One-Time Passwords via SMS

Mohsen Gerami[1]- Satar Ghiasvand[2]

The Faculty of Applied Science of Post and Communications
Danesh Blv, Jenah Ave, Azadi Sqr, Tehran, Iran.
Postal code: 1391637111
e-mail[1]: gerami@ictfaculty.ir
e-mail[2]: satar.rgh@gmail.com

ABSTRACT

SMS-based One-Time Passwords (SMS OTP) were introduced to counter phishing and other attacks against Internet services such as online banking. Today, SMS OTPs are commonly used for authentication and authorization for many different applications. Two-factor authentication provides improved protection, since users are prompted to provide something they know and something they have. This method delivers a higher-level of authentication assurance, which is essential for online banking security. This paper describes a method of implementing two factor authentication using mobile phones. The proposed system involves using a mobile phone as a software token for One Time Password generation.

Keywords: One Time Password, OTP, SMS, Authentication, Mobile

1. INTRODUCTION

SMS-based OTP is one of the most user friendly multi-factor authentication mechanisms today that does not require an additional device. Today, SMS OTPs are commonly used for authentication and authorization for many different applications.

One time passwords, or OTP, are used (as the name indicates) for a single session or transaction. OTP SMS provides a 2 stage security while utilizing Internet Banking. By using a one time password that is sent to your mobile phone in addition to your user ID and static password, you have a high level security.The passwords generated by the OTP SMS are one time passwords. Meaning that the OTP SMS password you have used for one of your transactions can't be used for a second time by you or another person [1].

One-time passwords sent over SMS (text messages) were designed to prevent replay attacks and add an additional layer of log on security. A unique password or code is sent to the user via text, and that code must be entered along with a traditional username and password combination to allow access to a site or authorize a transaction. OTP over SMS is a form of multi-factor authentication. Multi-factor is considered stronger than simple username and password combos because the user must meet: 1) Something you know (i.e. a username/password) and 2) Something you have (the device). In some cases, a third authenticating factor is required.

Multi-factor authentication is not a new concept. For example an ATM requires two-factor authentication: the card as something you have and the PIN as something you know. Many websites, particularly in banking, have recently begun using OTP over SMS [2].

TWO-STEP AUTHENTICATION

The rapid growth in the number of online services leads to an increasing number of different digital identities each user needs to manage. But passwords are perhaps the most common type of credential used today [3]. To avoid the tedious task of remembering difficult passwords, users often behave less securely by using low entropy and weak passwords. Most systems today rely on *static passwords* to verify the user's identity.

However, such passwords come with major management security concerns. Users tend to use easy-to-guess passwords, use the same password in multiple accounts or store them on their machines, etc. Furthermore, hackers have the option of using many techniques to steal passwords such as shoulder surfing, snooping, sniffing, guessing, etc. Moreover passwords can be guessed, forgotten, written down and stolen, eavesdropped or deliberately being told to other people [4].

To better understand this system we start by defining authentication. It is the use of one or more mechanisms to prove that you are who you claim to be. Once the identity of the human or machine is validated, access is granted.

Authentication is the process of verifying the correctness of a claimed identity. It is a way of ensuring that users are who they claim to be when they access systems. Authentication relies on at least one of three types of information: *something you know* (e.g., Password or Pin), *something you have* (e.g., Smartcards or Token), or *something you are* (e.g., a Finger prints or Iris scan, Biometrics) [5].

The traditional system only uses one level of authentication —the humble password. Two-factor authentication requires that two pieces of data be presented, each being from a different category. This dramatically reduces the risk of a system being compromised because the chance of both authentication factors being broken or lost at the same time is minimal [4].

