International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-4 Issue-1, October 2014

Enhanced Preventation of Password Stealing using Biometric Factor

Bharti Vijay Nikose, Gaurav Shrivastav, Ravindra Gupta

Abstract- Wording password is typically the most popular form involving user authentication on websites car without any convenience in addition to simplicity. On the other hand, users' passwords are inclined to be ripped off and sacrificed under various threats in addition to vulnerabilities. To begin with, users usually select vulnerable passwords in addition to reuse exactly the same passwords all over different web sites. Routinely reusing accounts causes a domino effect; when the adversary compromises one password, she may exploit the item to gain access to more web sites. Second, keying in passwords in untrusted personal computers suffers pass word thief risk. An adversary can start several pass word stealing attacks to snatch passwords, including phishing, keyloggers in addition to malware. Within this paper, we design a user authentication process named oPass which usually leverages a user's cellular and limited message support to thwart password thieving and pass word reuse attacks. oPass simply requires each and every participating site possesses an original phone variety, and requires a telecommunication service agency in signing up and recovery phases. 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 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.

Keywords: - Network Security, Password Attacks, Authentication.

I. INTRODUCTION

Within the last few years, text password has become adopted because primary indicate of consumer authentication regarding websites. People pick their login name and textual content passwords whenever registering accounts on a website. So that you can log into your website efficiently, users should recall the particular selected passwords.

Manuscript published on 30 October 2014.
***** Correspondence Author (s)
Ms. Bharti Vijay Nikose, M. Tech Student of CSE Department RKDF Institute of Science & Technology,Hoshangabad Road, Bhopal, India.
Mr. Gaurav Shrivastav, HOD of CSE Department RKDF Institute of Science & Technology, Hoshangabad Road, Bhopal, India.
Mr. Ravindra Gupta, Asst. Prof., of CSE Department RKDF Institute of Science & Technology, Hoshangabad Road, Bhopal, India.

Generally, password-based consumer authentication may resist brute force in addition to dictionary attacks if consumers select robust passwords to produce sufficient entropy. On the other hand, password-based consumer authentication has a major problem that humans will not be experts within memorizing textual content strings. Therefore, most consumers would pick easy-to-remember passwords (i. electronic., weak passwords) even if they recognize the passwords may be unsafe. Another important problem is that users have a tendency to reuse passwords across a variety of websites. With 2007, Florencio in addition to Herley indicated a user reuses some sort of password throughout 3. 9 distinct websites on average. Password reuse causes users to forfeit sensitive data stored in several websites when a hacker compromises certainly one of their passwords. This attack is known as the security password reuse episode. The earlier mentioned problems are a result of the negative influence of human variables. Therefore, it is very important take human factors under consideration when building a consumer authentication standard protocol. Up in order to now, researchers possess investigated many different technology to lessen the negative influence of human factors within the user authentication course of action. Since humans will be more adept within remembering graphical passwords when compared with text passwords, many graphical password techniques were built to address human's security password recall difficulty. Using security password management tools can be an alternative. These tools automatically make strong passwords per website, which addresses security password reuse in addition to password remember problems. The advantage is that users simply have to remember some sort of master password to gain access to the management tool. Despite the assistance of these a couple technologies graphical password in addition to password management tool the person authentication system still is suffering from some sizeable drawbacks. Although graphical password is a superb idea, it's not necessarily yet older enough to be widely implemented used and remains to be vulnerable to several attacks. Password management tools work effectively; however, general consumers doubt the security therefore feel unpleasant about deploying it. Furthermore, they possess trouble utilizing these tools a result of the lack of security expertise. Besides the particular password reuse attack, it's also important to think about the side effects of security password stealing attacks. Adversaries take or bargain passwords in addition to impersonate users' identities in order to launch harmful attacks, acquire sensitive data, perform unauthorized transaction actions, as well as leak personal secrets.

