Volume 4, No. 4, April 2013
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

# PASSWORD KNIGHT SHIELDS PASSWORD STEALING AND RE-USE ATTACK

D Caine[*1], V Radhey Shyam[2] and G Michael[3]
*[1]The Computer Science Department, Bharath University, Chennai, TamilNadu, India
sidharthcaine@gmail.com[1]
[2] The Computer Science Department, Bharath University, Chennai, TamilNadu, India
vradheyshyam1991@gmail.com[2]
[3] The Computer Science Department, Bharath University, Chennai, TamilNadu, India
micmgeo@yahoo.co.in[3]

***Abstract*—** Passwords are the powerful tools that tend to keep all data and information digitally safe. It is often noticed that text password remains predominantly popular over the other formats of passwords, due to the fact that it is simple and convenient. However, text passwords are not always strong enough and are very easily stolen and misused under different vulnerabilities. Others can acquire a text password when a person creates a weak password or a password that is completely reused in many sites. In this condition if one password is stolen, it can be used for all the websites. This is called as the Domino Effect. Another risky environment is when a person enters his/her password in a computer that is not trust-worthy; the password is prone to stealing attacks such as phishing, malware and key loggers etc. In this paper, a user authentication protocol named Password Knight is designed, that makes use of the customer's cellular phone and short message service to ensure protection against password stealing attacks. Password Knight requires a unique phone number that will be possessed by each participating website. The registration and the recovery phases involve a telecommunication service provider.

***Index Terms*—** Phishing, Session Magnifier, Key logging, Malware, Personal Identification Number, Human factors in security, hash visualization, user authentication through image recognition, root key validation.

## INTRODUCTION

Text password has been adopted over the past few decades as the primary means of user authentication for websites. People use their username and text passwords when registering accounts on a website. To be logged into the website successfully, users have to recall the selected passwords. Generally, a password- based user authentication might resist brute force and dictionary attacks if users select strong passwords to provide sufficient entropy. Password-based user authentication, however, as a major problem that humans are not experts in memorizing text strings. Thus, most users would choose easy - to-remember passwords (i.e., passwords) even if they know the passwords might unsafe [1]. Crucial problem is that users tend to reuse passwords across various websites [2] [3]. In 2007, Florencio and Herley [4] indicated that a user reuses a password across a sum of 3.9 different websites on average. Password-reuse can causes users to lose their sensitive information stored in different websites if a hacker compromises one of their passwords. These sort of attacks are usually referred to as password-reuse attack. The above problems are mainly caused by the negative influence of human factors. Therefore, it is very much important to always consider human factors into when designing a user authentication protocol.

## RELATED WORKS

The current security systems mainly suffer from the fact that they always fail to account for considering human factors. There are two main human factors: First, people are very slow and inevitably unreliable when creating meaningless strings; and second, people often have difficulties in remembering very strong passwords or PINs. Validation of root keys in public-key infrastructures and user authentication [14] are the two applications where these human factors negatively affect security. Password hashing can help users manage their multiple accounts by just turning a single memorized password into a different password for each account. A user-assigned site labels (pet names) can help users securely identify the sites in the face of determined attempts at an impersonation (phishing). The password-strengthening measures are defined against the dictionary attacks. Customization of the user interface can defend user-interface spoof-attacks [15].

Various researchers have proposed to protect user credentials from different phishing attacks in user authentication. The proposed systems leverage variable technologies, for example, a mobile device, a trusted platform module (TPM), or any public key infrastructure (PKI). These solutions However, were short of considering the human factors that have negative influence, such as password-reuse and weak-password problems. In order to prevent these compromising user credentials, Wu et al. in 2004 [5] proposed an authentication protocol based on a trusted-proxy and user-mobile devices. Secure login is successfully authenticated by a token (mobile device) on the untrustworthy computers, e.g., kiosks. A random session name is sent by SMS to the mobile device from the proxy in order to prevent phishing sites. It is declared by the authors that the security of the proposed system depends on SMS, that are encrypted with A5/1. The algorithm A5/1, however, has successfully been broken by Barkan and Biham in 2006 [6]. The system also can be vulnerable to cellular phone theft. On the contrary part, Password Knight tends to encrypt each and every SMS before it sends out, and a long-term password is utilized to protect the cellular phone.

