Integrating Authentication method on Java Based Mobile Devices

Zahid Iqbal, Sarfraz Alam
zahid.iqbal@kics.edu.pk, sarfraz@kics.edu.pk
Al Khwarizmi Institute of Computer Science, UET, Lahore, Pakistan

Abstract

Strong authentication is the first pillar of trusted networks where identities can be securely shared and trusted across independent partners. This allows the identities to access the network and applications and share the resources. Now-a-days different strong authentication methods are in use some of them can be embedded in security devices such as tokens, smart cards etc. The intent of this paper is twofold: First we study the different authentication methods by keeping in view the Java ME security architecture, then on the basis of study we integrate the HOTP based two factor authentication method in more flexible mobile devices (such as cell phones, PDA) in a cost effective manner. For the illustration purpose, we use a case study of online banking system where users authenticate to the bank server through OTP generated from mobile device to access bank services.

1. Introduction

Services that communicate over an insecure network like the Internet need to prove their identity to each other to access resources. Mostly the services such as web commerce applications used user names and passwords for authentication. But the advancement in the field of information security exposed the weakness of user names and passwords [1]. Therefore, there is a need of robust authentication methods to distinguish the genuine individual from an unauthorized one. One time password (OTP) is certainly one of the simplest and most popular forms of securing the identity of a user [2]. In a common implementation model, the end-user carries an authentication device (called a hard token) that could be a standalone device, such as a smart card or a token. These hard tokens provide a secure and reliable method for identifying users; however they have some significance disadvantages. Hardware tokens are expensive, provide authentication to one application only. Users who wish to gain access to multiple applications such as VPN, web-based services and online banking would be required to carry multiple tokens. People often complaint they do not wish to carry even a single token in addition to ID badges, cell phones and PDA’s they already carry on daily basis. Hardware tokens also suffer from a usability perspective in a number of areas. People tend to forget to carry them and they are frequent lost or broken. Batteries run down, and in case of time synchronized token, time drift between the token and the server can mean a replacement is required. As the need for strong two factor authentication broadens to include consumer-based online banking, e-commerce and authentication to specialized online communities a more universal and inexpensive solution is required. The solution is soft token in form of MIDP application and idea here is to use the mobile device as token generator instead of delivering token to mobile devices. The use of mobile phones as the provider of one time passwords is a logical choice. The mobile phone is rapidly becoming a device that is carried all the time, is less likely to be forgotten at home and is a device that people take great care not to lose. The logic of using a mobile phone as the means to obtain authentication tokens begins to make sense when organization are considering mass deployments. Banks are realized that hard tokens are too expensive. The option of delivering one time passwords to mobile phones is an attractive alternative, especially if there is no need for additional infrastructure required by the bank. The soft token use the standard protocols for authentication and also provides the simple mechanism of provisioning and management.

2. Related Work

A considerable number of research works is conducted on strong authentication methods. NTT Information Platform Sharing Laboratories is giving facility for authentication using multiple communication channels [3] as shown in figure 1. In this solution there is no mechanism to generate OTP directly. It also confront many issues such as device authentication, certificate supported mobile phones and usability (not all the mobile phones have camera to read barcode).

![Figure 1. Simple Challenge Response Authentication using 2-Dimensional Barcode [3]](image)

NordicEdge AB also developed an OTP based authentication solution [4], when the user id and password...
is successfully verified, a "One Time Password" is sent to the user's mailbox or mobile phone through SMS (Short Message Services, also known as mobile text). This "One Time Password" will be verified and only then will the user be authenticated to the application as shown in the figure 2.

![Figure 2. NordicEdge OTP Solution](image)

This solution has some security breaches such as the SMS is sent in plain text, no device authentication as well.

### 3. Authentication Methods

Authentication is the act of proving identity of an entity. The authentication process consists of obtaining the authentication information from an entity, analyzing the data, and determining if it is associated with that entity. This means that the computer must store some information about the entity. An entity can be verified by one or more different factors.

