uses her/his cell phone to generate 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 preshared secret credential *c*, server S can validate and authenticate user based on δ_i. Fig. 4 shows the detail flows of the login phase. The protocol starts when user *u* wishes to log into her favorite web server (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_s and a fresh nonce n to the browser.

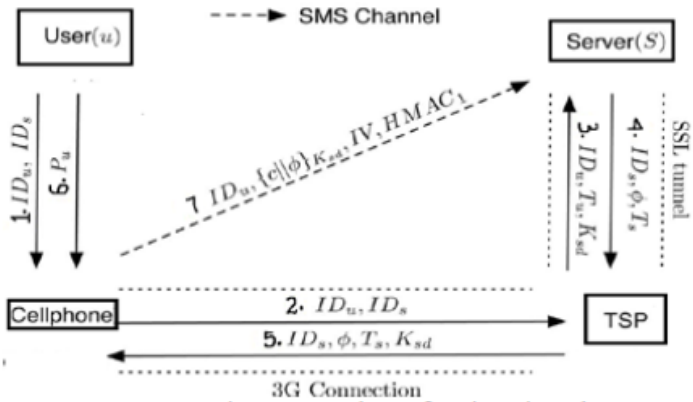

Fig. 3. Procedure of registration phase.

Meanwhile, this message is promoted to the cellphone through bluetooth or wireless interfaces(wifi). After reception of the message, the cellphone findouts related information from its database via ID_s, which comprises server's phone number T_s and other parameters $\{\phi, i\}$. The next promoting a dialog for her long-term password P_u. Secret shared credential *c* can be regenerated by inputting the correct P_u on the cellphone.

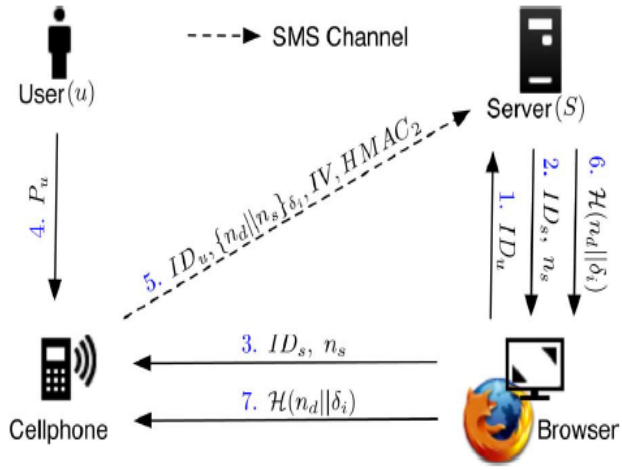

Fig. 4. Procedure of login phase.

The one-time password δ_i for current login is recalculated using the following operations:

$c = (P_u || ID_s || \phi)$. (7)

$= (c)$. (8)

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

Cellphone }, IV, $HMAC_2$ S : ID_u, {. (9)

After receiving the login SMS, the server recomputes δ_i (i.e., δ_{N-1} H^{i} (c))) to decrypt and confirm the authenticity of the login SMS. If the collected n_s equals the previously generated n_s, the user is rightful; if not, the server will reject the login request. Upon successful confirmation, the server sends back a success message through the Internet, $H(n_d||\delta_i)$, to the user device(cellphone). The cellphone will confirm the received message to ensure the completion of the login process. The last verification on the cellphone is used to thwart the phishing attacks and the man-in-the-middle attacks. If the verification is 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, the index is able to automatically increased, $i=i+1$, in both the device and the server for synchronization of one-time pass- word. After N-1 rounds, the user and the server can reset their random seed by the *recovery* phase to refresh the one-time password.

Table 1: shows the Notations of Opass System

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.

4.5 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/his new cellphone assuming she/he still uses the same phone number (apply a new SIM card with old phone number).

Once user set up the OPass program on her/his new cellphone, she/he can initiate the program to send a recovery request with her/his account ID $_u$ and requested server ID $_s$ to predefined TSP through a 3G connection. ID $_s$ can be the domain name or URL link of server S. Once server receives the request, check outs the account information in its database to validate if account 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 generates a fresh nonce and replies a message which consists of ID $_s$, ϕ, T $_s$, *i*, and n $_s$. This message consists of all necessary elements for generating the next one-time passwords to the user *u*.