MP-Auth protocol presented by Mannan and Oorschot in

2007 [7] is another well-known approach. MP-Auth forces the input of a long-term secret (typically a user's text password) through a trusted mobile device, to strengthen password-based authentication in untrustworthy environment. A pre-installed public key on a remote server encrypts the password, before it sends the password to an untrustworthy kiosk. MP-Auth has a focused intention to guard passwords from various attacks raised by several untrustworthy kiosks that includes key-loggers and malware. However, MP-Auth suffers inevitably from password reuse vulnerability. An attacker can easily compromise a weak server, e.g., a server that has no security patches, will be sufficient for obtaining a victim's password and exploiting it in order to gain his access rights of various websites. MP-Auth, on the contrary, assumes that the account and password set is secure. Users need to setup the account and the password through physical contact, such as the banks that requires the customers to initialize their account personally or sending passwords though postal service. In this, Password Knight addresses the above-mentioned weakness and removes all these assumption. The Password Knight achieves a one-time password approach to prevent the password reuse problem, and involves a TSP in order to ensure that the registration and recovery phases are completely secure.

Similarly, Parno [8], to build an anti-phishing mechanism, utilized mobile devices as authentication tokens in order to build an anti-phishing mechanism, which was called as Phoolproof, through mutual authentication between the users and several websites. To log on into the websites, a user has to provide the pre-issued public-key and the username/password combination. Again, Phoolproof also is still vulnerable to the human influential password reuse problem and thus needs physical contacts in order to ensure that the account setup is secure. On the contrary, some of the literatures represent different approaches to prevent phishing attacks. SessionMagnifier has the ability that tends to enable extended browser on a mobile device along with a regular browser on a public computer that collaborates to secure a web session [9]. SessionMagnifier can separate user access to very sensitive interactions (online banking or payment) from the regular interactions (web surfing or photo viewing). For very sensitive interactions, the content is always sent to the extended browser on the users' mobile device for a further confirmation. Another area is TPM adaption. McCune et al. designed a bump in ether (BitE) based on the TPM [10]. Through BitE, all user inputs will be protected under an encrypted tunnel tha lies between the mobile device and an application running on a TPM-equipped untrustworthy computer. Garriss et al. invent another system leveraging by TPM and virtual machine (VM) technologies [11], to ensure trustworthy computing on kiosks.

In order to setup a secure SSL tunnel, many of the proposed systems require user involvement in the certificate confirmation (UICC). Prior research however concluded that users cannot understand the SSL and are quite often prone to ignore those SSL warnings caused by several illegal certificates [12], [13]. Users consequently, often accept the received certificates without verification. This inattentive behavior can cause users to inevitably suffer from severe potential attacks, such as MITM and DNS spoofing attacks. From previous literature, it is noted that the users should pay attention in confirming server certificate validity at their own risks. The significant difference between Password Knight and many other related schemes is that the Password Knight effectively reduces the negative impact of user misbehaviors to a possible extent. In Password Knight, the SSL tunnel is successfully established between a TSP and the web site server. From the perspective of users, they could feel comfortable as there isn't any further need to verify the servers' certificate by the users them self, i.e., the overhead on a verifying server certificates for users, will be switching to the TSP. The TSP tends to act as a users' agent to validate the server certificates and also will establish the SSL tunnels correctly.

The setup process of an account is classified into two broad types: physical and logical setup. All schemes like the MP-Auth, Phoolproof, and Wu et al. assume that users must setup their accounts physically. Shared secrets are established with the server through a secret (conceals) out-of-band channel. For instance, several banks often require the users to setup accounts personally by physical contact or by the utilization of the postal service. Contrarily, the Password Knight deploys an alternative approach called as logical account setup, in which the users are allowed to build their accounts without any physical contact with the respective server. In this Password Knight system, we are inviting a TSP in the registration phase in order to accomplish the same security as a typical physical account setup. Password Knight tries to inherit the existing trust relations between the TSP and the subscribers (i.e., users) that exist in the telecommunication system. The TSP authenticates the users identities, when they applied their cellular phone numbers. With this very trust relation, users can very well smoothly setup their required accounts with the help of their cellular phones without having to physically contact the server. In the perspective of commercial considerations, it is very much easier to promote a new system if we really could make the system seamless (only requiring a few additional efforts).

A TSP (trusted proxy) is required in Password Knight, to enhance the security. This requirement as we think is reasonable and not costly since the 3G-telecommunication is vastly applied. Considering the performance, only the registration and the recovery phases involves TSP. The above-mentioned two phases would be executed a few times for each use. Thus to conclude, the Password Knight resists most of the attacks and has a very fewer requirements when compared to other systems.