Phishing is the most typical and useful password obtaining attack. According to APWG's record, the quantity of unique phishing internet websites detected with the second season of 2010 is 97 388. Many previous studies possess proposed schemes to defend against security password stealing attacks. The main contribution of this project is a generic framework for three-factor authentication in distributed systems. The proposed framework has several merits as follows: First, we demonstrate how to incorporate biometrics in the existing authentication based on smart card and password. Our framework is generic rather than instantiated in the sense that it does not have any additional requirements on the underlying smart-card-based password authentication. Not only will this simplify the design and analysis of three-factor authentication protocols, but also it will contribute a secure and generic upgrade from two-factor authentication to three-factor authentication possessing the practice-friendly properties of the underlying two-factor authentication system. Second, authentication protocols in our framework can provide true three-factor authentication, namely a successful authentication requires password, smart card, and biometric characteristics. In addition, our framework can be easily adapted to allow the server to decide the authentication factors in user authentication (instead of all three authentication factors). Last, in the proposed framework clients' biometric characteristics are kept secret from servers. This not only protects user privacy but also prevents a single-point failure (e.g., a breached server) from undermining the authentication level of other services. Furthermore, the verification of all authentication factors is performed by the server. In particular, our framework does not rely on any trusted devices to verify the authentication factors, which also meets the imperfect feature of distributed systems where devices cannot be fully trusted.

II. PREVIOUS WORK

Web browsing has become such an integral part of our everyday lives that we use browsers to perform many important tasks such as banking, shopping, and bill-paying. To facilitate ubiquitous Web access, many kiosk environments such as cafes, airport lounges, and hotel business centers provide people with Internet-connected public computers. These public computers often have high-speed network connections. They are also convenient to use since they normally have full-size keyboards and large displays. People who do not own a computer or carry a laptop with them frequently use these public computers to browse the Web. Unfortunately, public computers are usually far less trustworthy than peoples' own computers. By "trustworthy", we mean that it is less likely that malware or spyware has been installed on a computer to log user input, steal account information, and even secretly hijack a secure (HTTPS) Web browsing session to make fraudulent transactions. Public computers are used by many people to run different applications and visit various websites; consequently, it is very likely for them to be infected with malware or spyware. Simply searching "public computer security" online, we can find numerous articles suggesting that people should not use public computers to perform sensitive activities. For example, Microsoft suggests that to be really safe, a user should not enter any sensitive information into a public computer. To secure kiosk computing environments, researchers have proposed a number of solutions. Most of these solutions use a trusted mobile device such as a PDA (Personal Digital Assistant) or a mobile phone to enhance the security of kiosk computing environments, and we refer to them as PDA-based solutions. Mobile devices are favored by PDA-based solutions because (1) they are more portable than desktop and laptop computers, and (2) they are generally more trustworthy than public computers. Nevertheless, using small user interfaces on mobile handheld devices is inherently difficult. Many of these PDA-based solutions focus on specific objectives such as securing application or data access, securing user authentication or input, and verifying software integrity, so they cannot be easily adopted to secure an entire Web browsing session. Some solutions do have the objective of securing an entire kiosk browsing session, but they suffer from a few drawbacks that limit their practical use. Balfanz and Felten introduced a splitting-trust paradigm to divide an application between a small trusted mobile device and a bigger, more powerful, but possibly untrusted computer. Session Magnifier is inspired by this paradigm; however, we do not split a browser but instead enable an extended browser on a trusted mobile device and a regular browser on an untrusted computer to collaboratively support a Web session. The splitting-trust paradigm has also inspired many other kiosk computing solutions that rely on a trusted mobile device. We classify these solutions into four categories based on their different objectives.

2.1 Securing Application or Data Access

Oprea et al. proposed a three-party secure remote terminal architecture to enable users to access their sensitive home computing environment via a trusted mobile device and an untrusted terminal. This three-party architecture is based on a thin-client VNC (Virtual Network Computing) remote display system, in which a VNC server can update the frame buffer displayed on a VNC client. Sharp et al. proposed a VNC-based thin-client architecture to support secure access to unmodified applications. This architecture is similar to the three-party architecture, but it provides additional mechanisms to obfuscate the content displayed on an untrusted display. These VNC-based secure application or data access solutions work at the frame buffer level with high overhead, so they cannot be naturally adopted to support smooth Web interactions. In addition, trusted VNC servers must be deployed in these solutions.

