oPass: A Protocol for Prevention of Password Stealing and Password Reuse Attacks

Mr. Navnit Kumar Singh, Mrs. Archana Singh
navnits8@gmail.com, archanasingh0786@yahoo.com

*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, keyloggers and malware. In this thesis, 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.This oPass protocol removes ambiguity of graphical password and password management tool. 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.**

Key Words: Network security, password reuse attack, password stealing attack, user authentication, oPass.

I .INTRODUCTION

Last some time period, text password has been accepted as the primary need 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.

Researchers have investigated a variety of technology to reduce the negative influence of human factors in the user authentication procedure. Since humans can remember graphical password then text password 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 ad-vantage is that users only have to remember a master password to access the management tool.

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, user often selects weak passwords and reuses 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 pass-words 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 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 [26]), 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.

This Paper, design a user authentication protocol named oPass which leverages a user's cell phone 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.

A. Problem Definition

Users' passwords are prone to be stolen and compromised under different threats and vulnerabilities. Also 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. The problem to find the best way to protect and manage passwords. 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. This behavior causes a risk of a domino effect due to password reuse. 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. 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 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. we believe SMS is a suitable and secure medium to transmit

B. Motivation

The study of existing approach showed that one time passwords have also not seen broad acceptance. The difficulty for users of remembering many passwords is obvious. Various Password Management systems over to assist users by having a single sign-on using a master password. Again, the use of these systems does not appear widespread. For a majority of users, it appears that their growing herd of password accounts is maintained using a small collection of passwords.

2. METHODOLOGY

A. Architecture of oPass and Its Assumptions

oPass adopts the one-time password strategy therefore, which is based on method one time password,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.

a. One-Time Password:- The one-time passwords in oPass are generated by a secure one-way hash function. With a given input , 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,

Hence, the general formula is given as follows:

$\delta i \rightarrow H^{n-i}(c)$ where i=0,1,2……n

For security reasons, we use these one-time passwords in reverse order, i.e., using then 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.

*b. 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 . SMS represents the most successful data transmission of telecom systems hence, it is the most wide spread 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.

*c. 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.

B. Architecture of oPass

OPass 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.