#### 3.1 One Factor Authentication

A factor of authentication is considered one of the three things [5].

1. Something you have such as a hardware token, a USB token
2. Something you know such as a password or PIN number
3. Something you are such as a human being with a fingerprint

Using these categories it is clear that a user ID and password scheme is simple one factor authentication. In this scenario the password is the something you know. It does not involve something you have or something you are. Many attacks and threats are noticed such as phishing, man in the middle attack and dictionary attack against one factor authentication [1]. Due to these security breaches the industry requires stronger authentication method than one factor authentication.

#### 3.2 Two-Factor Authentication

Two-factor authentication is simply any authentication scheme which involves two of the factors listed above. The majority of schemes involve “something you have” and “something you know” which usually involves a USB token or smart card along with a PIN number.

Currently, in market two-factor authentication has four categories [6]:

- **One Time Password (OTP)** authentication can be divided in two types; synchronous (based on a transformation of a common shared secret and a moving value that is synchronous on both the server side and the client side) and challenge response (in which server generates a challenge value that will be transformed by the client based on a secret shared between the client and the server).

- **Challenge-Response** Mechanism authentication is usually based on a shared secret transformation using symmetric key hash techniques. The server side sends the client a challenge. The client uses the challenge as the data input and the shared secret as the key in the transformation. The resulting code is called the Response and is sent back to the server.

- **Public Key Infrastructure** authentication uses public key encryption techniques, supported by a public key infrastructure (PKI) for key and certificate management. Digital certificates are issued by a certificate authority and bind the user’s identity to their public key. In a typical certificate-based authentication protocol, the client uses a private key to sign a challenge from the server, and the server verifies the signature using the client’s certificate along with other PKI-based information provided by the certificate authority.

- **Biometrics** authentication is based on a physiological characteristic of a user, such as a fingerprint, iris image or facial image. Biometric authentication represents the “what you are” component of multi-factor authentication. Biometric authentication is based on data matching of the biometric characteristic of the user.

### 4. HOTP Based Two-Factor Authentication

Federal Information Processing Standards Publication endorsed a new OTP algorithm standard called HMAC-based OTP (HOTP), based on the HMAC SHA-1 algorithm[7]. It is an event-based OTP algorithm, in which a counter value is used in order to calculate HOTP and incremented on the client and server after each use according to eq1.

\[
\text{HOTP}(K, C) = \text{Truncate}(\text{HMAC-SHA-1}(K, C)) \ldots (\text{eq1})
\]

where

/ K = symmetric key know only to the token and the validation service
/ C is the counter.
/ \text{Truncate} represent the function that convert HMAC-SHA-1 value into the OTP value.
Since it is an event based solution, resynchronization of client and server is a critical task and to accomplish this, a resynchronize parameter \((s)\) is used to verify a received authenticator across \((s)\) consecutive counter value. To mitigate denial of service (DoS) a throttling parameter \((T)\) is used on server side which refuses connections from a client after \((T)\) unsuccessful authentication attempts \([2]\). The algorithm can be implemented by any hardware manufacturer or software developer to create interoperable authentication devices and software agents. Therefore, we implement HOTP on JME supported mobile phones.

5. Java ME (JME) Overview

The HOTP is implemented on Java ME supported mobile phones as soft tokens to provide flexibility and cost advantages. In this section we discuss that why we choose Java ME to implement HOTP.

The popularity and presence of wireless communications has been increasing steadily over the past few years. Worldwide omnipresence of third-generation systems that include Java is imminent. Personal devices that host practical applications will become available to the masses. With the proliferation of mobile, wireless and internet enabled devices such as PDAs, cell phones etc; Java is emerging as standard execution environment due to its security, portability, mobility and network support features. The creators of J2ME have defined only two configurations to avoid a fragmented landscape of incompatible platforms. The two configurations that exist currently represent the two categories of pervasive devices there are \([8]\):

- Personal, intermittently connected mobile devices- supported by the Connected, Limited Device Configuration (CLDC)
- Constantly connected network devices-supported by the Connected Device Configuration (CDC)

CLDC platform \([9]\) consists of several components that can be classified into: Virtual Machine, API and tool. The virtual machine is KVM. The APIs are CLDC and MIDP. MIDP \([10]\) is a layer on top of CLDC configuration. It extends the latter with more specific capabilities, namely, networking, graphics, security, application management and persistent storage.