2.2 Securing User Authentication or Input

Parno et al. built a Phoolproof phishing prevention system that uses a trusted mobile device to perform mutual authentication between a user and a website. Mannan and Oorschot proposed the MP-Auth protocol, in which a trusted mobile device turns a long-term password into a onetime password via the public key of an intended server; therefore, a user's long-term password will not be revealed to phishing sites or untrusted computers. McCune et al. proposed a BitE framework that leverages the features of TPM (Trusted Platform Module) to establish an encrypted input tunnel from a trusted mobile device to an application running on a TPM-equipped untrusted computer.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Clarke et al. and Wu et al. designed protocols that rely on both a trusted third-party proxy and a trusted mobile device to secure authentication on untrusted computers. In addition, Florencio and Herley proposed approaches to secure password input on untrusted computers without using mobile devices. All these solutions focus on securing user authentication or input, so they are not directly applicable for securing Web browsing sessions.

2.3 Verifying Software Integrity

Garriss et al. built a system that uses a mobile device to establish trust in a kiosk computing environment. This system employs both a TPM module equipped on a kiosk computer and an integrity attestation server of the kiosk, and it focuses on verifying the identity and integrity of software loaded on a public computer before revealing sensitive information to the computer. However, our Session Magnifier focuses on securing Web browsing sessions on potentially untrusted public computers.

2.4 Securing Web Browsing Sessions

A few kiosk computing solutions share the same objective with our Session Magnifier: securing Web browsing sessions. Ross et al. proposed a composable secure proxy architecture to provide secure multi-modal access to Web services from any device. A similar proxy-based architecture called Delegate was proposed to enable users to access Web services from untrusted computers. In these solutions, essentially it is the browser on an untrusted computer that accesses remote Web servers; meanwhile, secure proxies perform content and control filtering functionalities. Two main obstacles impede the adoption of these proxy-based solutions. First, secure third-party proxies must be widely deployed, well managed, and fully trusted by users. Second, to secure Web browsing, a proxy must use very complicated and comprehensive rules to validate requests, remove sensitive content, maintain user information, and manage session information such as HTTP cookies.

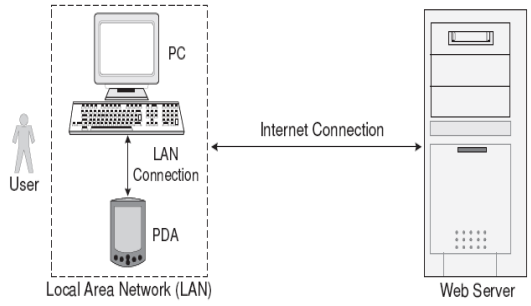

Figure 1. Kiosk browsing environment.