## PROPOSED SYSTEM

We propose a user authentication protocol named Password Knight that leverages a user's cellular phone and short message service (SMS) in order to prevent several password stealing and password reuse attacks. In our perspective, it is quite difficult to prevent the password reuse attacks from any of the schemes where in the users have to remember something. We can also state that the main cause of stealing password attacks is purely when the users type passwords to an untrustworthy public computer. Password Knight also involves a new component, the cell phone, which will be

used to pass passwords and also a new communication channel, SMS, which will be used to transmit the required authentication messages. As far as now, researchers have investigated a variety of technology that reduces the negative influence of the human factors in a user authentication procedure. Due to the fact that humans are much more adept in remembering the graphical passwords than text password, many of the graphical password schemes were designed in order to address the human-password recall problem.

Despite the very assistance of the graphical password and several password management tools—a user authentication system still can suffer from some drawbacks considerably. Though the graphical passwords are a great idea, it is not yet have matured enough to be widely implemented in practice and still is vulnerable to several attacks. Many password management tools also work well; even though, common users can doubt the security of graphical password and thus might feel uncomfortable about using the same. To add up, they also have trouble in using these sorts of tools due to the fact that they lack knowledge of security.

Certain researches mainly focus on a three-factor authentication process rather than password-based authentication in order to provide a more reliable user authentication. The three-factor authentication depends on what do you know (e.g., password), what do you have (e.g., token), and who you are (e.g., biometric). To pass these authentication steps, the user needs to input a password and also has to provide a pass code that is generated by the token (e.g., RSA SecureID), and finally has to scan the user's biometric features (e.g., fingerprint or pupil). The three-factor authentication is a very comprehensive defense mechanism that can work against password stealing attacks, but it surely requires comparative high cost.

Thus, two-factor authentication is far more attractive and quite practical than three-factor authentication. Although many banks support the two-factor authentication, it still also suffers from the negative influence of the human factors, such as password reuse attack. The users have to again memorize another four-digit PIN code in order to work together with the token, for instance RSA SecureID. To add up, users can easily forget to bring in the token.

Unlike the very generic user authentication, Password Knight tends to involve a new component, the cell phone, which is used in generating one-time passwords and a new communication channel, the SMS, which will be used to transmit the authentication messages. The Password Knight has the following advantages.

***Anti-malware:***

A malware (e.g., keylogger) is a program that gathers sensitive information from a user; especially when their password is surprisingly common. In Password Knight, users are able to log into web services without having to enter passwords on their computers. Thus, a malware cannot obtain the user's password from an untrustworthy computer.

***Phishing Protection:***

Many adversaries quite often launch several phishing attacks in order to steal users' passwords by cheating the users when they connect to a forged website. Password Knight allows the users to successfully log into the required websites without having to reveal passwords to computers. The users who adopt Password Knight are guaranteed for withstanding phishing attacks.

***Secure Registration and Recovery:***

SMS is an out-of-band communication interface. Password Knight successfully will cooperate with the telecommunication service provider (TSP) in order to obtain the appropriate phone numbers of websites and that of the users respectively. The SMS tends to aid Password Knight in establishing a very secure channel for exchange of messages in the registration and recovery phases. The recovery phase is designed in order to deal with cases where in a user loses his cell phone. With the help of new SIM cards, Password Knight still can work on new cell phones.

***Password Reuse Prevention and Weak Password Avoidance:***

The Password Knight tends to achieve a one-time password approach. The cellular phone automatically will derive different passwords for each login. The password varies during each login. Under this approach, users do not have the need to remember any password for login. All they have to do is only to keep a long-term password to access their cellular phones, and shall leave the rest of the work to Password Knight.

***Cell phone Protection:***

Some adversary can steal users' cellular phones and try to pass through the user authentication. The cellular phones will be protected by a long-term password during registration. So the adversary will not be able to impersonate a legal user in order to login without being detected.

The Main advantage is that users have to remember only a master password in order to access the management tool.