Passwords are known to be one of the easiest targets of hackers. Therefore, most organizations are looking for more secure methods to protect their customers and employees. Biometrics are known to be very secure and are used in special organizations, but they are not used much to secure online transactions or ATM machines given the expensive hardware that is needed to identify the subject and the maintenance costs, etc. Instead, banks and companies are using tokens as a mean of two factor authentication. A security token is a hardware device that is given to authorize user. It is also referred to as an authentication token or a cryptographic token. Tokens come in two formats: hardware and software. Hardware tokens are small devices which are small and can be conveniently carried. Some of these tokens store cryptographic keys or biometric data, while others display a PIN that changes with time. At any particular time when a user wishes to log-in, i.e. authenticate, he uses the PIN displayed on the token in addition to his normal account password. Software tokens are programs that run on computers and provide a PIN that change with time. Such programs implement a One Time Password (OTP) algorithm[6].

2. TYPES OF ONE-TIME PASSWORD

There are basically three types of one-time passwords. The first uses a mathematical algorithm to generate a new password based on the previous password. The second is based on time synchronization between the authentication server and the user providing the password. The third uses a mathematical algorithm, but the new password is based on a challenge and a counter.

OTP generation algorithms typically make use of pseudo randomness or randomness. This is necessary because otherwise it would be easy to predict future OTPs by observing previous ones. Concrete OTP algorithms vary greatly in their details. Various approaches for the generation of OTPs are listed below:

- Based on **time-synchronization** between the authentication server and the client providing the password (OTPs are valid only for a short period of time)
- Using a mathematical **algorithm** to generate a new password **based on the previous password** (OTPs are effectively a chain and must be used in a predefined order).

 Using a mathematical **algorithm** where the new password is **based on a challenge** (e.g., a random number chosen by the authentication server or transaction details) and/or a counter.

There are also different ways to make the user aware of the next OTP to use. Some systems use special electronic security tokens that the user carries and that generate OTPs and show them using a small display. Other systems consist of software that runs on the user's mobile phone. Yet other systems generate OTPs on the server-side and send them to the user using an out-of-band channel such as SMS messaging. Finally, in some systems, OTPs are printed on paper that the user is required to carry [7].

3. FIVE WAYS TO AUTHENTICATE WITH A MOBILE PHONE

To increase the security on the authentication process a hardware authentication device can be used. But instead of using a separate device a mobile phone can be used as the hardware authentication device. In their article “Strong Authentication with mobile phone as a security token”, van Thanh et al [10] display four different solutions of using the mobile phone as a hardware authentication device. There are also many other solutions offered that use the mobile phone as a part of the authentication process. One such solution is ActivIdentity [9]. Below is a list of these five different solutions, which serves as a representative sample of all the different solutions that exist.

1. SMS authentication with Session ID verification

A session ID is sent both to the user’s computer, and is shown in the web browser, as well to the user’s mobile phone. The user then verifies that the session IDs are duplicates and confirms by returning a text message to the sender [10].

2. One-time password from PC to SMS

When the user tries to login, the authentication server generates a challenge which is then sent to the user’s web browser, and moreover an OTP. The person enters the challenge in the mobile phone which has an OTP applet installed. This applet generates an OTP and returns an answer to the authentication server through an SMS. If the answer is correct, i.e. if it matches the first OTP generated, the user is logged in [10].

3. One-time password from SMS to PC

In this solution, when the user tries to login, the authentication server generates and sends an OTP in an SMS to the user’s mobile phone. The user types this OTP into the web browser and is by this authenticated by the authentication server [10].

4. SIM strong authentication via mobile phone

This solution is using the EAP-SIM protocol, which means that the protocol communicates directly to the SIM-card and authenticates the SIM through the international mobile subscriber identity (IMSI). It can be used automatically or manually depending on the Bluetooth availability, see note below [10].

5. Software token in the mobile phone

In this solution a software token application is downloaded to the mobile phone. The token generates OTPs that are used to access the system or service in question. This solution therefore involves manual input of OTPs, but no information is sent via additional channels such as via SMS [11].

Note: Solution 1-4 it is also worth to mention that if the mobile phone and the computer are linked with Bluetooth the user do not need to verify the session IDs; the user only needs to make sure that the Bluetooth connection is working. Otherwise some kind of traffic over the GSM network will take place, either through SMS or data traffic, depending on the solution [12].