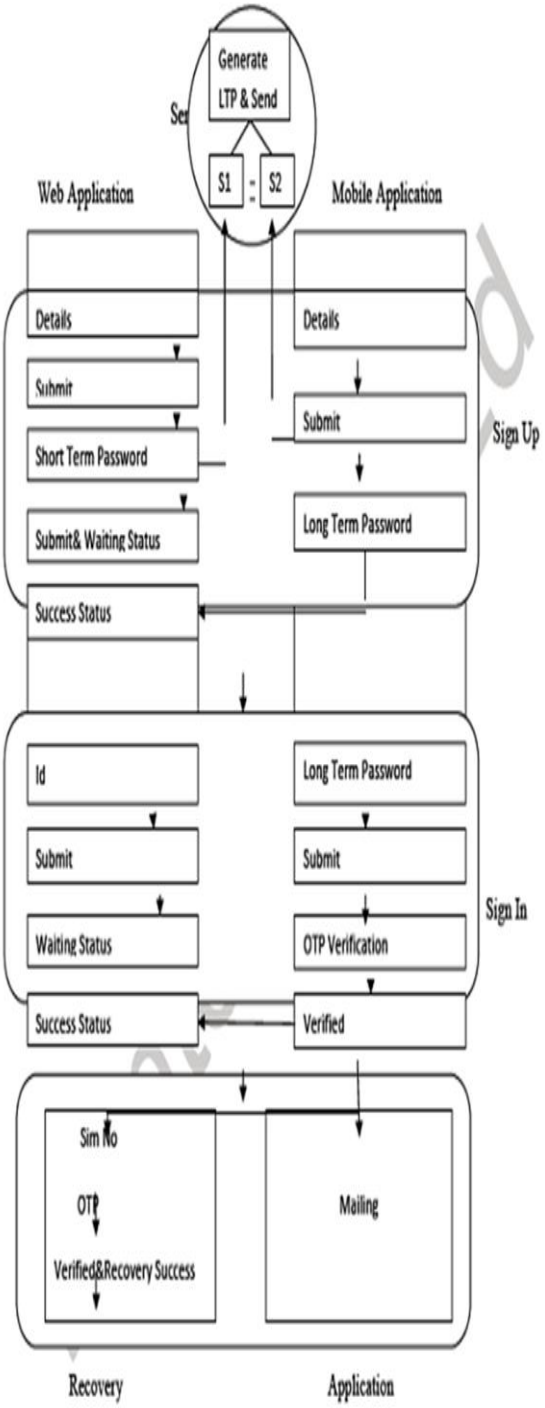


Fig 2.1 Architectue of oPass

In the given figure 2.1 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 UDP 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.

C. Phases of oPass

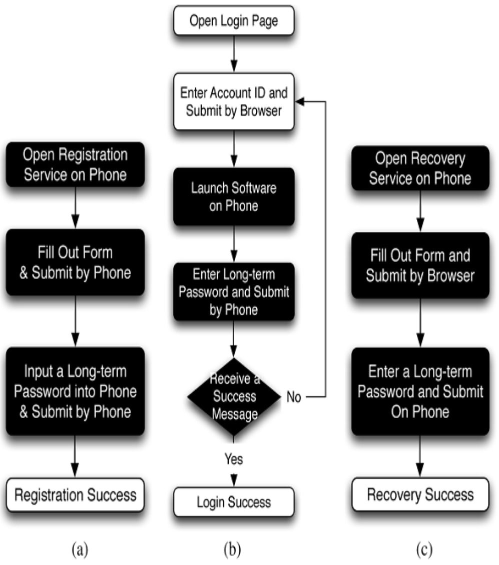

Open Login Page
Enter Account ID and Submit by Browser
Launch Software on Phone
Enter Long-term Password and Submit by Phone
Receive a Success Message
No
Yes
Login Success
Open Registration Service on Phone
Fill Out Form & Submit by Phone
Input a Long-term Password into Phone & Submit by Phone
Registration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

Fig 2.2 oPass Phases

The above figure describes the operation flows of users during each phase of oPass. Unlike generic web logins, oPass utilizes a user's cellphone as an authentication token and UDP as a secure channel. Different from regular login processes, additional steps are required for oPass and are marked in back rectangles in the above Figure. In the *registration* phase, a user starts the oPass program to register her new account on the website he/she wishes to visit in the future. OPass also designed a *recovery* phase to fix problems in some conditions, such as losing one's cellphone.

1) Registration Phase

Fig.2.2 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 cellphone. She enters ID_u (account id she prefers) and ID_s (usually the website URL or domain name) to the program. The mobile program sends ID_u and ID_s to the telecommunication service provider (TSP) through a Internet connection to make a request of registration. Once the TSP received the ID_u and the ID_s, 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} between the user and the server. The shared key K_{sd} is used to encrypt the registration UDP with AES-CBC. The TSP and the server will establish an SSL tunnel to protect the communication. Then the TSP forwards ID, 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, a random seed Ø, and server's phone number T_u. The TSP then forwards ID_s, T_u, Ø, and K_{sd} a shared key 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 C by the following operation: C=H (P_u||ID_s|| Ø).

To prepare a secure registration UDP, 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 [35], [36]. Then, the cellphone sends an encrypted registration UDP to the server by phone number T_s as follows:

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

Server S can decrypt and verify the authenticity of the registration UDP and then obtain C with the shared key K_{sd} .Server S also compares the source of received UDP with T_u to pre-vent UDP spoofing attacks. At the end of registration, the cell-phone stores all information {ID_s, $T_{s,}$ Ø, 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, $T_{u,}$ Ø, c, i} and then completes the registration.

ALGORITHM: Registration Phase
 Input: Server IP Address, User Id, Port no, domain, one time Password.
Output: Registration successful ,one time credential calculated and stored in database.
Step 1: Open Registration service on Android phone.
Step 2:Fill out the form like User ID,IP Address,Domain Name,Port No, and submit by phone.
Step 3:Then input a long term password and submit by phone.
Step 5:Then all the information stored in Server Databse..
Step 6:Registration Successful.