Margolin et al. introduced a Guardian framework that uses a PDA as a proxy for all interactions between an untrusted computer and remote Web servers. This framework eliminates the requirement of using secure third-party proxies by moving their content and control filtering functionalities to a PDA. However, since it is still the browser on the untrusted computer that accesses remote Web servers, this solution does not reduce the inherent complexity of proxy based solutions. Our Session Magnifier directly uses the Web browser on a trusted mobile device to access remote Web servers, so it provides strong security assurances and greatly reduces the complexity of content and control filtering. Recently, Sharp et al. proposed a split-trust browsing architecture to explore splitting trust at the HTML level for Web applications. However, this architecture has three drawbacks that limit its practical application. First, its critical component the RDC (Remote Device Communication) agent must be installed on an untrusted computer. Second, its end-to-end security between a trusted mobile device and a remote Web server depends on an extra authentication and key-exchange process coordinated by the RDC agent. Third, it assumes that either Web applications are explicitly written or secure HTML-rewriting proxies are used to support split trust browsing. In contrast, our Session Magnifier is much simpler and more practically applicable – nothing needs to be installed or configured on an untrusted computer, end to end security between a trusted mobile device and a remote Web server is ensured by existing HTTPS connections, no third-party proxy is needed, and no modification needs to be made to existing Web applications. [1] Special consideration is required to design usable, understandable, and manageable security features. At first glance, it seems like applying standard usability and Human-Computer Interaction (HCI) principles should suffice, but security constraints make this problematic. Most importantly, some design features that might make a system more usable would also make it less secure. Addressing these security weaknesses can too easily render the software unusable again. Even worse, one might argue that an unusable security system is inherently insecure, since users will then misuse or bypass the security mechanisms. One must also consider how the design affects the observable behavior of legitimate users, in case such behavior could be exploited by attackers. The challenge is to design software that is both secure and usable. Security is rarely a user's primary task, and typically involves an extra step in addition to the main task, such as having to log in to read one's email. Users need security features to be as non-disruptive as possible, but still need them to work properly to preserve integrity and privacy. A second unusual characteristic of security software is that it attracts illegitimate users of the system who are actively trying to gain unauthorized access. These attackers will take advantage of all information available. Usable security software must therefore offer assistance to legitimate users, without giving assistance to attackers. In particular, this changes the nature of feedback in interaction design, which must inform legitimate users while revealing no useful information to others. With any authentication system where users are expected to recall information to log in, there is a risk of memory interference. Multiple password interference occurs when users must remember passwords for many systems and the memories of the different passwords interfere with each other. Studies have shown that users typically create easy-to-guess text passwords and reuse these passwords across several accounts. When trying to log in, they will cycle through their passwords until they find one that works. Gaw and Felten report that users in their lab study tried an average of 2.43 passwords before a correct login.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