4. ONE-TIME PASSWORD ALGORITHM

In existing one-time password algorithm, Java MIDlet is a client application and we assume that this runs in client mobile phones which can be able to receive one time passwords. A MIDlet is an application that uses the Mobile Information Device Profile (MIDP) of the Connected Limited Device Configuration (CLDC) for the Java ME environment.

Typical applications include games running on mobile devices and cell phones which have small graphical displays, simple numeric keypad interfaces and limited network access over HTTP. This whole design describes the two main protocols used by Java MIDlet system. Initially, the user downloads the client (Java MIDlet) to his mobile phone. Then the client executes a protocol to register with both server and a service provider utilizing server system for user authentication. After the successful execution of the activation protocol the user can run the authentication protocol an unlimited number of times [13].

In order to secure the system, the generated OTP must be hard to guess, retrieve, or trace by hackers. Therefore, it's very important to develop a secure OTP generating algorithm. Several factors can be used by the OTP algorithm to generate a difficult-to-guess password. Users seem to be willing to use simple factors such as their mobile number and a PIN for services such as authorizing mobile micropayments. Note that these factors must exist on both the mobile phone and server in order for both sides to generate the same password. In most two-step authentication design, the following factors are chosen:

- *IMEI number*: The term stands for International Mobile Equipment Identity which is unique to each mobile phone allowing each user to be identified by his device. This is accessible on the mobile phone and will be stored in the server's database for each client.
- *IMSI number*: The term stands for International Mobile Subscriber Identity which is a unique number associated with all GSM and Universal Mobile Telecommunications System (UMTS) network mobile phone users. It is stored in the Subscriber Identity Module (SIM) card in the mobile phone. This number will also be stored in the server's database for each client.
- *Username*: Although no longer required because the IMEI will uniquely identify the user anyway. This is used together with the PIN to protect the user in case the mobile phone is stolen.
- *PIN*: This is required to verify that no one other than the user is using the phone to generate the user's OTP.

The PIN together with the username is data that only the user knows so even if the mobile phone is stolen the OTP cannot be generated correctly without knowing the user's PIN. Note that the username and the PIN are never stored in the mobile's memory. They are just used to generate the OTP and discarded immediately after that. In order for the PIN to be hard to guess or brute-forced by the hacker, usually, a minimum of 8-characters long PIN is requested with a mixture of upper- and lower-case characters, digits, and symbols.

- *Hour*: This allows the OTP generated each hour to be unique.
- *Minute*: This would make the OTP generated each minute to be unique; hence the OTP would be valid for one minute only and might be inconvenient to the user. An alternative solution is to only use the first digit of the minute which will make the password valid for ten minutes and will be more convenient for the users, since some users need more than a minute to read and enter the OTP. Note, that the software can modified to allow the administrators to select their preferred OTP validity interval.
- *Day*: Makes the OTP set unique to each day of the week.
- *Year/Month/Date*: Using the last two digits of the year and the date and month makes the OTP unique for that particular date.

The time is retrieved by the client and server from the telecommunication company. This will ensure the correct time synchronization between both sides.

For example, in most OTP algorithm the above factors are concatenated and the result is hashed using SHA-256 which returns a 256 bit message. The message is then XOR-ed with the PIN replicated to 256 characters. The result is then Base64 encoded which yields a 28 character message. The message is then shrunk to an administrator-specified length by breaking it into two halves and XOR-ing the two halves repeatedly. This process results in a password that is unique for a ten minute interval for a specific user. Keeping the password at 28 characters is more secure but more difficult to use by the client, since the user must enter all 28 characters to the online webpage or ATM machine. The shorter the OTP message the easier it is for the user, but also the easier it is to be hacked. The proposed system gives the administrator the advantage of selecting the password's length based on his preference and security needs [6][14].