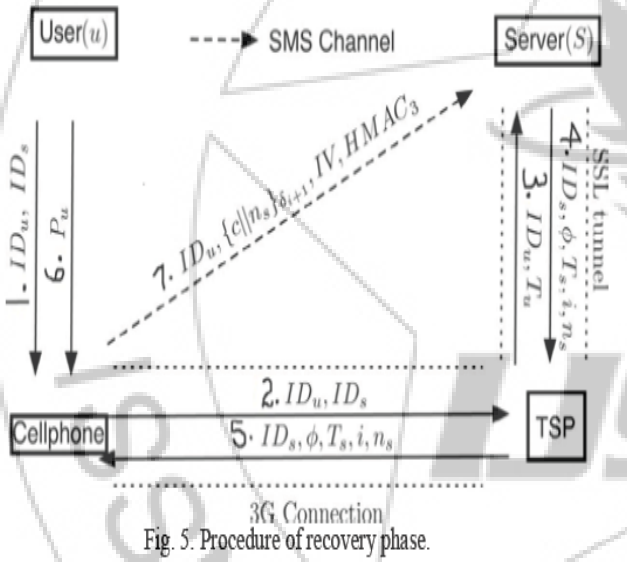

Fig. 5. Procedure of recovery phase.

When the mobile program receives the message, it forces the user *u* to enter her long-term password to reproduce the correct one-time password δ_{i+1}. During the last step, the user's cellphone encrypts the secret credential and server nonce to a cipher text. The recovery SMS message is carried back to the server for checking.

5. Conclusions

OPass which influences cellphones and SMS to thwart password stealing and password reuse attacks. We assume that each website acquires a unique phone number. We also believe that a telecommunication service provider participates in the registration and recovery phases. The design principle of OPass is to eradicate the harmful persuade of human factors as much as possible. Through OPass, each user only needs to memorize a long-term password which has been used to defend her/his cellphone. Users are free from typing any passwords into untrusted computers (kiosks) for login on all websites. Compared with previous schemes, OPass is the initial user authentication protocol to thwart password stealing (i.e., phishing and malware) and password reuse attacks (keylogger or backdoors) instantaneously. The reason is that OPass adopts the one-time password approach to make certain independence between each login. To make OPass completely functional, password recovery is also considered and supported when users lose their cellphones. They can make progress our OPass system with reissued SIM cards and long-term passwords. A prototype of OPass is also executed to measure its performance. The standard time spent on registration and login is 21.8 and 21.6 s, respectively. Besides, the performance of login of OPass is better than graphical password schemes, for example, Pass faces. Therefore, we believe OPass is believable and trustworthy for users.

References

[1] Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin, "oPass: A User Au-thentication Protocol Resistant to Password Stealing and Password Reuse Attacks" ieee transactions on information forensics and security, vol. 7, no. 2, April 2012 651

[2] 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.

[3] 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.

[4] T. Holz, M. Engelberth, and F. Freiling, "Learning more about the under-ground economy:Acase-study of keyloggers and dropzones," Proc. Comput-er Security ESORICS 2009, pp. 1–18, 2010.

[5] Phishing Activity Trends Rep., 2nd Quarter/2010 Anti-Phishing Working Group [Online]. Available: http://www.antiphishing.org/

[6] 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.

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

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

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

[10] S. Chiasson, A. Forget, R. Biddle, and P.C. van Oorschot. User Interface Design Affects Security: Patterns in Click-Based Graphical Passwords. In-ternational Journal of Information Security, 8(6):387–398, 2009.

[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.

Author Profile

Joga Venkata Hari Babu V received the B.Tech degree in computer science and Engineering from JNTU Hyderabad in 2011 and pursuing M. Tech degree in Computer science and Engineering from Anurag Group of Institutions (Formerly CVSR College of Engineering) JNTU Hyderabad.

G.Lavanya working as Assistant Professor in Computer Science Engineering from Anurag Group of Institutions (Formerly CVSR College of Engineering) JNTU Hyderabad. She has received the Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology, JNTU Hyderabad and M.Tech in Software Engineering from Sree Nidhi Institute of Science and Technology, JNTU Hyderabad.