This may be under-reporting the problem, however, because users in their study were only allowed 90 seconds per account. While this trial-and error approach helps users deal with password systems and multiple password interference, revealing all of one's passwords at every login can amplify security risks, for example in the presence of key loggers or when passwords are sent to phishing sites. One proposed solution to the password problem is to use a password manager. With a password manager, users typically have one master password and the password manager creates, stores, and enters passwords for individual accounts on behalf of the user. The individual passwords are typically much more random than what users would select on their own and are thus stronger against attack. However, implementations of some password managers have usability problems that can leave users even more vulnerable than when they were managing passwords themselves. A second drawback is that a centralized scheme has a new single point of failure: if attackers gain access to the master password, they now have control over all of the user's accounts. While password managers may be appropriate in some circumstances, authentication schemes that are both secure and memorable are still needed. We are interested in the graphical password approach. It has been suggested that graphical passwords may be less susceptible to multiple password interference since humans have better memory for recognizing and recalling images than text. Surveys of graphical passwords circa 2005 are available from Suo et al. and from Monrose and Reiter. Proposed schemes include click-based graphical passwords such as PassPoints. Many of these have the added advantage of presenting a cue to the user to help trigger the appropriate memory. Cued-recall has been established as an easier memory task than uncued recall. With cued-recall, the system provides a cue to help prompt the user's memory of the password (or a portion thereof). This is a desirable usability feature that reduces the memory load on users. With click based graphical passwords, a password consists of user-selected click-points on the images presented. Therefore, the images act as mnemonic cues to remember the corresponding click-points. In PassPoints, users are presented with an image, and a password consists of 5 click-points on the image. To log in, users must select the same 5 click-points in the same order. The system allows for a tolerance area around each click-point so that approximately correct login attempts are accepted. Several user studies and security analyses have been conducted on Pass- Points. While these have found PassPoints to be generally usable, security concerns have been raised because users tend to select predictable passwords which are exploitable in dictionary attacks. Newer click-based graphical password schemes, such as Persuasive Cued Click-Points, address two important security concerns with respect to user selected passwords: they offer a significant reduction in hotspots (i.e., areas of the image that have higher probability of being selected by users) and in the use of click-point patterns (such as selecting click-points that form a straight line across the image). These characteristics significantly reduce vulnerability to dictionary attacks. The present paper uses the better-known PassPoints scheme for these interference tests, in order to leverage a more closely-examined and understood password scheme and to build on existing results on interference between two passwords only. A few studies have compared text passwords to graphical passwords, but in these cases, users only had one password to remember (either text or graphical). Wiedenbeck et al. compared user performance of text passwords and PassPoints in a lab study. Their results were mixed, but slightly favored text passwords. Komanduri and Hutchings's study compared text passwords to their newly proposed picture-password scheme. They found better memorability for their picture-passwords although the results were not statistically significant due to a small user sample. [2] Most people find it difficult to remember alphanumeric passwords, a problem magnified by the fact that an average Web user has passwords on 25 unique Web sites. This difficulty leads people to adopt a number of unsafe strategies, including writing passwords down, reusing the same password, using minor variants of a single password, or frequently reinitializing passwords upon failure to authenticate. All of these behaviors increase the likelihood of passwords being lost, stolen, or compromised. Graphical password systems have received significant attention as one potential solution to the need for more usable authentication. Graphical password systems take many forms, such as requiring the selection of target images from sets of distracter images or requiring clicks on target regions of an image. Graphical passwords are generally considered to be easier to remember and use than alphanumeric passwords because graphical passwords take the proven approach of relying upon *recognition* instead of requiring *recall*. A separate advantage of graphical passwords is their natural appropriateness for situations where text entry is difficult or limited (e.g., when using a small mobile device with limited keyboard input, such as popular touch screen phones). Some graphical password systems provide a level of strength (entropy) against password guessing attacks that is equal to or greater than typical alphanumeric passwords, but this is not a strict requirement. Instead, it is clear that different approaches exist at a range of points in a trade-off between usability and cryptographic strength. When password usability is important to an application, even a weak (low entropy) password system can provide sufficient security when used as part of a larger multi-factor authentication system. Widely used four-digit PINs, for example, are typically paired with the need to physically possess an ATM card and a limit on the number of failed attempts allowed before the ATM card is confiscated. The continuing emergence of the mobile Web seems to promise many additional opportunities for multi-factor approaches. A social networking site, for example, may want to reduce the burden of mobile authentication, but mobile text entry is relatively difficult (especially for the special characters and non-word sequences common in passwords). The site might therefore require that a device initially be authenticated using an alphanumeric password, but then place a cookie on the device. Future access could then use a combination of the cookie on the authenticated device and an easier graphical password. As in the ATM card example, this cookie could be revoked after as little as a single failed attempt at the graphical password.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

This system would allow people to easily access protected sites from their mobile devices, but even the use of a weak password will guard against illegal access to those sites by someone who might have found or stolen the device.

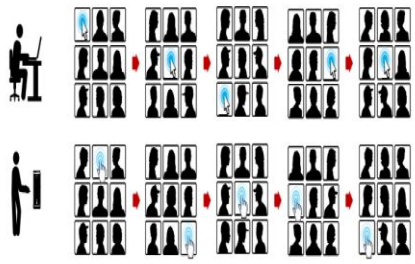

Figure 1: A person attempting to authenticate with a facial graphical password is presented a sequence of 3x3 grids of faces. Successful authentication requires choosing the correct face from each set. We selected facial graphical passwords for study in this work because of their commercial deployment in the PassFaces™ system [17] and because of their use in prior research [4, 9, 21]. Such a system is appropriate for many situations, including the desktop and in mobile situations where text input is more difficult.