5.1 JME Security

Prior to MIDP 2.0 security was not addressed. In MIDP 2.0 several mechanisms specified. The most are code signing, private storage, secure sockets and Security and Trust Services API for JME \([11]\).

Code Signing - In code signing executable code of an application is signed. Integrity checked at run time. Only the original issuer of the application can replace/update the application. Code signing also prevents Trojans and virus.

Private Storage i MIDP 2.0 specifies Record Management System (RMS) and provides private access to MIDlet. If the MIDlet is signed, no other MIDlet can access the RMS storage.

Secure Socket i MIDP 2.0 specifies secure communication over SSL. This authenticates the server and content securely transmitted over the internet.

J SR -177 Security Trust Services API for JME – This specifies the secure execution of code and secure storage on sim card also specifies cryptographic functions. Highly dependent on operator “sim card policy”.

It is clear from the above discussion that JME and its security features are mature enough for business critical applications.

6. Case Study - Online Banking

The online banking is a highly profitable channel for financial institutions. With online banking people are able to manage their bank accounts locally and remotely. Information about financial institutions, their customers, their accounts, theirs transactions are extremely sensitive. For doing business via public network introduces new challenges for security and trustworthiness. Any online banking system must solve the issue of authentication, confidentiality, integrity and non repudiation which means only authorized customer can access an Internet banking account. Therefore, information viewed private and cannot be modified by third parties and transactions made are traceable and verifiable.

6.1 Online banking challenges

Internet banking systems must authenticate users before granting access to particular services. Only successful authentication eventually enables an authorized user to access his private information. User’s credentials are vulnerable against phishing attack if they are valid for long period of time. These credentials are vulnerable against malicious software such as Trojan horse when stored or entered on a potentially insecure device such as user PC. The most prominent example is static passwords which are assigned once and used repeatedly afterwards. Security hereby is simply based on the assumption that the password is non-trivial and kept secret, which in turn requires a trusted environment in which the password is used. Phishing attack is also common in online bank system where attacker works by hijacking the trusted brands of well-known financial institutions and tricking users into entering their credentials into some faked web form. Man in the middle attack is possible during online communication between user and bank. Instead of trying to get hold of a user’s credentials, messages between the
client PC and the banking server are unnoticeably intercepted, the intruder masquerading as the server to the client and as the client to the server, respectively.

**6.2 Demonstration of Two-Factor authentication using HOTP**

We have developed a demo system which authenticates a user with generated OTP on bank web server. Server side and user side (soft token) architecture are shown in figure 3a and 3b respectively.

![Figure 3a. Soft Token Architecture](image)

![Figure 3b. Server Side Architecture](image)

To implement soft token on mobile devices we use a three layered architecture which consists of:
- Presentation Layer (MIDP form)
- Application Logic Layer (HOTP algorithm)
- Data Access Layer (user credentials storage)

And on bank server side same three layered architecture is used which is:
- Presentation Layer (Servlet/JSP)
- Application Logic Layer (HOTP algorithm)
- Data Access Layer (user credentials storage)

**6.3 Synchronous Key Generator (SKG)**

Synchronous key generator generates and handles symmetric keys preferably for closed environments. It uses the SHA-1 hash algorithm [1] to synchronously generate identical 160-bit keys in physically separated locations. Due to its construction it can easily generate and synchronize keys frequently and is therefore a very dynamic technique. The keys created by System SKG can be used for authentication and encryption. The advantage of this system is that there is no need to transmit the keys over insecure communication channels. To generate a key the input parameter to SKG are seed, iteration count and shared pass phrase (password). In our case, the process is:

1. The Synchronous key generator generates a 160-bit key from the input parameters (counter, iteration count, password).
2. Combine password and input parameters with the previously generated key. After this the key has length double of the previous as shown in figure 4.
3. Repeat the step until we have a key of length 512.