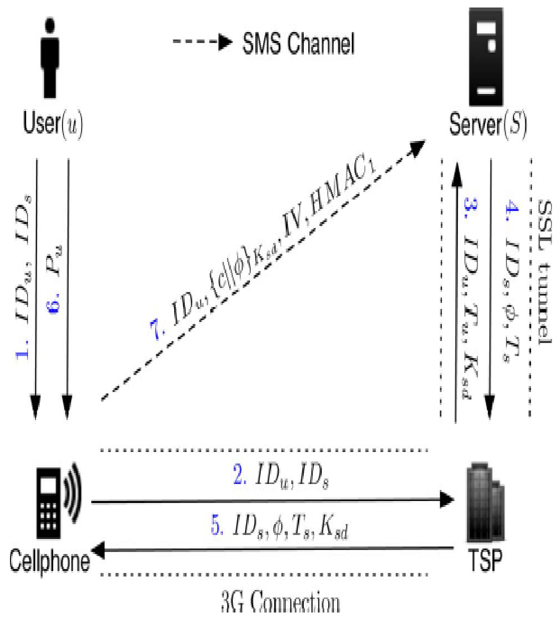


Fig 2.3Registration Phase

2). Login Phase

The login phase begins when the user *u* 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., $§_i$, and deliver necessary information encrypted with $§_i$ to server S via an UDP message. Based on pre-shared secret credential c, server S can verify and authenticate user u based on $§_i$. Fig. 4 shows the detail flows of the login phase.

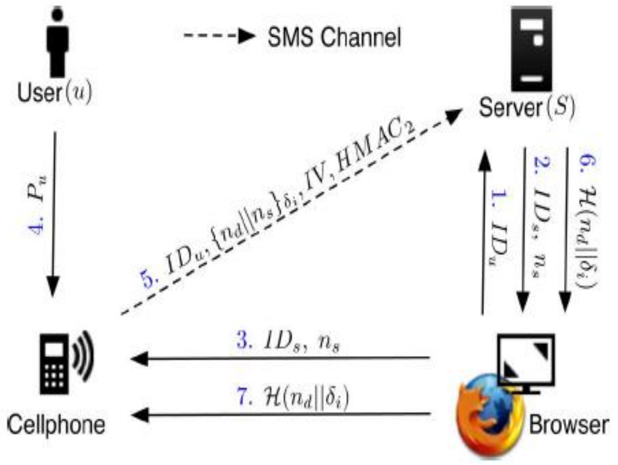

SMS Channel
User(u)
Server(S)
4. P_u
5. $ID_u, \{n_d\|n_s\}_{\delta_i}, IV, HMAC_2$
1. ID_u
2. ID_s, n_s
6. $\mathcal{H}(n_d\|\delta_i)$
3. ID_s, n_s
7. $\mathcal{H}(n_d\|\delta_i)$
Cellphone
Browser

Fig 2.4 Login Phase

The protocol starts when user u wishes to log into her favorite web server S (already registered). However, *we* 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_s and 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_s, which includes server's phone number T_s and other parameters {Ø, i}. The next step is promoting a dialog for her long-term password P_u. Secret shared credential C can regenerate by inputting the correct P_u on the cellphone. The one-time password $§_i$ for current login is recomputed using the following operations.

ALGORITHM: Login Phase.
Input: User name and Password.
Output: successful login into desired domain.
Step 1:open login page.
Step 2:Select the domain name.
Step 3:Enter account ID and Submit by browser.
Step 4:Open software on phone.
Step 5:Enter long term password and submit on phone.
Step 6:Receives a success message.
Step 7:other wise go to step 3.

3). Recovery Phase

Recovery phase is designated for some specific conditions; for example, a user *u* 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).