Given the need for more usable authentication and existing interest in graphical passwords as a potential solution, we identify an important limitation of existing work: although there have been many studies of graphical passwords, nearly all prior work focuses on a *single* password. People will need to remember and use many graphical passwords, just as they currently use many alphanumeric passwords, but no work has systematically explored the use of multiple graphical passwords. [3]

III. PROPOSED SYSTEM

3.1 Initialization

Each web server possesses a unique node identifier. Via the identifier, users can interact with each website through an SMS channel. The users' cellphones are malware-free. Hence, users can safely input the long-term passwords into cellphones. 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 eachweb 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 to the corresponding web server based on the received ID. Subscribers (i.e., users) connect to the TSP via 3G connections to protect the transmission. 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 sent from the subscriber. If a user loses her cellphone, she can notify her TSP to disable her lost SIM card and apply a new card with thesame phone number. Therefore, the user can perform the recovery phase using a new cellphone. See Section IV for additional details..

3.2 Registration Phase

This module 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 (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 based on user's SIMcard. The TSP also plays the role of third-party to distribute a shared key between the user and the server. The shared key is used to encrypt the registration SMS with AES. The TSP and the server will establish an SSL tunnel to protect the communication. Then the TSP forwards ID to the assigned server. Server will generate the corresponding information for this account and reply a response, including server's identity ID, a random seed, and server's number. The TSP then forwards ID, and a shared key to the user's cellphone. Once reception of the response is finished, the user continues to setup a long-term password with her cellphone. The cellphone computes a secret credential to prepare a secure registration SMS, the cellphone encrypts the computed credential with the key and generates the corresponding MAC takes input user's identity, ciphertext. Then, the cellphone sends an encrypted registration SMS to the server by phone number. At the end of registration, the cellphone stores all information ID, except for the longterm password and the secret.

3.3 Login Phase

The *login* phase begins when the user sends a request to the server through an untrusted browser. The user uses her cellphone to produce a one-time password, and deliver necessary information encrypted with to server via an SMS message. Based on preshared secret credential, server can verify and authenticate user. The protocol starts when user wishes to log into her favorite web server. However, begins the login procedure by accessing the desired website via a browser on an untrusted kiosk. The browser sends a request to with 's account ID . Next, server supplies the ID and a fresh nonce to the browser. Meanwhile, this message is forwarded to the cellphone through wireless interfaces. After reception of the message, the cellphone inquires related information from its database via ID which includes server's phone number and other parameters. The next step is promoting a dialog for her long-term password. Secret shared credential can regenerated by inputting the correct on the cellphone. The one-time password for current login is recomputed. To prepare a secure login SMS, the cellphone encrypts and with and generates the corresponding MAC. Upon successful verification, the server sends back a success message through the Internet, 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. If the user is successfully log into the server, index is able to automatically increased, in both the device and the server for synchronization of one-time password. After rounds, the user and the server can reset their random seed by the *recovery* phase to refresh the one-time password.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

3.4 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. Once user installs the oPass program on her new cellphone, she can launch the program to send a recovery request with her account ID and requested server ID to predefined TSP. As we mentioned before, ID can be the domain name or URL link of server . Similar to registration, TSP can trace her phone number based on her SIM card and forward her account ID to server through an SSL tunnel. Once server receives the request, probes the account information in its database to confirm if account is registered or not. If account ID exists, the information used to compute the secret credential will be fetched and be sent back to the user. The server generates a fresh nonce and replies a message which consists of ID. This message includes all necessary elements for generating the next one-time passwords to the user. 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. During the last step, the user's cellphone encrypts the secret credential and server nonce to a ciphertext. The recovery SMS message is delivered back to the server for checking. Similarly, the server computes and decrypts this message to ensure that user is already recovered..

3.5 Three Factor Initialization

In this module we describe the initialization phase in the proposed framework. This phase generates a public parameter and a secret parameter for three-factor authentication. The 2- Factor-Initialization is the initialization algorithm in the underlying protocol. Given a security parameter k, which is the size of the public and secret keys, the authentication server S in our framework runs 2-Factor- Initialization twice to generate two separate PK and SK. The two pairs (PK1; SK1) and (PK2; SK2) are generated in an independent manner. The public parameter in three-factor authentication is the pair (PK1; PK2), and the corresponding secret parameter is the pair (SK1; SK2).