5. IMPLEMENTATION ISSUES

How the software generates a one-time password:

First the user registers in the system control panel software that is installed on a server. Then the user by pressing the request key (on the software installed on mobile embedded) one-time password request is sent to the server. After a few moments the user request is received by the server and then it will be checked and user authentication process begins.

After approval of the user identity, the server responds to user requests and the user password requested will code by encryption algorithms, and it sent to the user.

Software installed on the user's phone has received one-time password and then decode it and show it to the user.

And the end, the user types the password received in own panel. Server processes the user otp password and in the case the accuracy that allows the user to login to the user’s page.

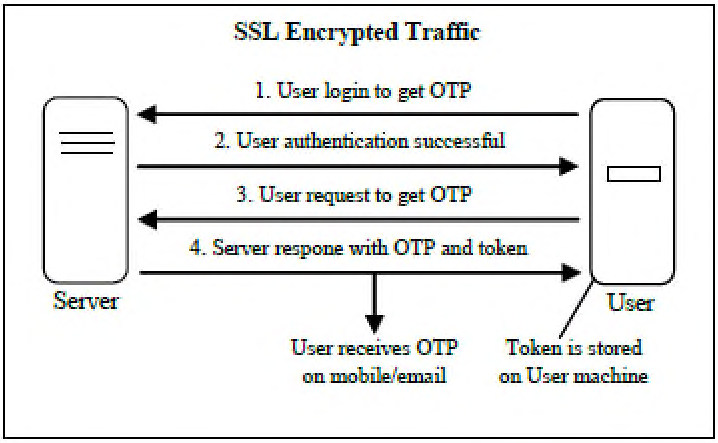
SSL Encrypted Traffic
1. User login to get OTP
2. User authentication successful
3. User request to get OTP
4. Server respone with OTP and token
Server
User
User receives OTP on mobile/email
Token is stored on User machine

Fig. 1. Getting one time password and identifying machine

This sector outlines three application segments used by main program.

1. This section has the task of carrying out cryptographic transaction and key parameter value is achieved through a special encryption algorithms.

```
    public class Encode {

public static String encoded(String str) {
    String result = "";
    for (int i = 0; i < str.length(); i++) {
        char char1 = (char) (str.charAt(i) + key + i);
        String a_letter = Character.toString(char1);
        result += a_letter;
    }
    return result;
}

public static String decoded(String str) {
    String result = "";
    for (int i = 0; i < str.length(); i++) {
      char char1 = (char) (str.charAt(i) - key - i);
      String a_letter = Character.toString(char1);
      result += a_letter;
    }
    return result;
  }
}
```

2 -This section has the task of Registration of the user on the server, which it receives the user registration information and checks.

```
      public class RegisterUserActivity {

  private EditText etName = null;
  private EditText etFamily = null;
  private EditText etTell = null;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.register_user);

    etName = (EditText) findViewById(R.id.editTextName);
    etFamily = (EditText) findViewById(R.id.editTextFam);
    etTell = (EditText) findViewById(R.id.editTextTell);
  }


  public void btnSaveUser(View view) {
    save();
  }

  public void btncancel(View view) {
    RegisterUserActivity.this.finish();
  }

  private void save() {

    try {
      String name = etName.getText().toString();
      String fam = etFamily.getText().toString();
      String tell = etTell.getText().toString();

      if (name.isEmpty()) {
        Toast.makeText(getBaseContext(), getString(R.string.EnterName),
Toast.LENGTH_LONG).show();
        return;
      }

      if (fam.isEmpty()) {
        Toast.makeText(getBaseContext(), getString(R.string.EnterFamily),
Toast.LENGTH_LONG).show();
```

```
                return;
            }

            if (tell.isEmpty()) {
                Toast.makeText(getBaseContext(), getString(R.string.EnterTell),
Toast.LENGTH_LONG).show();
                return;
            }

            SaveData saveData = new SaveData(getBaseContext());

            saveData.saveUser(name,fam,tell);

        } catch (Exception e) {
            e.printStackTrace();
        }


    }

}
```

