OTP : Web Authentication Protocol To Restrict Password Stealing And Password Reuse Attacks

S. Renuka
[II-M.Tech]-CSE, Dr.KVSRCEW Kurnool

B. Mahesh
Asst.Prof in CSE Dept.,Dr.KVSRCEW Kurnool

Abstract

Text password is the most popular form of user authentication on websites due to its convenience and simplicity. However, users' passwords are prone to be stolen and compromised under different threats and vulnerabilities. Firstly, users often select weak passwords and reuse the same passwords across different websites. Routinely reusing passwords causes a domino effect; when an adversary compromises one password, she will exploit it to gain access to more websites. Second, typing passwords into untrusted computers suffers password thief threat. An adversary can launch several password stealing attacks to snatch passwords, such as phishing, key loggers and malware. In this paper, we design a user authentication protocol named oPass which leverages a user's cellphone and short message service to thwart password stealing and password reuse attacks. oPass only requires each participating website possesses a unique phone number, and involves a telecommunication service provider in registration and recovery phases. Through oPass, users only need to remember a long-term password for login on all websites. After evaluating the oPass prototype, we believe oPass is efficient and affordable compared with the conventional web authentication mechanisms.

Index Terms—*Network security, password reuse attack, password stealing attack, user authentication.*

1. Introduction

Over the past few decades, text password has been adopted as the primary mean of user authentication for websites. People select their username and text passwords when registering accounts on a website. In order to log into the website successfully, users must recall the selected passwords. Generally, password-based user authentication can resist brute force and dictionary attacks if users select strong passwords to provide sufficient entropy. However, password-based user authentication has a major problem that humans are not experts in memorizing text strings. Thus, most users would choose easy-to-remember passwords (i.e., weak passwords) even if they know the passwords might be unsafe. Another crucial problem is that users tend to reuse passwords across various websites [1], [2]. In 2007, Florencio and Herley [3] indicated that a user reuses a password across 3.9 different websites on average. Password reuse causes users to lose sensitive information stored in different websites if a hacker compromises one of their passwords. This attack is referred to as the password reuse attack. The above problems are caused by the negative influence of human factors. Therefore, it is important to take human factors into consideration when designing a user authentication protocol.

Up to now, researchers have investigated a variety of technology to reduce the negative influence of human factors in the user authentication procedure. Since humans are more adept in remembering graphical passwords than text passwords [4], many graphical password schemes were designed to address human's password recall problem [5]–[9]. Using password management tools is an alternative [10]–[12]. These tools automatically generate strong passwords for each website, which addresses password reuse and password recall problems. The advantage is that users only have to remember a master password to access the management tool. Despite the assistance of these two technologies—graphical password and password management tool—the user authentication system still suffers from some considerable drawbacks.

Although graphical password is a great idea, it is not yet mature enough to be widely implemented in practice, and is still vulnerable to several attacks. Password management tools work well; however, general users doubt its security and thus feel uncomfortable about using it. Furthermore, they have trouble using these tools due to the lack of security knowledge. Besides the password reuse attack, it is also important to consider the effects of password stealing attacks. Adversaries steal or compromise passwords and impersonate users' identities to launch malicious attacks, collect sensitive information, perform unauthorized payment actions, or leak financial secrets. Phishing is the most common and efficient password stealing attack. According to APWG's report, the number of unique phishing websites detected at the second season of 2010 [(Q2, 2010)] is 97 388. Many previous studies have proposed schemes to defend against password stealing attacks.

Some researches focus on three-factor authentication rather than password-based authentication to provide more reliable user authentication. Three-factor authentication depends on what you know (e.g., password), what you have (e.g., token), and who you are (e.g., biometric). To pass the authentication, the user must input a password and provide a pass code generated by the token (e.g., RSA SecureID), and scan her biometric features (e.g., fingerprint or pupil). Three-factor authentication is a comprehensive defense mechanism against password stealing attacks, but it requires comparative high cost.

Thus, two-factor authentication is more attractive and practical than three-factor authentication. Although many banks support two-factor authentication, it still suffers from the negative influence of human factors, such as the password reuse attack. Users have to memorize another four-digit PIN code to work together with the token, for example RSA SecureID. In addition, users easily forget to bring the token. In this paper, we propose a user authentication protocol named oPass which leverages a user's cellphone and short message service (SMS) to prevent password stealing and password reuse attacks. In our opinion, it is difficult to thwart password reuse attacks from any scheme where the users have to remember something. We also state that the main cause of stealing password attacks is when users type passwords to untrusted public computers. Therefore, the main concept of oPass is free users from having to remember or type any passwords into conventional computers for authentication. Unlike generic user authentication, oPass involves a new component, the cellphone, which is used to generate one-time passwords and a new communication channel, SMS, which is used to transmit authentication messages. oPass presents the following advantages.

1. **Anti-malware**—Malware (e.g., key logger) that gather sensitive information from users, especially their passwords are surprisingly common. In oPass, users are able to log into web services without entering passwords on their computers. Thus, malware cannot obtain a user's password from untrusted computers.
2. **Phishing Protection**—Adversaries often launch phishing attacks to steal users' passwords by cheating users when they connect to forged websites. As mentioned above, oPass allows users to successfully log into websites without revealing passwords to computers. Users who adopt oPass are guaranteed to withstand phishing attacks.
3. **Secure Registration and Recovery**—in oPass, SMS is an out-of-band communication interface. oPass cooperates with the telecommunication service provider (TSP) in order to obtain the correct phone numbers of websites and users respectively. SMS aids oPass in establishing a secure channel for message exchange in the registration and recovery phases. Recovery phase is designed to deal with cases where a user loses his cellphone. With the aid of new SIM cards, oPass still works on new cellphones.
4. **Password Reuse Prevention and Weak Password Avoidance**—oPass achieves one-time password approach. The cellphone automatically derives different passwords for each login. That is to say, the password is different during each login. Under this approach, users do not need to remember any password for login. They only keep a long term password for accessing their cellphones, and leave the rest of the work to oPass.

$$\delta_1 = \mathcal{H}^{N-1}(c).$$

5. **Cellphone Protection**—An adversary can steal users' cellphones and try to pass through user authentication. However, the cellphones are protected by a long-term password. The adversary cannot impersonate a legal user to login without being detected.