3.6 Three Factor Registrations

In this module the registration in our framework is made up of the following steps. In the module we use h as cryptographic hash function chosen by the client C. and An initial password PW1 is chosen by the client C. The function GenBioData is computed where a pair R; P is generated using C's biometric template BioData and the algorithm Gen in the fuzzy extractor. We assume there is a device extracting the biometric template and carrying out all calculations in the fuzzy extractor. This step does not involve any interaction with the authentication server. Let PW2 be the second password. The second "password" PW2 is calculated from the random string R. R will be deleted immediately once the calculation of PW2 is complete. Then C (using PW1) and S (using SK1) first execute the 2-Factor-Reg protocol. Let Data1 be the data generated by S at this step. C and S have another run of 2-Factor-Reg protocol, where C registersPW2 and S uses SK2 to generate the corresponding data Data2. PW2 will be deleted immediately once the registration is complete. S generates a smart card SC which contains Data1 and Data2. The client C is given SC which is the smart card. C updates the data in the smart card SC by adding Data3 which contains the auxiliary string P, the description of the hash function h, the reproduction algorithm.

3.7 Three Factor Login Authentications

The client C first retrieve the smart card SC data from the card reader, which will extract the data Data1; Data2; Data3. After that, C inputs the password PW1 and his/her biometric data. Let BioData' be the biometric template extracted at this phase. Then it calculate R; P and PW2. A random string R is calculated from the biometric template BioData' and the auxiliary string P by running the algorithm. The random string R will be the same as the one generated at the registration phase if BioData' is close to BioData. C (using PW1 and Data1) and S (using SK1) first execute the 2-Factor-Login-Auth protocol of SCPAP. C and S have another run of 2-Factor-Login-Auth, where C uses PW2 and Data2, and S uses SK2. The protocol outputs "1" if and only if both executions of 2-Factor-Login-Auth protocol output "1." Otherwise, the protocol outputs "0."

IV. RESULTS

The concept of this paper is implemented and different results are shown below, The proposed paper is implemented in Java technology on a Pentium-IV PC with minimum 20 GB hard-disk and 1GB RAM. The propose paper's concepts shows efficient results and has been efficiently tested on different Datasets.

Fig. 1 Registration, Login and Recovery Menu on Mobile

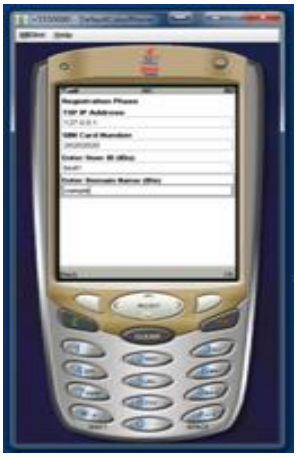

Fig. 1 Performing Registration Phase

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Fig. 3 Performing Generation of Certificate