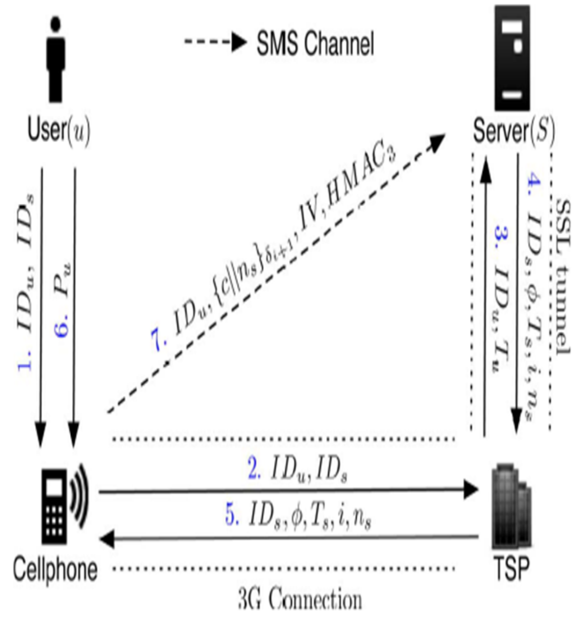


Fig 2.5: Recovery Phase

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, ID_s can be the domain name or URL link of server S. Similar to registration, TSP can trace her phone number T_s 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 C 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*.

When the mobile program receives the message, like registration, it forces the user *u* to enter her long-term -time password $§_{i+1}$ (assuming the last successful login before lost her cellphone is $§_i$). During the last step, the user's cellphone encrypts the secret credential C and server nonce n_s to a cipher text. The recovery UDP message is delivered back to the server S for checking. Similarly, the server S computes $§_{i+1}$ 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 $§_{i+2}$ will be used for user authentication.

ALGORITHM: Recovery Phase

Input:Port no,IP Address, User_ID,Domain,Long Term Password

Output:Recovery Success

Step 1:Open Recovery Service on Phone.

Step 2:Fillout the form and submit by browser.

Step 3:Enter a long term password and submit on phne.

Step 4: Recovery Process Successful.

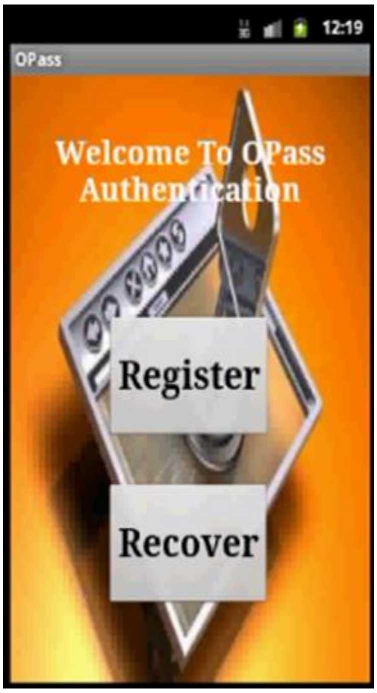
III. RESULTS AND ANALYSIS

We implemented a prototype of oPass according to its three phases. The prototype consists of three components: a mobile program running on Android smart phones (Android OS v2.1); an extension on Firefox browser; and a web server. The server offers a web service by an Apache server running on a workstation with Windows XP, and SMS service with a GSM modem connected to itself. The communication interface between the phone and the browser extension is based on a client/server model over the TCP/IP network. Phones utilizes their WiFi or 3G to connect the TCP server built by the extension. Other mediums, such as Bluetooth and cable line, can substitute for current communication interface. Moreover, we reduced the amount of user interactions and optimize the whole performance in all components.

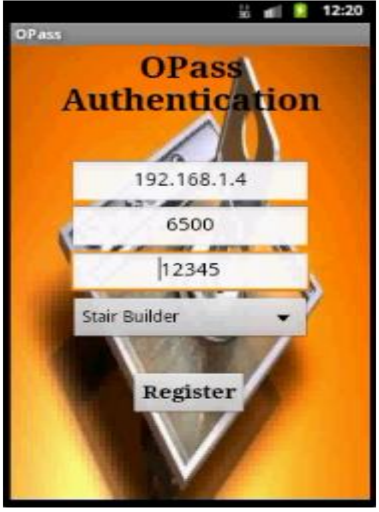
We developed the client program on Android OS due to its popularity and generality. The program has been established and conducted on any android mobile phone. For safety operations, fundamental information of oPass is kept safe in an encrypted SQLite database with as an encryption key. After installing the program, a user creates an account to a website via the registration procedure. Upon successful registration, the user can log into the website. 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 program through clicking a button. All required interactions are eliminated to ensure oPass's efficiency.