## ARCHITECTURE DIAGRAM

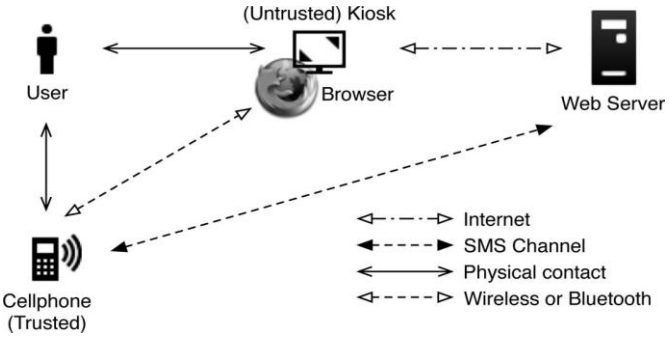



Figure 1. Architecture of Password Knight.

## MODULE DESCRIPTIONS

There are three modules in this system:

a. Registration Phase
b. Login Phase
c. Recovery Phase

***Registration Phase:***

The primary aim of this registration phase is to allow the user and the server in order to negotiate a shared secret so as

to authenticate the succeeding logins for this particular user. The user has to begin by opening the Password Knight program that is installed over the cellular phone. The user then enters the $ID_u$ (account id as of preference) and $ID_s$ (usually the website url or domain name) to the program. The mobile program eventually sends the $ID_u$ and the $ID_s$ to the respective telecommunication service provider (TSP) through a 3G connection in order to make a request for registration. Once the TSP receives the $ID_u$ and the $ID_s$ , it then can trace out the user's phone number $T_u$ that is based on the user's SIMcard. The TSP also will play the role of a third-party in order to distribute a shared key $K_{sd}$ that lies between the user and the server. The shared key $k_{sd}$ is then used to encrypt the registration SMS with the AES-CBC. The TSP and the server S will eventually establish an SSL tunnel in order to protect the communication process. The TSP then forwards the $ID_u$, $T_u$ and $K_{sd}$ on to the assigned server S. The server S then will generate the corresponding information related with this account and replies with a response, that includes the server's identity $ID_s$ , a random seed $\phi$, and server's phone number $T_s$ . The TSP then tends to forward $ID_s$, $\phi$ ,$T_s$ , along with a shared key $K_{sd}$ over to the user's cellphone. As soon as the response is received, the user then continues setting up a strong long-term password $P_u$ with the required cellular phone. The cellular phone computes a secret credential c by the following operation:

$$C=H(P_u||ID_s||\phi).$$

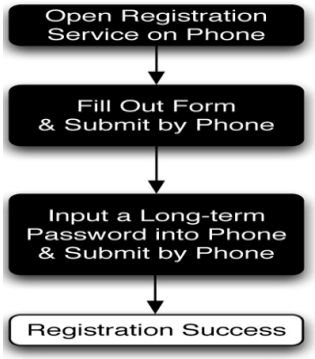
In order to prepare a very secure registration SMS, the cellular phone will encrypt the computed credential c with the key $K_{sd}$ and then will generate the corresponding MAC, i.e., $HMAC_1$ . HMAC-SHA1 which takes input user's identity, ciphertext, and IV to output theMAC. Then, the cellular phone will send an encrypted registration SMS over to the server by the phone number $T_s$ as follows:

$$\text{Cellphone } S: ID_u, \{c||\phi\}k_{sd}, IV, HMAC1.$$

Table 1. Legend

| Name | Description |
|---|---|
| $ID_x$ | Identity of entity $x$. |
| $T_y$ | Entity $y$'s phone number. |
| $\phi$ | 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. |
| $\delta_i$ | $i^{th}$ 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 $\delta_i$. |
| $HMAC_3$ | The HMAC-SHA1 digest of $ID_u\|\|IV\|\|\{c\|\|n_s\}_{\delta_{i+1}}$ under the $\delta_{i+1}$. |

The server S can successfully decrypt and verify the authenticity of the respective registration SMS and then will obtain c with the shared key, $K_{sd}$. Server S also will compare the source of the received SMS with the $T_u$ in order to prevent the different SMS spoofing attacks. At the end of every succesfull registration, the cellphone will store all the information $\{ID_s,T_s,\phi,i\}$, except for the long term password $P_u$ and the secret c. A variable i indicates the current index of the one-time password and is initially set to 0 by default. With the help of i, the server can very successfully authenticate the user-device during each login. After the message is received (6), the server will store $\{ID_u,T_u,c,\phi,i\}$ and then completes the registration.