2. BACKGROUND

oPass adopts the one-time password strategy [28]; therefore, the strategy is given later. We also describe the secure features of SMS channel and explain why SMS can be trusted. Finally, we introduce the security of 3G connection used in the registration and recovery phases of oPass.

2.1 One-Time Password

The one-time passwords in oPass are generated by a secure one-way hash function. With a given input *c*, the set of onetime passwords is established by a hash chain through multiple hashing. Assuming we wish to prepare one-time passwords, the first of these passwords is produced by performing hashes on input *c*.

$$\delta_0 = \mathcal{H}^N(c)$$

The next one-time password is obtained by performing N-1 hashes

Hence, the general formula is given as follows:

$$\delta_i = \mathcal{H}^{N-i}(c).$$

For security reasons, we use these one-time passwords in reverse order, i.e., using δ_{N-1}, then δ_{N-2}……δ_0. If an old one-time password is leaked, the attacker is unable to derive the next one. In other words, she cannot impersonate a legal user without the secret shared credential. Besides, the input is derived from a long-term password (P_u), the identity of server (ID_s), and a random seed (Φ) generated by the server.

$$\delta_1 = \mathcal{H}^{N-1}(c).$$

$$c = \mathcal{H}(P_u \| \mathrm{ID}_s \| \phi).$$

Note that function $\mathcal{H}$ is a hash which is irreversible in general cryptographic assumption. In practice, is realized by SHA-256 in oPass. Therefore, the bit length of is 256.

2.2 SMS Channel

SMS is a text-based communication service of telecommunication systems. oPass leverages SMS to construct a secure user authentication protocol against password stealing attacks. As we know, SMS is a fundamental service of telecom, which belongs to 3GPP standards [30]. SMS represents the most successful data transmission of telecom systems; hence, it is the most widespread mobile service in the world. Besides the above advantages, we chose SMS channel because of its security benefits. Compared with TCP/IP network, the SMS network is a closed platform; hence, it increases the difficulty of internal attacks, e.g., tampering and manipulating attacks. Therefore, SMS is an out-of-band channel that protects the exchange of messages between users and servers. Unlike conventional authentication protocols, users securely transfer sensitive messages to servers without relying on untrusted kiosks.

2.3 3G Connection

3G connection provides data confidentiality of user data and signal data to prevent eavesdropping attacks. It also provides data integrity of signal data to avoid tampering attacks. The confidentiality and integrity algorithms are f8 and f9, respectively. Algorithm f8 and f9 are based on a block cipher named KASUMI where f8 is a synchronous binary stream cipher and f9 is a MAC algorithm. oPass utilizes the security features of 3G connection to develop the convenient account registration and recovery procedures. Users can securely transmit and receive information to the web site through a 3G connection.

3. PROBLEM DEFINITION AND ASSUMPTIONS

In this section, we consider various methods of password stealing. Afterwards, we introduce the architecture of our oPass system and make some reasonable assumptions.

3.1 Problem Definition

People nowadays rely heavily on the Internet since conventional activities or collaborations can be achieved with network services (e.g., web service). Widely deployed web services facilitate and enrich several applications, e.g., online banking, e-commerce, social networks, and cloud computing. But user authentication is only handled by text passwords for most websites. Applying text passwords has several critical disadvantages. First, users create their passwords by themselves. For easy memorization, users tend to choose relatively weak passwords for all websites [2]. This behavior causes a risk of a domino effect due to password reuse [1]. To steal sensitive information on websites for a specific victim (user), an adversary can extract her password through compromising a weak website because she probably reused this password for other websites as well. Second, humans have difficulty remembering complex or meaningless passwords [4]. Some websites generate user passwords as random strings to maintain high entropy, even though users still change their passwords to simple strings to help them recall it. Florencio and Herley [3] indicated that users forget passwords a lot: 1.5% of Yahoo users forget their passwords every month. Some studies pay attention to password management. These approaches could mitigate this problem, but they also make the system more complicated to use. In addition, phishing attacks and malware are threats against password protection. Protecting a user's password on a kiosk is infeasible when key loggers or backdoors are already installed on it. Considering the current mechanisms, authenticating users via passwords is not a comprehensive solution.

Therefore, we proposed a user authentication, called oPass, to thwart the above attacks. The goal of oPass is to prevent users from typing their memorized passwords into kiosks. By adopting one-time passwords, password information is no longer important. A one-time password is expired when the user completes the current session. Different from using Internet channels, oPass leverages SMS and user's cellphones to avoid password stealing attacks. As we mentioned in Section II, we believe SMS is a suitable and secure medium to transmit important information between cellphones and websites. Based on SMS, a user identity is authenticated by websites without inputting any passwords to untrusted kiosks. User password is only used to restrict access on the user's cellphone. In oPass, each user simply memorizes a long-term password for access her cellphone. The long-term password is used to protect the information on the cellphone from a thief.

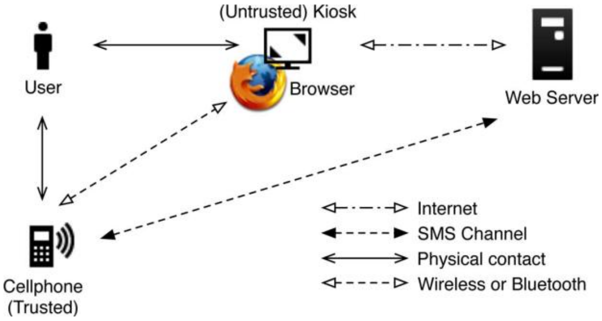


Fig. 1. Architecture of oPass system.

3.2 Architecture of oPass and Its Assumptions

Fig. 1 describes the architecture (and environment) of the oPass system. For users to perform secure login on an untrusted computer (kiosk), oPass consists of a trusted cellphone, a browser on the kiosk, and a web server that users wish to access. The user operates her cellphone and the untrusted computer directly to accomplish secure logins to the web server. The communication between the cellphone and the web server is through the SMS channel. The web browser interacts with the web server via the Internet. In our protocol design, we require the cellphone interact directly with the kiosk. The general approach is to select available interfaces on the cellphone, Wi-Fi or Bluetooth. Section VI will explain the oPass construction in more detail.