In the web server implementation, we developed a server program which consists of main server codes (JAVA) and setup scripts for database (MYSQL). Server program can be installed and performed on an Apache HTTP server.

Step 1:oPass Page on Android mobile



Step 2: Registration Page



OPass Authentication

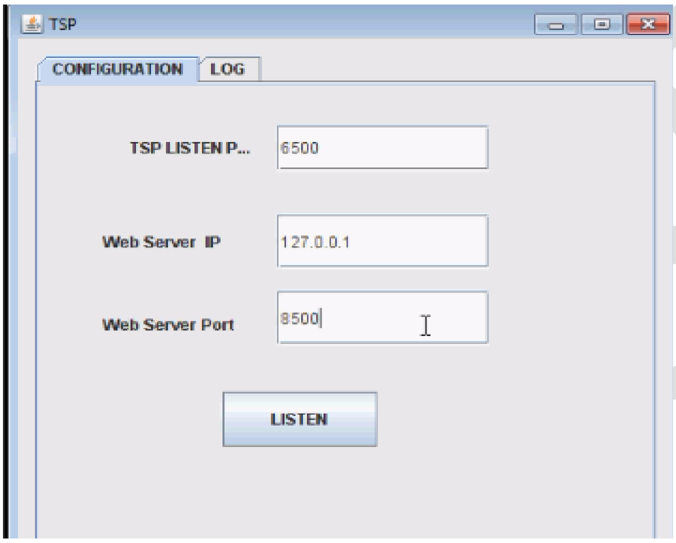
192.168.1.4

6500

12345

Stair Builder

Step 3: TSP Page

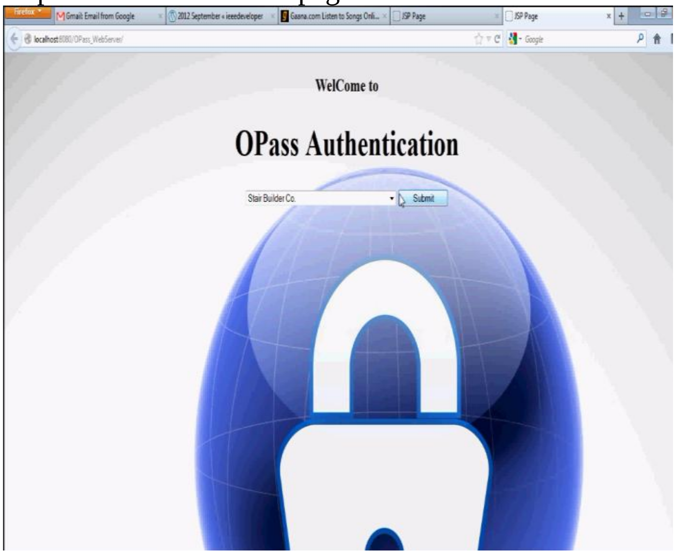


TSP LISTEN P... 6500

Web Server IP 127.0.0.1

Web Server Port 8500

Step 4:oPass Web server page with domain name.

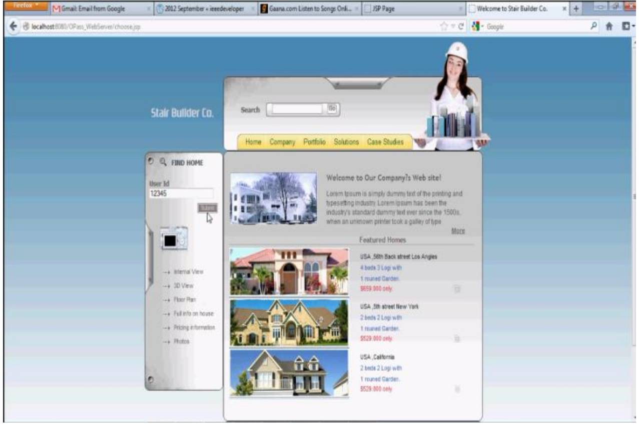


WelCome to

OPass Authentication

Stair Builder Co.