***Login Phase:***

[18] The login phase begins at the point when the user sends a request to the required server S through an untrustworthy web browser (on a kiosk). The user then uses the registered cellphone in order to produce a one-time password, e.g.,$\delta_i$, and delivers all the necessary information that is encrypted with $\delta_i$ over to the server S through an SMS message. Based on the pre shared secret credential c, the server S can successfully verify and authenticate the user u based on $\delta_i$. The protocol starts immediately when the user *u* wishes to log into her favorite web server S (already registered). However, *u* begins the login process by accessing the desired website through a browser on an untrustworthy kiosk. The browser then sends a request to S with *u*'s account $ID_u$. Next, the server *S* will supply the $ID_s$ and a very fresh nonce *n* to the browser. Meanwhile, this particular message is forwarded to the cellphone via Bluetooth or wireless interfaces. After the message is recieved, the cellphone inquires all the related information from its database through IDs, that includes server's phone number $T_s$ and other parameters $\{\phi,i\}$. The next step is the promotion of a dialog for her long-term password $P_u$. Secret shared credential *c* can regenerated by inputting the correct $P_u$ on the cellphone. The one-time password $\delta_i$ for the current login is successfully recomputed using the following operations:

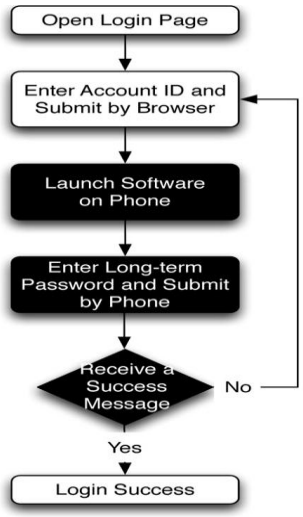
$$C=H(P_u||ID_s||\phi)$$
$$\delta_i=H^{N-i}(c).$$

$\delta_i$ is only just used for this login (*i*th login after user registered) and will be regarded as a secret key with the AES-CBC. The cellphone successfully generates a fresh nonce $n_d$. In order to prepare a secure login SMS, the cellphone will encrypt $n_d$ and $n_s$ with $\delta_i$ and then generates the corresponding MAC, i.e., $HMAC_2$. The next action on

the cellphone is to send the following SMS message to server S:

$$\text{Cellphone}_{(SMS)} \rightarrow \text{S:ID}_u, \{n_d \| n_s\}_{\delta i}, \text{IV}, \text{HMAC}_2.$$

After the login SMS is received, the server once again re-computes $\delta_i$(ie.,$\delta_i = H^{N-i}(c)$) in order to decrypt and verify the authenticity of the respective login SMS. If the received $n_s$ equals to the previously generated $n_s$, the user is legitimate; otherwise, the server will consequently reject this login request. Upon the successful verification, the server will send back a success message via the Internet, $H(n_d \| \delta_i)$, to the user's device. The cellphone then will verify the received message to ensure the successful completion of the login procedure. The last verification on the cellphone will be used in order to prevent the phishing attacks and the "man-in-the-middle" attacks. If the verification fails, the user is notified with the failure of login, and the device might possibly not increase the index *i*. If the user is successfully logged into the server, the index *I* will be able to automatically increase, $i=i+1$, in both the device and the server for synchronization of the one-time password. After the *N*-1 rounds, the user and the server can reset their random seeds ϕ via the *recovery* phase to refresh the one-time password.

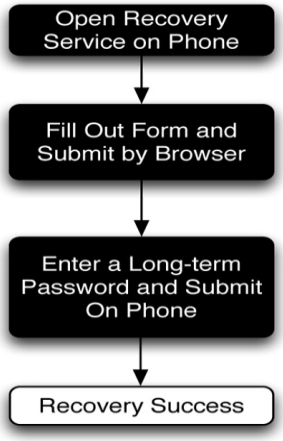



***Recovery Phase:***

The recovery phase is designated for some specific conditions; for instance, a particular user *u* might lose the cellular phone. The protocol is very much able to recover Password Knight setting on the user's new cellphone assuming that the user still uses the same phone number (apply a new SIM card with old phone number).

Once the user *u* installs the Password Knight program on the new cellphone, the user can launch the program in order to send a recovery request with the user's respective account $ID_u$ and requested server $ID_s$ to a predefined TSP through a 3G connection. As mentioned before, the $ID_s$ can be the domain name or a URL link of server *S*. Similar to registration, the TSP can successfully trace the user's phone number $T_u$ that is based on the user's SIM card and forwards the user's account $ID_u$ and the server through an SSL tunnel. As soon as the server *S* receives the request, *S* probes the account information in its database in order to confirm if the account *u* is registered. If the account $ID_u$ exists, the information used to compute the secret credential *c* will be successfully be fetched and will be sent back to the respective user. The server *S* then generates a fresh nonce $n_s$ and replies back with a message which consists of $ID_s$, ϕ, $T_{s,}i$ , and $n_s$. This message does include all the necessary elements that generates the next one-time passwords to the user *u*.