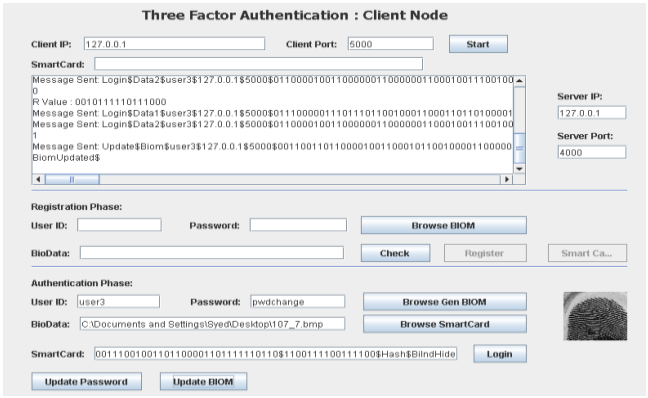

Fig. 4 Certificate Generation with Biometric Parameter

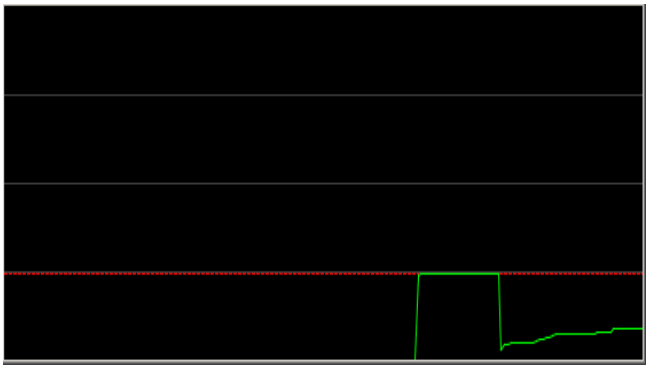

Fig. 5 Time Taken by Mobile Node to Initialize

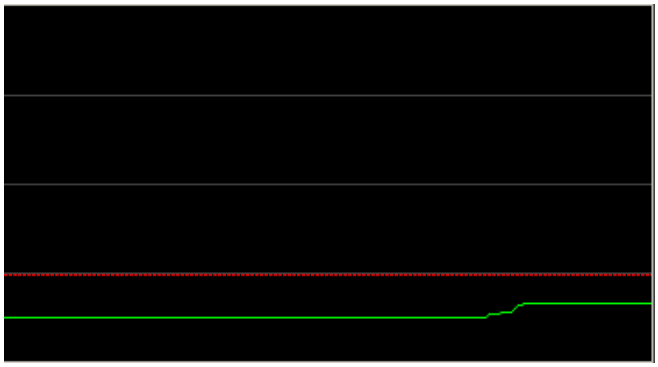

Fig. 6 Time Taken by Registration Protocol

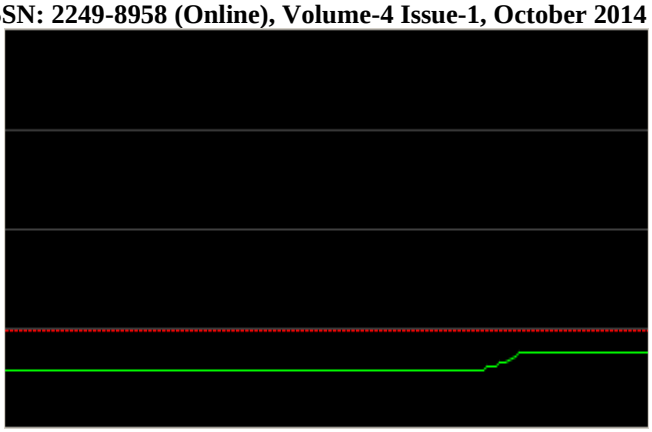

Fig. 7 Time Taken by Login Phase

V. CONCLUSIONS

With this paper, most of us proposed the user authentication project named oPass which leverages cellphones and TEXT MESSAGE to circumvent password taking and code reuse assaults. We assume that every website possesses an original phone amount. We in addition assume a telecommunication service provider participates in the registration in addition to recovery phases. The layout principle connected with oPass is to eliminate your negative have an effect on of man factors as much as possible. Through oPass, each user only should remember the long-term password which has been helpful to protect the woman cellphone. Users are clear of typing almost any passwords in untrusted pcs for get access on all websites. Compared with previous plans, oPass will be the first user authentication protocol to avoid password taking (i. age., phishing, keylogger, in addition to malware) in addition to password reuse attacks together. The cause is that will oPass explores the one-time password method of ensure self-reliance between each and every login. For making oPass totally functional, password recovery can be considered in addition to supported whenever users get rid of their cellphones. They can recover our oPass method with reissued SIM cards and long-term accounts. The authentication is based on password, smart card, and biometrics. Our framework not only demonstrates how to obtain secure three-factor authentication from two-factor authentication, but also addresses several prominent issues of biometric authentication in distributed systems (e.g., client privacy and error tolerance). The analysis shows that the framework satisfies all security requirements on three-factor authentication and has several other practice-friendly properties (e.g., key agreement, forward security, and mutual authentication). The future work is to fully identify the practical threats on three-factor authentication and develop concrete three factor authentication protocols with better performances.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.