Step 5:Login to domain.



Stair Builder Co.
Search
Home Company Portfolio Solutions Case Studies
FIND HOME
12345
Welcome to Our Company's Web site!
Featured Homes

A. User Study

To analyze the effectiveness and usability of oPass, we conducted a user study with 24 participants, of which eight are female.The average age of the participants was 22. All were university students from distinct departments. Half were computer science students who have knowledge about security and half were not.All were regular computer users, and the average computer experience was 11.9 years. 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 software 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 password, 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 cellphone 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.

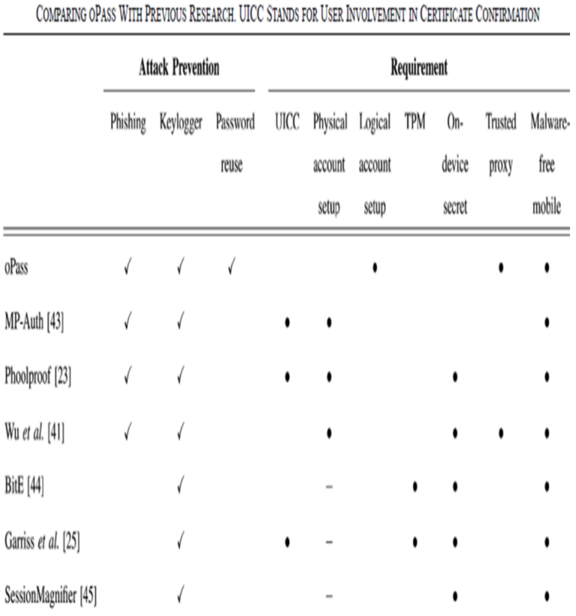
B. *Collected Results*

Our data analysis is used to show the usability of the oPass system and to estimate its performance also we examine the comparison between existing system to oPass system.

To prevent compromising user credentials, Wu *et al.* in 2004 [41] 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 [42]. 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 cellphone.

Another well-known approach is MP-Auth protocol presented by Mannan and Oorschot in 2007 [43]. 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. MP-Auth suffers from password reuse vulnerability. An attacker can compromise a weak server, e.g., a server without security patches, to obtain a victim's password and exploit it to gain his access rights of different websites. On the other hand, MP-Auth assumes that account and password setup is secure. 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, and involves a TSP to ensure that the registration and recovery phases is secure.Parno [23] utilized mobile devices as authentication tokens to build an anti-phishing mechanism, called Phoolproof, 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, Phoolproof is still vulnerable to the password reuse problem and needs physical contacts to ensure that account setup is secure.

Table 3.1 :Comparison between opass and other system.



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]		√			–			•		•

Comparisons Between oPass and Other Systems:

The account setup process is classified into two types: physical and logical setup. MP-Auth, Phoolproof, 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 require 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 assume a TSP in the registration phase to accomplish as the same security as physical account setup. oPass inherits existing trust relations between the TSP and the subscribers (i.e., users) in the telecommunication system. The users' identities were authenticated by the TSP when they applied their cellphone numbers.With this trust relation, users can smoothly setup their accounts via cellphones without physical contact.

REFERENCES

[1] Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin, "The domino effect of password reuse," IEEE Transactions On Information Forensics And Security, Vol. 7, No. 2, April 2012

[1] B. Ives, K. R. Walsh, and H. Schneider, "The domino effect of password reuse," *Commun. ACM*, vol. 47, no. 4, pp. 75–78, 2004.

[2] S. Gawand E. W. Felten, "Password management strategies for online accounts," in *SOUPS '06: Proc. 2nd Symp. Usable Privacy . Security*, New York, 2006, pp. 44–55, ACM.