3. This section has the task of storing all incoming information over the network and during process of software; it puts necessary information to each part.

```
           public class SaveData {

    Context context;

    public SaveData(Context context) {
        this.context = context;
    }

    public void saveUser(String name, String fam,String tell ) {
        try {
            SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(context);
            SharedPreferences.Editor editor = preferences.edit();
            editor.putString( tell + "neme", name);
            editor.putString(tell +  "fam", fam);
            editor.putString(tell , tell);
            editor.apply();
            Toast.makeText(context, context.getString(R.string.userAdded), Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public  UserModel getUser(String tell) {
        UserModel userModel = new UserModel();
        try {
            SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(context);
            String name = preferences.getString(tell + "neme", context.getString(R.string.unknowUser));
        String fam = preferences.getString(tell+"fam", context.getString(R.string.unknowUser));
        String phoneNum = preferences.getString(tell, "0");
        userModel.NAME = name;
        userModel.FAMILY = fam;
        userModel.TELL = phoneNum;
      } catch (Exception e) {
        e.printStackTrace();
      }
      return userModel;
    }
}
```

6. CONCLUSION

In this paper, we have presented an application for one time password generation and transaction between server and mobile handset.

The advantage of this application compared to the similar software is using of highly complex and non-return encryption algorithm, which relationship between user and network security is fully guaranteed and the high flexibility of the software, enabling it to different communication methods such as sms and ussd. For example, all employees of an institution or organization can use this software to connect to networks for a secure and away from any attack. This system is easy to use and is economical in terms of cost.

REFERENCES

[1] OTP SMS, http://www.yapikredi.com.tr/en/limitless-banking/internet-banking/security/otp-sms.aspx

[2] Joe McDonald, (2014), Problems and Vulnerability of One-Time Passwords over SMS

[3] The mobile phone as multi otp device using trusted computing http://eprints.qut.edu.au/37711/

[4] D.Parameswari a, L.JoseSET with SMS OTP, using Two Factor Authentication, Journal of Computer Applications (JCA) ISSN: 0974-1925, Volume IV, Issue 4, 2011

[5] Authentication http://en.wikipedia.org/wiki/Authentication

[6] Sagar Acharya, Apoorva Polawar, P.Y.Pawar, 2013,Two Factor Authentication Using Smartphone Generated One Time Password, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 11, Issue 2 (May. - Jun. 2013), PP 85-90, www.iosrjournals.org

[7] Ms. E.Kalaikavitha M.C.A., M.Phil., Mrs. Juliana gnanaselvi M.Sc., M.Phil., Ph.D.,Secure Login Using Encrypted One Time Password (Otp) and Mobile Based Login Methodology, Research Inventy: International Journal Of Engineering And Science Vol.2, Issue 10 (April 2013), Pp 14-17

[8] A. Vapen and N. Shahmehri. “Security levels for web authentication using mobile phones.” PrimeLife/IFIP Summer School Post-proceedings, Springer, 2011 (In Press).

[9] ActivIdentity. OTP tokens. [Online] Available: http://www.actividentity.com/products/authenticationdevices/OTPTokens/ [2011-04-09]

[10] D. van Thanh, I. Jorstad, T. Jonvik, and D. van Thuan. “Strong authentication with mobile phone as security token.” In Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on, pages 777 - 782, 2009.

[11] ActivIdentity. ActivIdentity_SoftT#5C7F992. [Online] Available: http://www.actividentity.com/download/ document/171 [2011-04-09]

[12] Pernilla Stolpe Johansson,2014,Economic aspects of web authentication

[13] S.Uvaraj, Dr.E.Mohan, (2013),Two Aspect Authentication System Using Secure Mobile Devices, International Journal Of Computer Science And Management Research Vol 2 Issue 6 June 2013,Issn 2278-733x

[14] Stefan Certic,(2013), The Future Of Mobile Security, http://www.cs-networks.net