The assumptions in oPass system are as follows.

1. Each web server possesses a unique phone number. Via the phone number, users can interact with each website through an SMS channel.
2. The users' cellphones are malware-free. Hence, users can safely input the long-term passwords into cellphones.
3. The telecommunication service provider (TSP) will participate in the registration and recovery phases. The TSP is a bridge between subscribers and web servers. It provides a service for subscribers to perform the registration and recovery progress with each web service. For example, a subscriber inputs her id ID and a web server's id ID to start to execute the registration phase. Then, the TSP forwards the request and the subscriber's phone number (T_u) to the corresponding web server based on the received ID_S.
4. Subscribers (i.e., users) connect to the TSP via 3G connections to protect the transmission.
5. The TSP and the web server establish a secure sockets layer (SSL) tunnel. Via SSL protocol, the TSP can verify the server by its certificate to prevent phishing attacks. With the aid of TSP, the server can receive the correct T_u sent from the subscriber.
6. If a user loses her cellphone, she can notify her TSP to disable her lost SIM card and apply a new card with the same phone number. Therefore, the user can perform the recovery phase using a new cellphone. See Section IV for additional details.

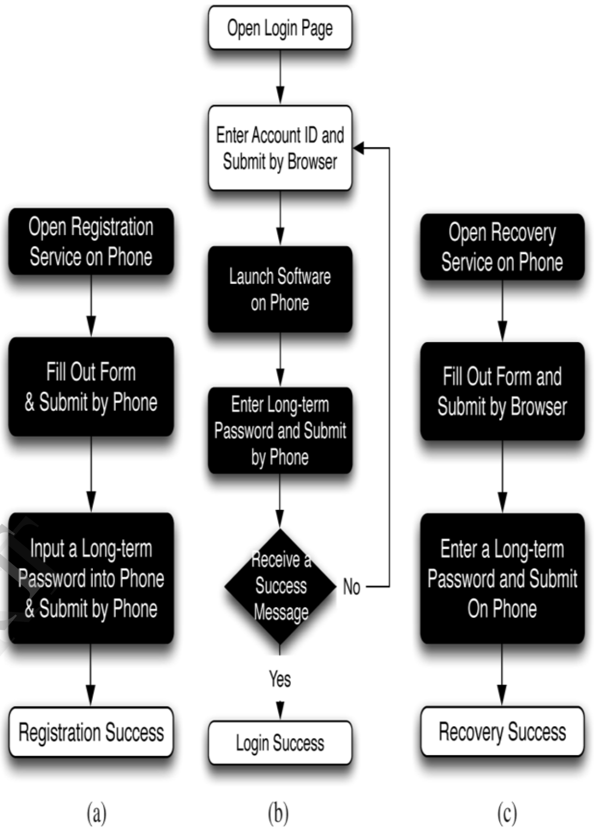

Open Registration Service on Phone
Fill Out Form & Submit by Phone
Input a Long-term Password into Phone & Submit by Phone
Registration Success
(a)
Open Login Page
Enter Account ID and Submit by Browser
Launch Software on Phone
Enter Long-term Password and Submit by Phone
Receive a Success Message
No
Yes
Login Success
(b)
Open Recovery Service on Phone
Fill Out Form and Submit by Browser
Enter a Long-term Password and Submit On Phone
Recovery Success
(c)

Figure 2. Operation flows for user in each phase of oPass system respectively. Black rectangles indicate extra steps contrasted with the generic authentication system: (a) Registration, (b) Login, and (c) Recovery.

4. oPASS

In this section, we present oPass from the user perspective to show operation flows. oPass consists of *registration*, *login*, and *recovery* phases. We introduce the details of these three phases respectively.

4.1 Overview

Fig. 2 describes the operation flows of users during each phase of oPass. Unlike generic web logins, oPass utilizes a user's cell phone as an authentication token and SMS as a secure channel. Different from regular login processes, additional steps are required for oPass and are marked in back rectangles in Fig. 2. In the registration phase, a user starts the oPass program to register her new account on the website she wishes to visit in the future. Unlike conventional registration, the server requests for the user’s account id and phone number, instead of password. After filling out the registration form, the program asks the user to setup a long-term password. This long-term password is used to generate a chain of one-time passwords for further logins on the target server. Then, the program automatically sends a registration SMS message to the server for completing the registration procedure. The context of the registration SMS is encrypted to provide data confidentiality. oPass also designed a recovery phase to fix problems in some conditions, such as losing one’s cell phone.

Contrasting with general cases, login procedure in oPass does not require users to type passwords into an untrusted web browser. The user name is the only information input to the browser. Next, the user opens the oPass program on her phone and enters the long-term password; the program will generate a one-time password and send a login SMS securely to the server. The login SMS is encrypted by the one-time password.

Finally, the cell phone receives a response message from the server and shows a success message on her screen if the server is able to verify her identity. The message is used to ensure that the website is a legal website, and not a phishing one. Protocol details of each phase are provided as follows. Table I shows the notations used in the oPass system.

Table 1: Notations

Name	Description
ID_x	Identity of entity x.
T_y	Entity y’s phone number.
ϕ	random seed
N	Pre-define length of hash chain ($\{\delta_0 \sim \delta_{N-1}\}$).
n_z	Nonce generated by entity z.
P_u	User u’s long-term password.
K_{sd}	Shared secret key between cellphone and the server.
c	Secret shared credential between cellphone and the server.
δ_i	i^{th} one-time password.
$\|\|$	concatenate operation.
$\{\ \}_k$	symmetric encryption[1] with key k.
$\mathcal{H}(\circ)$	Hash function $\mathcal{H}^2$ with input $\circ$.
IV	Initialization vector of AES-CBC.
$HMAC_1$	The HMAC-SHA1 digest of $ID_u\|\|IV\|\|\{c\|\|\phi\}_{K_{sd}}$ under the K_{sd}.
$HMAC_2$	The HMAC-SHA1 digest of $ID_u\|\|IV\|\|\{n_d\|\|n_s\}_{\delta_i}$ under the δ_i.
$HMAC_3$	The HMAC-SHA1 digest of $ID_u\|\|IV\|\|\{c\|\|n_s\}_{\delta_{i+1}}$ under the δ_{i+1}.