[3] D. Florencio and C. Herley, "A large-scale study of web password habits," in *WWW '07: Proc. 16th Int. Conf. World Wide Web.*, New York, 2007, pp. 657–666, ACM.

[4] S. Chiasson, A. Forget, E. Stobert, P. C. van Oorschot, and R. Biddle, "Multiple password interference in text passwords and click-based graphical passwords," in *CCS '09: Proc. 16th ACM Conf. Computer Communications Security*, New York, 2009, pp. 500–511, ACM.

[5] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, "The design and analysis of graphical passwords," in *SSYM'99: Proc. 8th Conf. USENIX Security Symp.*, Berkeley, CA, 1999, pp. 1–1, USENIX Association.

[6] A. Perrig and D. Song, "Hash visualization: A new technique to improve real-world security," in *Proc. Int.Workshop Cryptographic Techniques E-Commerce*, Citeseer, 1999, pp. 131–138.

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

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

[9] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, "Design and evaluation of a shoulder-surfing resistant graphical password scheme," in *AVI '06: Proc. Working Conf. Advanced Visual Interfaces*, New York, 2006, pp. 177–184, ACM.

[10] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks," in *CCS '02: Proc. 9th ACM Conf. Computer Communications Security*, New York, 2002, pp. 161–170, ACM.

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

[12] 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.

[13] S. Chiasson, R. Biddle, and P. C. van Oorschot, "A second look at the usability of click-based graphical passwords," in *SOUPS '07: Proc. 3rd Symp. Usable Privacy Security*, New York, 2007, pp. 1–12, ACM.

[14] K. M. Everitt, T. Bragin, J. Fogarty, and T. Kohno, "A comprehensive study of frequency, interference, and training of multiple graphical passwords," in *CHI '09: Proc. 27th Int. Conf. Human Factors Computing Systems*, New York, 2009, pp. 889–898, ACM.

[15] J. Thorpe and P. C. van Oorschot, "Graphical dictionaries and thememorable space of graphical passwords," in *SSYM'04: Proc. 13th Conf. USENIX Security Symp.*, Berkeley, CA, 2004, pp. 10–10, USENIX Association.

[16] J. Thorpe and P. C. van Oorschot, "Human-seeded attacks and exploiting hot-spots in graphical passwords," in *SS'07: Proc. 16th USENIX Security Symp. USENIX Security*, Berkeley, CA, 2007, pp. 1–16, USENIX Association.

[17] P. van Oorschot, A. Salehi-Abari, and J. Thorpe, "Purely automated attacks on passpoints-style graphical passwords," *IEEE Trans. Information Forensics Security*, vol. 5, no. 3, pp. 393–405, Sep. 2010.

[18] R. Dhamija, J. D. Tygar, andM. Hearst, "Why phishing works," in *CHI '06: Proc. SIGCHI Conf. Human Factors Computing Systems*, New York, 2006, pp. 581–590, ACM.

[19] C.Karlof,U. Shankar, J. D.Tygar, andD.Wagner, "Dynamic pharming attacks and locked same-origin policies for web browsers," in *CCS '07: Proc. 14th ACMConf. Computer Communications Security*, NewYork, 2007, pp. 58–71, ACM.

[20] T. Holz, M. Engelberth, and F. Freiling, "Learning more about the underground economy:Acase-study of keyloggers and dropzones," *Proc. Computer Security ESORICS 2009*, pp. 1–18, 2010.

[21] N. Provos, D. Mcnamee, P. Mavrommatis, K. Wang, and N. Modadugu, "The ghost in the browser: Analysis of web-based malware," in *Proc. 1st Conf. Workshop Hot Topics in Understanding Botnets*, Berkeley, CA, 2007.

[22] Phishing Activity Trends Rep., 2nd Quarter/2010 Anti-Phishing Working Group [Online]. Available: http://www.antiphishing.org/ [23] B. Parno, C. Kuo, and A. Perrig, "Phoolproof phishing prevention," *Financial Cryptography Data Security*, pp. 1–19, 2006.

