm [Online] Available: http://www.comms.engg.susx.ac.uk/fft/crypto/initial_securid_analysis.pdf

[6] VASCO Digipass Family of Authentication Devices, [Online] http://www.neocom.pl/dokumenty/VASCO/Whitepaper_Digipass.pdf

[7] M. Leiva-Gomez (December, 2011) PhoneFactor Tightens Authentication in Smartphones and Tablets [Online] Available: http://www.tmcnet.com/topics/articles/242225-phonefactor-tightens-authentication-smartphones-tablets.htm

[8] R. Rivest (April 1992), RFC1321 MD5 Message Digest Algorithm, MIT Laboratory for Computer Science and RSA Data Security, Inc.

[9] http://www.faqs.org/rfcs/rfc1321.html

Manav Singhal is pursuing Integrated Post Graduation in Information and Communication Technology from ABV-Indian Institute of Information Technology and Management, Gwalior. His areas of interests include Mobile computing, Cloud Computing and had won many National awards in Android Application Development. He won Best Application Awards in Social and Business Category in Indian Android Developer Contest 2011, Pune.

Shashikala Tapaswi is Professor in Information Technology Department, Atal Bihari Vajpayee - Indian Institute of Information Technology and Management, Gwalior, India. She has obtained her Ph.D. (Computer Engineering) from Indian Institute of Technology, Roorkee, India, M.Tech (Computer Science) from University of Delhi, India and B.E.(Electronics Engineering) from Jiwaji University, Gwalior, India. Her primary research areas of interest are Artificial Intelligence, Neural Networks, Fuzzy Logic, Digital Image Processing, Computer Networks etc.