[1] Symmetric encryption algorithm in oPass is AES-256.

[2] Hash function is SHA-256.

4.2 Registration Phase

Fig. 3 depicts the registration phase. The aim of this phase is to allow a user and a server to negotiate a shared secret to authenticate succeeding logins for this user. The user begins by opening the oPass program installed on her cell phone. She enters ID (account id she prefers) and ID (usually the website url or domain name) to the program. The mobile program sends ID and ID to the telecommunication service provider (TSP) through a 3G connection to make a request of registration. Once the TSP received the ID and the ID, it can trace the user’s phone number T_u based on user’s SIM card. The TSP also plays the role of third-party to distribute a shared key K_{sd} is used to encrypt the registration SMS with AES-CBC. The TSP and the server will establish an SSL tunnel to protect the communication. Then the TSP forwards ID_u, T_u and K_{sd} to the assigned server *S*. Server *S* will generate the corresponding information for this account and reply a response, including server’s identity ID_s, a random seed Φ, and server’s phone number T_s. The TSP then forwards ID_s,Φ, T_s and a shared key K_{sd} to the user’s cellphone. Once reception of the response is finished, the user continues to setup a long-term password P_u with her cellphone. The cellphone computes a secret credential by the following operation:

$$c = \mathcal{H}(P_u \| \mathrm{ID}_s \| \phi).$$

To prepare a secure registration SMS, the cellphone encrypts the computed credential *c* with the key K_{sd} and generates the corresponding MAC, i.e., $HMAC_1$. HMAC-SHA1 takes input user’s identity, cipher text, and IV to output the MAC. Then, the cellphone sends an encrypted registration SMS to the server by phone number T_s as follows:

$$\text{Cellphone} \overset{SMS}{\rightarrow} S : \mathrm{ID}_u, \{c \| \phi\}_{K_{sd}}, IV, \\ \mathrm{HMAC}_1.$$

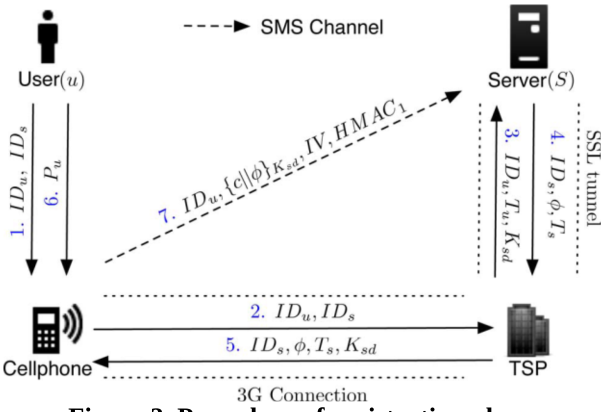


Figure 3. Procedure of registration phase

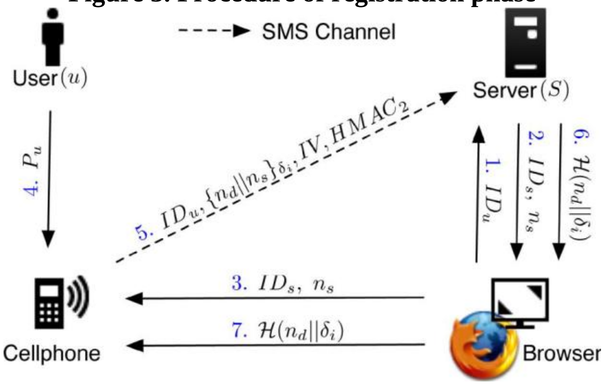

SMS Channel
User(u)
Server(S)
4. P_u
5. $ID_u, \{n_d\|n_s\}_{\delta_i}, IV, HMAC_2$
1. ID_u
2. ID_s, n_s
6. $\mathcal{H}(n_d\|\delta_i)$
3. ID_s, n_s
7. $\mathcal{H}(n_d\|\delta_i)$
Cellphone
Browser

Figure 4. Procedure of login phase

Server can decrypt and verify the authenticity of the registration SMS and then obtain *c* with the shared key K_{sd}. Server *S* also compares the source of received SMS with T_u to prevent SMS spoofing attacks. At the end of registration, the cellphone stores all information {ID_s, T_s,Φ,i}, except for the long term password P_u and the secret *c*. Variable *i* indicates the current index of the one-time password and is initially set to 0. With *i*, the server can authenticate the user device during each login. After receiving the message (6), the server stores {ID_u, T_u,*c*,*Φ*,i}

4.3 Login Phase

The *login* phase begins when the user sends a request to the server *S* through an untrusted browser (on a kiosk). The user uses her cellphone to produce a one-time password, e.g., δ_i, and deliver necessary information encrypted with δ_i to server *S* via an SMS message. Based on pre shared secret credential *c*, server *S* can verify and authenticate user *u* based on δ_i. Fig. 4 shows the detail flows of the login phase.