[24] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda, "Panorama: Capturing system-wide information flow for malware detection and analysis," in *CCS '07: Proc.e 14th ACM Conf. Computer Communications Security*, New York, 2007, pp. 116–127, ACM.

[25] S. Garriss, R. Cáceres, S. Berger, R. Sailer, L. van Doorn, and X. Zhang, "Trustworthy and personalized computing on public kiosks," in *Proc. 6th Int. Conf. Mobile Systems, Applications Services*, 2008, pp. 199–210, ACM.

[26] RSA SecureID [Online]. Available: http://www.rsa.com/node.aspx?id=1156/

[27] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proc. IEEE*, vol. 91, no. 12, pp. 2021–2040, Dec. 2003.

[28] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, pp. 770–772, Nov. 1981.

[29] H. Gilbert and H. Handschuh, "Security analysis of SHA-256 and sisters," in *Selected Areas Cryptography*, 2003, pp. 175–193, Springer.

[30] TS 23.040: Technical Realization Short Message Service (SMS) 3GPP [Online]. Available: http://www.3gpp.org/

[31] I. T. Report, ITU Internet Rep. 2006: Digital.Life [Online]. Available: http://www.itu.int/

[32] TS 35.201: Specification 3GPP Confidentiality Integrity Algorithms Document 1: f8 and f9 Specification 3GPP [Online]. Available: http://www.3gpp.org/

[33] TS 35.202: Specification 3GPP Confidentiality Integrity Algorithms Document 2: KASUMI Specification 3GPP [Online]. Available: http://www.3gpp.org/

[34] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell, "Stronger password authentication using browser extensions," in *SSYM'05: Proc. 14th Conf. USENIX Security Symp.*, Berkeley, CA, 2005, pp. 2–2, USENIX Association.

[35] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," *Advances Cryptology—ASIACRYPT 2000*, pp. 531–545, 2000.

[36] 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.

[37] B. Blanchet, ProVerif: Cryptographic Protocol Verifier Formal Model [Online]. Available: http://www.proverif.ens.fr/

[38] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," in *Proc. 14th IEEE Computer Security Foundations Workshop*, 2001, pp. 82–96.

[39] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *Proc. 17th ACM Conf. Computer Communications Security*, New York, 2010, pp. 162–175, ACM.

[40] T. Delenikas *et al.*, SMSLib API—Java Library for Sending/Receiving SMS [Online]. Available: http://smslib.org/

[41] M.Wu, S. Garfinkel, and R. Miller, "Secure web authentication with mobile phones," in *DIMACS Workshop Usable Privacy Security Software*, Citeseer, 2004.

[42] E. Barkan and E. Biham, "Conditional estimators: An effective attack on A5/1," in *Selected Areas in Cryptography*. NewYork:Springer, 2006, pp. 1–19.

[43] M. Mannan and P. van Oorschot, "Using a personal device to strengthen password authentication from an untrusted computer," *Financial Cryptography Data Security*, pp. 88–103, 2007.

[44] J.McCune, A. Perrig, andM. Reiter, "Bump in the ether: A framework for securing sensitive user input," in *USENIX Annu. Tech. Conf.*, 2006, pp. 185–198.

[45] C. Yue and H. Wang, "SessionMagnifier: A simple approach to secure and convenient kiosk browsing," in *Proc. 11th Int. Conf. Ubiquitous Computing*, 2009, pp. 125–134, ACM.

[46] D. Wendlandt, D. G. Andersen, and A. Perrig, "Perspectives: Improving ssh-style host authentication with multi-path probing," in *Proc. USENIX 2008 Annu. Tech. Conf.*, Berkeley, CA, 2008, pp.321–334, USENIX Association.

[47] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer, "Emperor's new security indicators: An evaluation of website authentication and the effect of role playing on usability studies," in *Proc. 2007 IEEE Symp. Security Privacy*, 2007.

[48] R. Biddle, S. Chiasson, and P. van Oorschot, “Graphical passwords: Learning from the first twelve years,” in *ACM Computing Surveys, Carleton Univ.*, 2010.