![Figure 4. SKG Process](image)

4. Repeat the whole process iteration times.

**6.4 Authentication Process**
To use the application pre registration with the bank authentication server is required. Client’s credentials can be stored in the bank database. Client will get the jar/jad file of the OTP generator application from authorized bank. To authenticate with bank server user needs to provide his ID. The server will compute HOTP called $S_{HOTP}$ using eq2 and display to the user as shown in figure 5:

$$S_{HOTP} = HOTP(K, C+IMEI+ID) \quad \text{..... (eq2)}$$

where
- $K$ = key generated from user password using SKG
- $C$ = user counter
- IMEI = Register user device IMEI(International Mobile Equipment Identity) uses for device authentication
- ID = user Id

The user will supply $S_{HOTP}$ to MIDlet (soft token) as shown in figure 6 and then application validate this by computing $S_{HOTP}$ using equation 2. If $S_{HOTP}$ is same as $S_{HOTP}$ then the server is authenticated.

After this user will generate HOTP called $U_{HOTP}$ using eq3 and supply this to bank server in order to authenticate.

$$U_{HOTP} = HOTP(K, C+S_{HOTP}+IMEI+ID) \quad \text{..... (eq3)}$$

where
- $K$ = key generated from user password using SKG
- $C$ = user counter
- $S_{HOTP}$ = OTP received from server
- IMEI = Register user device IMEI(International Mobile Equipment Identity) uses for device authentication
- ID = user Id

And server will also compute $U_{HOTP}$ to compare with $U_{HOTP}$. If both are same then the user and device is authenticated as shown in figure 8.

7. Security Analysis

Our solution uses defense in depth principal (mixing a number of authentication techniques) which is the most viable strategy for authenticating the user. This solution provides mutual authentication, device authentication, mitigate man in the middle attack, protection in case of stolen devices and protection in case of illegal transformation of MIDlet. As OTPs are generated run time so the system is not vulnerable to phishing and Trojan horses attack. Mutual authentication of server and user is performed by computing and validating of $S_{HOTP}$ and $U_{HOTP}$. Hence man in the middle attack is also prevented. IMEI guaranteed device authentication. Since
the application is PIN protected therefore in case of device stolen attacker needs to try about 5000 PINs but application is blocked after three unsuccessful attempts. Hopefully the victim will report his stolen device to related authorities before the attacker access the application. Now if the attacker transfer MIDlet from victim mobile device to his mobile device either by infrared or Bluetooth, still there is a protection. Because transformation of MIDlet does not mean transformation of user counter value and attacker would still need the password (hopefully not weak). And also attacker needs to recode(reverse engineering) the MIDlet in order to feed the correct IMEI which is not a simple task. Although there is no 100% security in the world but the collective effect of different authentication techniques provide a fairly high bar for attacker to jump.

Conclusion

We have discussed here a way to use mobile devices as a soft token generator. In demonstration we provide a case study of online banking system and we see that the solution overcome the online banking challenges as we discussed in section 7. This solution is not limited to online banking one can use it in eCommerce and mCommerce applications for strong authentication anywhere from mobile devices. In our future work we are looking to perform formal verification of the proposed solution and also looking to implement multiple authentication methods such as PKI certificate based authentication to support on single mobile device. We will also modify this solution to overcome the Enterprise Interoperable issues so that user has one soft token for multiple applications.

Reference


[4]. NordicEdge, accessed on November 15, 2006 at URL: http://www.nordicedge.se/


[6]. “OATH Reference Architecture release 1.0” accessed on November 15, 2006 at URL: http://www.openauthentication.org/