The protocol starts when user wishes to log into her favorite web server *S* (already registered). However, *u* begins the login procedure by accessing the desired website via a browser on an untrusted kiosk. The browser sends a request to *S* with *u*'s account ID_u. Next, server *S* supplies the ID_u and a fresh nonce n_s to the browser. Meanwhile, this message is forwarded to the cellphone through bluetooth or wireless interfaces. After reception of the message, the cellphone inquires related information from its database via ID_u, which includes server's phone number T_s and other parameters {*Φ*,i} . The next step is promoting a dialog for her long-term password P_u. Secret shared credential can regenerate by inputting the correct P_u on the cellphone. The one-time password δ_i for current login is recomputed using the following operations:

$$c = \mathcal{H}(P_u\|\mathrm{ID}_s\|\phi).$$

$$\delta_i = \mathcal{H}^{N-i}(c).$$

δ_i is only used for this login (*i*th login after user registered) and is regarded as a secret key with AES-CBC. The cellphone generates a fresh nonce n_d. To prepare a secure login SMS, the cellphone encrypts n_d and n_s with δ_i and generates the corresponding MAC, i.e., $HMAC_2$. The next action on the cellphone is sending the following SMS message to server *S*:

$$\text{Cellphone} \overset{\text{SMS}}{\rightarrow} \boldsymbol{S} : \mathrm{ID}_{\boldsymbol{u}}, \{\boldsymbol{n_d}\|\boldsymbol{n_s}\}_{\boldsymbol{\delta_i}}, \boldsymbol{IV}, \mathrm{HMAC}_{\mathbf{2}}.$$

After receiving the login SMS, the server recomputes δ_i (i.e., $\delta_i = \mathcal{H}^{N-i}(c)$.) to decrypt and verify the authenticity of the login SMS. If the received n_s equal the previously generated n_s, the user is legitimate; otherwise, the server will reject this login request. Upon successful verification, the server sends back a success message through the Internet, $\mathcal{H}(n_d\|\delta_i)$, , to the user device. The cellphone will verify the received message to ensure the completion of the login procedure. The last verification on the cellphone is used to prevent the phishing attacks and the man-in-the-middle attacks. If the verification failed, the user knows the failure of login, and the device would not increase the index *i*. If the user is successfully log into the server, index is able to automatically increased, *i=i+1*, in both the device and the server for synchronization of one-time password. After *N-1* rounds, the user and the server can reset their random seed *Φ* by the *recovery* phase to refresh the one-time password.

4.4 Recovery Phase

Recovery phase is designated for some specific conditions; for example, a user may lose her cellphone. The protocol is able to recover oPass setting on her new cellphone assuming she still uses the same phone number (apply a new SIM card with old phone number).

Once user *u* installs the oPass program on her new cellphone, she can launch the program to send a recovery request with her account ID_u and requested server ID_s to predefined TSP through a 3G connection. As we mentioned before, IDs can be the domain name or URL link of server *S*. Similar to registration, TSP can trace her phone number T_u based on her SIM card and forward her account ID_u and the T_u to server S through an SSL tunnel. Once server S receives the request, S probes the account information in its database to confirm if account *u* is registered or not. If account ID_u exists, the information used to compute the secret credential will be fetched and be sent back to the user. The server S generates a fresh nonce n_s and replies a message which consists of ID_s,Ø,T_s,*i*. and n_s . This message includes all necessary elements for generating the next one-time passwords to the user *u*.

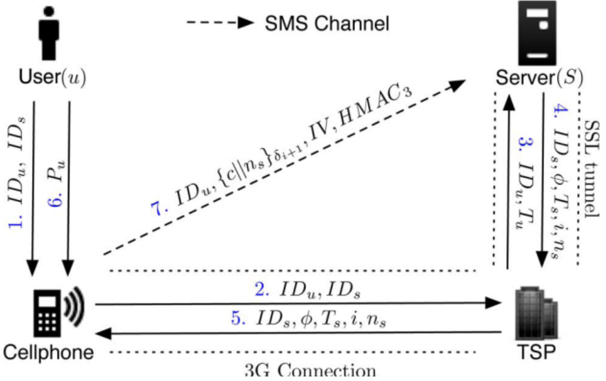


Figure 5. Procedure of recovery phase.

When the mobile program receives the message, like registration, it forces the user to enter her long-term password to reproduce the correct one-time password δ_{i-1} (assuming the last successful login before lost her cellphone is δ_i). During the last step, the user's cellphone encrypts the secret credential *c* and server nonce n_s to a cipher text. The recovery SMS message is delivered back to the server S for checking. Similarly, the server S computes δ_{i-1} and decrypts this message to ensure that user *u* is already recovered. At this point, her new cellphone is recovered and ready to perform further logins. For the next login, one-time password δ_{i+2} will be used for user authentication. Fig. 5 shows the detail flows of *recovery* phase.

5. SECURITY ANALYSIS

Protecting user credentials on ubiquitous web access kiosks is important since they are located everywhere, such as airport lounges, hotel business centers, and cafes. All sorts of attacks may happen in such settings, including key logger, malware, and phishing. Hence, we define a threat model of oPass and demonstrate that oPass is secure later. Furthermore, we examine oPass by a cryptographic protocol verifier, ProVerif, to guarantees the necessary security features claimed by oPass.

5.1 Threat Model

In our setting, attackers can intercept, eavesdrop, and manipulate any message transmitted over the Internet and other wireless networks. The attacker can also spoof an SMS to cheat websites. The computer which a user uses to log into websites is considered untrusted. Attackers can install malwares and setup a backdoor to collect a user's sensitive information (e.g., passwords). The attacker's goal is to masquerade itself as a legitimate user and to gain access to websites without being detected. The methods of gaining access can be classified into two categories based on the attacker's targets which are user and web server.

1. **User side**—In this category, the malicious threat is originated from user side. Password stealing is an effective method to complete the attacker's goal. The attacker can launch phishing attacks, such as phishing websites and phishing emails, to swindle passwords out of users. A user's computer probably installs some malwares (e.g., key logger and Trojan horses). Furthermore, as we mentioned earlier, users always choose weak passwords which are vulnerable to dictionary attacks. Users also suffer from password reuse attacks. In oPass, the attacker can steal a user's cellphone to log on websites.
2. **Server side**—The malicious threat in this category is different from user side. Attackers exploit vulnerabilities of oPass to pass the authentication without being detected by the websites. For example, the attacker can intercept and manipulate messages to launch reply and SMS spoofing attacks.

We assume that attackers cannot break basic cryptographic primitives. For instance, the cipher text cannot be decrypted without the corresponding secret key, and hash function *H*(o)is irreversible. Sections VI–IX will demonstrate how oPass resists the malicious threats mentioned above in the different phases respectively.