When a mobile program just receives the message, like the registration, it strictly forces the user *u* to enter the desired long-term password in order to reproduce the correct one-time password $\delta_{i+1}$ (assuming the last successful login before lost her cellphone is $\delta_i$). During this last step, the user's cellphone will be encrypted with the secret credential *c* and server nonce $n_s$ to a ciphertext. The recovery SMS message will be delivered back to the particular server *S* for checking. Similarly, the server *S* also computes $\delta_{i+1}$ and then decrypts this particular message in order to ensure that user *u* has already recovered. At this very point, the user's new cellphone is recovered and is ready to perform the further logins. For the very next login, the one-time password $\delta_{i+2}$will be used for the user authentication.




## EVALUATION

A data analysis is used to show the complete reliable usability of the Password Knight system and in order to estimate its performance.

***Usability Evaluation:***

Over 50% of accounts, reuse the same password, which are in different websites. Even more, the data states that half of the participants' passwords are very weak passwords. Skipping all these security risks, all the participants had never adopted any password management tool to protect their accounts unfortunately. This fact appears to be very consistent with our observation about the password reuses and weak password attacks.

All the participants felt that the registration and login processes in the Password Knight system were quite simple and hence was quick. Even more, they also agreed that the Password Knight was even more secure than their original login systems. It is often quite important to make users feel safe and secure. It also demonstrated that our proposed system was very well suited to the users, regardless of background.

***Performance Evaluation:***

As a part of the same study, a performance evaluation was also conducted on the Password Knight prototype. This evaluation consisted a total of 24 participants, each one of them performing one registration and five login processes. In addition with the activity of measuring the total execution time, we also had to measure the SMS delay in the registration and login phases. Due to fact that the operations in registration and recovery phases are quite similar, the experiment only just evaluates the total performance in the registration phase.

The average time for registering is 21.8 s and the time delay in SMS is 9.1 s. Based on this very observation, the SMS overhead will be the major factor in the registration phase (about 41%). As the GSM modem in our evaluation is a cost effective device, we strongly believe that the performance can be improved when we are into utilizing a more powerful GSM modem. Generally, a user has to execute the registration process only once in each website (if the user does not lose the registered cellphone). So, we argue that the overhead of the registration is very much acceptable. Similarly, the overhead of recovery is also acceptable. [17]

## CONCLUSION

A user authentication protocol named Password Knight, which leverages cellphones, and SMS to thwart password stealing and password reuse attacks, is . We assume that each website possesses a unique phone number. We also assume that a telecommunication service provider participates in the registration and recovery phases. The design principle of Password Knight is to eliminate the negative influence of human factors as much as possible. Through Password Knight, each user are only needed to remember a strong long-term password, which tends to protect the user's cellphone. The users are free from typing any passwords into untrusted computers for login on all websites. Compared with previous schemes, Password Knight is the first user authentication protocol to prevent password stealing (i.e., phishing, keylogger, and malware) and password reuse attacks simultaneously.

## 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", in IEEE Transactions On Information Forensics And Security, Vol. 7, No. 2, April 2012.

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

[3]. 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.

[4]. 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.

[5]. M. Wu, S. Garfinkel, and R. Miller, "Secure web authentication with mobile phones," in DIMACS Workshop Usable Privacy Security Soft- ware, Citeseer, 2004.

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

[7]. 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.

[8]. B. Parno, C. Kuo, and A. Perrig, "Phoolproof phishing prevention," Financial Cryptography Data Security, pp. 1–19, 2006.

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

[10]. 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.

[11]. 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.

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

[13]. 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.

[14]. 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.

[15]. A. Perrig and D. Song, "Hash visualization: A new technique to im- prove real-world security," in Proc. Int. Workshop Cryptographic Tech- niques E-Commerce, Citeseer, 1999, pp. 131–138.

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

[17]. Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin "oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks", in IEEE Transactions On Information Forensics And Security, Vol. 7, No. 2, April 2012.

[18]. Ms. R.R.Karthiga and Mr.K.Aravindhan "Enhancing Performance of User Authentication Protocol with Resist to Password Reuse Attacks", International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 8