5.2 Attacks on Registration

The main task of the registration phase is to generate a shared credential for computing one-time passwords between users and websites. The shared credential should be kept secret to guard oPass from attacks. To prove the guaranteed secrecy of credential in the registration phase, we translate our desired feature and system settings to the Horn clauses. Then we verify our aim through performing ProVerif with those clauses. The registration phase, which consists of seven messages, requires 26 rules to be defined. ProVerif assumes that an attacker can intercept every message (includes SMS) between the cellphone, the TSP, and the server. Meanwhile, we hypothesize that the attacker does not know the session key of 3G connection; the attacker also does not know the session key of SSL tunnel. We also make the assumption that the attacker cannot obtain the long-term password because it is directly typed into the malware-free cellphone by the user. ProVerif was able to confirm that the attacker cannot derive the secret credential. Although we only present the result of registration by using ProVerif, the other two phases, login and recovery, provide the same level of security.

5.3 Attacks on Login

In the login protocol, the attacker can launch attacks to masquerade itself as a legitimate user without being detected. The attacker has no way of obtaining a one-time password for login even if he builds a spoof website to launch a phishing attack because the secret key for encryption and is never transmitted. The attacker also cannot recover the encrypted login SMS. Hence, phishing attacks do not work under oPass. In oPass, users type their accounts into the kiosk and type their long-term passwords into the cellphones. A kiosk that is installed with malwares or key loggers to snatch user passwords is also useless. oPass achieves a one-time password approach to prevent against password reuse attacks. If an attacker steals a user's cellphone and attempts to log into a website that the victim has visited, he will not succeed because he does not know the user's long-term password, so he cannot generate a legal one-time password for the next round. Even if the attacker interrupts the login procedure after obtaining the login SMS, login will still fail, because the nonce generated by the server does not match. In order to launch a MITM attack, an attacker must fully control (i.e., interception, eavesdropping, and manipulation) all transmission channels. Suppose the attacker launches an MITM attack between the server and the browser (see Fig. 4).

5.4 Attacks on Recovery

Potential threat in the recovery protocol is whether an attacker who stole a user's cellphone can succeed in guessing the correct long-term password. This attack is referred to as the password guessing attack. The attacker may try to guess the user's to compute the one-time password for login. He only has to masquerade as a normal user and execute the recovery procedure. After receiving the message in Step 5) from the TSP, the attacker enters a guessed and computes a candidate one-time password. The attacker then transmits a login SMS to the server. However, the attacker has no information to confirm whether or not the candidate is correct. Therefore, the protocol prevents a password guessing attack.

5.4 Further Discussion

1. **Issues of Phone Number Authenticity:** Phone number is a critical factor of oPass since we adopt the SMS channel for message exchanging. The potential issue is how users ensure that the phone number received is actually from the desired website rather than a forged phishing website. To address this difficulty, *registration* and *recovery* phases involve a telecommunication service provider (TSP). We assume that TSP provides a service (e.g., cellphone application) to support registration and recovery procedures of oPass. Users input the identity of the desired websites (e.g., Facebook) to the TSP's service. TSP will establish an SSL tunnel with the website before forwarding messages sent from users to it. Based on the SSL protocol, TSP can verify the website's certificate to prevent phishing attacks. Therefore, we can ensure that the phone number is actually from the correct website rather than phishing websites. In addition, the SSL tunnel provides data confidentiality. The communication interface between cellphone and TSP is 3G. 3G provides data confidentiality to protect the messages exchange. Hence, the secret key can be securely distributed by the TSP to both the cellphone and the server for registration use. A malicious user cannot decrypt other users' registration SMS unless he compromised there. This mechanism guards against insider attacks. Another potential issue about the phone number is that websites may change their telecom service provider.
2. **One-Time Password Refreshment:** The hash chain of a one-time password will be consumed entirely. We introduce parameter to solve this problem. The server checks the status of hash chain after receiving a legal login SMS. If the rest of the one-time passwords are less than , the server sends a new seed to the cellphone at Steps 6) and 7) of the login procedure Once the cell phone gets the new seed, it computes a new credential and sends it to the server through the SMS channel. Hence, the user and the server will use the new hash chain for the next login. This facility can be automatically completed without user effort.
3. **Resistance to Phishing Attacks:** Although we setup a reasonable assumption: user cellphones should be malware-free, the long-term password still suffers from phishing attack by means of a browser in the cellphone. Via social engineering, the user might input his long-term password into a malicious web site through the cellphone's browser. Even though an attacker can obtain, oPass is still secure since the attacker has no enough information to recompute the credential. Message is only transmitted by the server in the registration and recovery phases; most important of all, the transmission through SSL tunnel and 3G

connection ensures data confidentiality and privacy.

4. **Password Reuse:** Password reuse is a serious problem in the present user authentication systems. To repair this problem, oPass adopts an OTP approach. Even if the long-term password is used for every account, the OTP approach still ensures that all logins are independent. Based on the design, is one of inputs to compute the credential. Ideally, different web servers randomize different to compute distinct. Then distinct derives distinct OTP sequence for login. Therefore, oPass users do not reuse same passwords for different accounts since generated OTP sequences guarantee randomness for each login. *5) Weak Password:* Regarding the weak password problem, users tend to pick weak passwords because the passwords are easy to remember. In oPass system, a user just remembers a long-term password for all accounts. Unfortunately, user behavior is not easy to change.

6. Experiment Design

We implemented a prototype of oPass and conducted a user study to analyze the performance and usability.

6.1 Prototype Implementation

To make the progress smooth, the user only has to key in her long-term password and select a website. Then the remaining operations would perform by the program through clicking a button. All required interactions are eliminated to ensure oPass's efficiency. Currently, a fully functional extension has been implemented on Firefox browsers. Users installed the extension on the browser in a kiosk. The major purpose of the browser extension on kiosks is allowing forwarding data from the web server to the user's cellphone during the login phase. While the user attempts to login on a predefined website, the extension automatically sets up a TCP server socket. After the user clicks the login button on her cellphone, a connection could be built and forward data from the website to the cellphone and vice versa. In the web server implementation, a server program which consists of main server codes (PHP) and setup scripts for database (MYSQL). Server program can be installed and performed on an Apache HTTP server. On the other hand, capacity of sending/receiving SMS via a GSM modem relies on an open source library SMS Lib. For simulating TSP, partial PHP codes and related information were also established by the database.

6.2 User Study

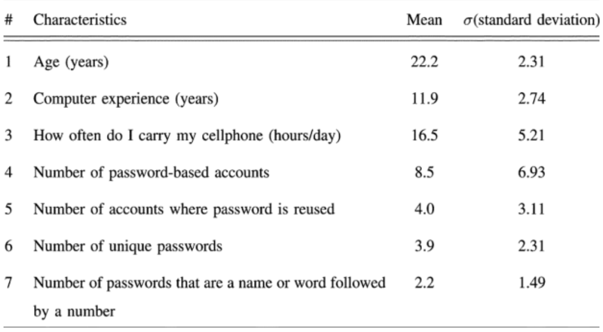
However, most of them were not familiar with the use of a smart phone, especially typing on phones. Participants completed individual tests which consisted of three processes that included setting up, registering, logging in. Before starting the study, participants were first asked to complete a demographics questionnaire. They were then introduced to the oPass system. They were told that they would be setting up the system, registering an account, and logging in via a cellphone. Further, they were instructed to choose a strong long-term password that should be at least eight digits long. Participants completed one practice test (not included in the analysis data) to ensure that they understood how to operate the system. They then proceeded to complete a formal test which consisted of the following steps.

1. Setting up the system: Different from the ordinary user authentication system, users should install a cellphone soft-ware and a browser extension to setup the oPass system.
2. Registering for an account: Users first open the registration software on the cellphone. Users then fill out a form, which includes an account id, a website's id, and a long-term pass-word, and submit it to the website.
3. Logging into the website: Users first enter their account id into the browser on the kiosk and submit it to the server. Users then type their long-term password into the cell-phone and submit to the server. The login succeeds if a success message is shown on the screen of cellphone. If login fails, participants should try again until they are successful. After the test, the participants also completed a post-test questionnaire in order to collect their opinions.

7. Related work

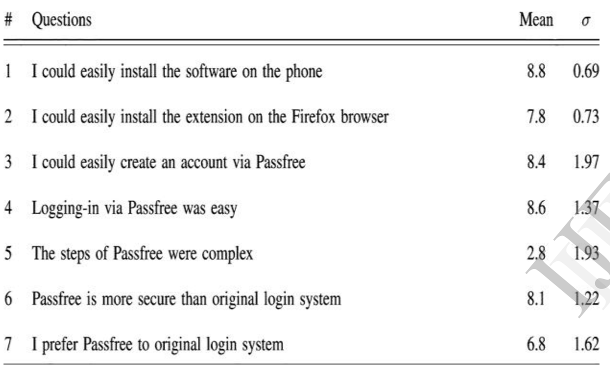
A number of previous researchers have proposed to protect user credentials from phishing attacks in user authentication. The proposed systems leverage variable technologies, for example, mobile devices, trusted platform module (TPM), or public key infrastructure (PKI). However, these solutions were short of considering the negative influence of human factors, such as password reuse and weak password problems. To prevent compromising user credentials, Wuet al .in 2004 proposed an authentication protocol depending on a trusted proxy and user mobile devices. Secure login is authenticated by a token (mobile device) on untrusted computers, e.g., kiosks. To thwart phishing sites, a random session name is sent by SMS from the proxy to the mobile device. The authors declared that security of the proposed system depends on SMS, which are encrypted with A5/1. However, algorithm A5/1 has been broken by Barkan and Biham in 2006. The system is also vulnerable to cellphone theft. On the contrary, oPass encrypts every SMS before sending it out and utilizes a long-term password to protect the cell phone. Another well-known approach is MP-Auth protocol presented by Mannan and Oorschot in 2007. To strengthen password-based authentication in untrusted environments, MP-Auth forces the input of a long-term secret (typically a user’s text password) through a trusted mobile device. Before sending the password to an untrusted kiosk, the password is encrypted by a preinstalled public key on a remote server.

Table 2: Demographic Characteristics (N=24)



#	Characteristics	Mean	σ(standard deviation)
1	Age (years)	22.2	2.31
2	Computer experience (years)	11.9	2.74
3	How often do I carry my cellphone (hours/day)	16.5	5.21
4	Number of password-based accounts	8.5	6.93
5	Number of accounts where password is reused	4.0	3.11
6	Number of unique passwords	3.9	2.31
7	Number of passwords that are a name or word followed by a number	2.2	1.49

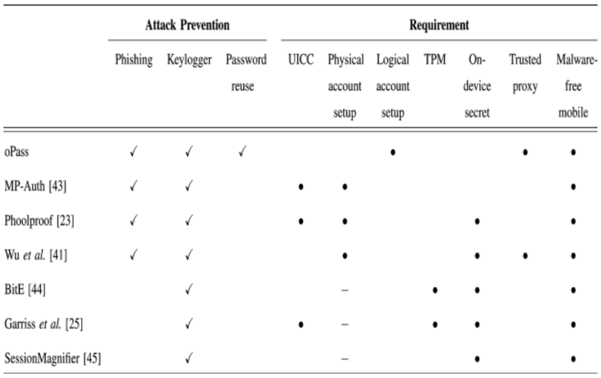
Table 3: Post-test questionnaire. Scores are out of 10.a 10 is most positive



#	Questions	Mean	σ
1	I could easily install the software on the phone	8.8	0.69
2	I could easily install the extension on the Firefox browser	7.8	0.73
3	I could easily create an account via Passfree	8.4	1.97
4	Logging-in via Passfree was easy	8.6	1.37
5	The steps of Passfree were complex	2.8	1.93
6	Passfree is more secure than original login system	8.1	1.22
7	I prefer Passfree to original login system	6.8	1.62

Users should setup an account and password via physical contact, such as banks requiring users to initialize their account personally or send passwords though postal service. In oPass, it addresses above weakness and removes this assumption. OPass achieves one-time password approach to thwart the password reuse problem. Similar to previous works, Parno utilized mobile devices as authentication tokens to build an anti-phishing mechanism is called Phool proof, via mutual authentication between users and websites. To log on the website, a user should provide the issued public key and username/password combination. Again, Phool-proof is still vulnerable to the password reuse problem and needs physical contacts to ensure that account setup is secure. On the other hand, some literature represents different approaches to prevent phishing attacks. Session Magnifier enables an extended browser on a mobile device and a regular browser on a public computer to collaboratively secure a web session. Session Magnifier separates user access to sensitive inter-actions (online banking). For sensitive interactions, the content is sent to the extended browser on the users mobile de-vice for further confirmation from a user. Another avenue is adopting TPM. McCuneet al designed a bump in the ether (BitE) based on TPM [12]. Via BitE, user inputs can be protected under an encrypted tunnel between the mobile device and an application running on a TPM-equipped untrusted computer. To ensure trustworthy computing on kiosks, Garriss et al. invent another system leveraging by TPM and virtual machine (VM) technologies. Table I summarizes comparisons of oPass with previous systems [14].Many of proposed systems require user involvement in certificate confirmation (UICC) in order to setup a secure SSL tunnel. However, prior research concluded that users do not understand SSL and often ignore the SSL warnings caused by illegal certificates [13]. Consequently, users often accept the received certificates without verification. This inattentive behavior causes users to suffer from potential attacks, such as MITM and DNS spoofing attacks. From previous literature, users should pay attention to confirming server certificate validities by themselves. The significant difference between oPass and other related schemes is that oPass reduces the negative impact of user misbehaviors as much as possible. In the oPass system, the SSL tunnel is established between a TSP and a web site server. From the perspective of users, they feel comfortable since there is no further need to verify the server’s certificate by themselves. In other words, overhead on verifying server certificates for users is switching to the TSP. The TSP acts as a users’ agent to validate server certificates and also establish SSL tunnels correctly. Therefore, oPass still resists phishing attacks even if users misbehave. The account setup process is classified into two types: physical and logical setup. MP-Auth, Phool proof, and Wu et al. schemes all assume that users must setup their accounts physically. They establish shared secrets with the server via a secret (conceals) out-of-band channel. For example, banks often re-quire users to setup accounts personally through physical contact or utilize the postal service. Conversely, oPass deployed an alternative approach, logical account setup, which allows users to build their accounts without physical contact with the server. In the oPass system, we invite a TSP in the registration phase to accomplish as the same security as physical account setup

Table 4 : Comparing opass with previous research. Uicc stands for user involvement in certificate confirmation



	Attack Prevention			Requirement						
	Phishing	Keylogger	Password reuse	UICC	Physical account setup	Logical account setup	TPM	On-device secret	Trusted proxy	Malware-free mobile
oPass	✓	✓	✓			•			•	•
MP-Auth [43]	✓	✓		•	•					•
Phoolproof [23]	✓	✓		•	•			•		•
Wu *et al.* [41]	✓	✓			•			•	•	•
BitE [44]		✓			–		•	•		•
Garriss *et al.* [25]		✓		•	–		•	•		•
SessionMagnifier [45]		✓			–			•		•

8. Conclusion

A user authentication protocol named oPass which leverages cell phone and SMS to thwart password stealing and password reuse attacks. oPass assume that each website possesses a unique phone number. It also assume that a telecommunication service provider participants in the registration and recovery phases. The design principle of oPass is to eliminate the negative influence of human factors as much as possible. Through oPass, each user only needs to remember a long-term password which has been used to protect their cell phone. Users are free from typing any passwords into untrusted computers for login on all websites. OPass is efficient for website authentication to prevent phising, key logger and malware. SMS delay could increase the execution time and reduce the performance. The performance of oPass can be improved by Round Robin DNS with the help of simultaneous response from the server for multiple users at a time. Internet relay chat protocol can be used for synchronous conferencing of SMS service. There by communication overhead can be reduced because of many transactions.

References

[1] Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin,‖oPass: A user authentication protocol resistant to password stealing and password reuse attacks,‖ ieee transactions on information forensics and security, vol. 7, no. 2, april 2012

[2] S.Gaw and E.W.Felten,―Password management strategies for online accounts,‖ inSOUPS '06: Proc. 2nd Symp. Usable Privacy. Security, New York, 2006, pp. 44–55, ACM.

[3] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, ―The design and analysis of graphical passwords,‖ inSSYM'99: Proc. 8thConf. USENIX Security Symp, Berkeley, CA, 1999, pp. 1–1, USENIX Association.

[4] J. Thorpe and P. van Oorschot, ―Towards secure design choices for implementing graphical passwords,‖ presented at the 20th. Annu. Computer Security Applicat. Conf, 2004.

[5] S.Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon,―Passpoints: Design and longitudinal evaluation of a graphical pass-word system,‖Int. J. Human-Computer Studies, vol. 63, no. 1–2, pp.102–127, 2005.

[6] B. Pinkas and T. Sander, ―Securing passwords against dictionary at-tacks,‖ in CCS '02: Proc. 9th ACM Conf. Computer Communications Security, New York, 2002, pp. 161–170, ACM.

[7] J. A. Halderman, B. Waters, and E. W. Felten, ―A convenient method for securely managing passwords,‖ inWWW '05: Proc. 14th Int. Conf. World Wide Web, New York, 2005, pp. 471–479, ACM.

[8] K.-P. Yee and K. Sitaker, ―Passpet: Convenient password management and phishing protection,‖ in SOUPS '06: Proc. 2nd Symp. Usable Privacy Security, New York,2006,pp.32–43,ACM.

[9] L. Lamport, ―Password authentication with insecure communication,‖ Commun. ACM, vol. 24, pp. 770–772, Nov. 1981.

[10] H. Krawczyk, ―The order of encryption and authentication for protecting communications (or: How secure is SSL?), ‖ in Advances Cryptology—CRYPTO 2001, 2001, pp. 310–331.

About The Authors

S.RENUKA, received his B.Tech degree in Information Technology from JNTU, Anantapur, India, in 2011. Currently pursuing M.Tech in computer science and engineering at Dr.KVSRCEW Institute of Technology, Kurnool, India.

B.Mahesh, Completed M.Tech(CSE) from JNTUA, Anantapur in 2011. Attended 2 International conferences & 1 National Conference. Area of interest is Network Security and Cloud Computing.