Novel Implementation of O-Pass Security Model Design of User Authentication for Password Stealing and Reuse Attacks

Dr.M.Ramabai[1]J.V.Prashanthi[2], P.Monika Raju[3], A.V.L Prasuna[4], M.Upendra Kumar[5]
[1]*Professor, IT and CSE, MGIT, Hyderabad, A.P. India*
[2]*M.Tech C.N.I.S. Student, MGIT, Hyderabad A.P. India*
[3]*B.Tech IT Student, MGIT, Hyderabad, A.P. India*
[4]*Associate Professor, IT, MGIT, Hyderabad, A.P. India*
[5]*Associate Professor, CSE, MGIT, Hyderabad, A.P. India*

ABSTRACT:*Text password is the most popular form of user authentication on websites due to its convenience and simplicity. However, users' passwords are prone to be stolen and compromised under different threats and vulnerabilities. Firstly, users often select weak passwords and reuse the same passwords across different websites. Routinely reusing passwords causes a domino effect; when an adversary compromises one password, she will exploit it to gain access to more websites. Second, typing passwords into untrusted computers suffers password thief threat. An adversary can launch several password stealing attacks to snatch passwords, such as phishing, key loggers and malware. In this paper, we design a user authentication protocol named oPass which leverages a user's 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. This paper provides a novel design and implementation of this OPass Security Model.*

Keywords –*Authentication, User Password, Stealing, Reuse Attacks, Design Implementation, OPass Model*

I. INTRODUCTION

Over the past few decades, text password has been adopted as the primary mean of user authentication for websites. People select their username and text passwords when registering accounts on a website. In order to log into the website successfully, users must recall the selected passwords. Generally, password-based user authentication can resist brute force and dictionary attacks if users select strong password 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. 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. Up to now, researchers have investigated a variety of technology to reduce the negative influence of human factors in the user authentication procedure

Since humans are more adept in remembering graphical passwords than text passwords many graphical password schemes were designed to address human's password recall problem. Using password management tools is an alternative. These tools automatically generate strong passwords for each website, which addresses password reuse and password recall problems. The advantage is that users only have to remember a master password to access the management tool. Despite the assistance of these two technologies graphical password and password management tool the user authentication system still suffers from some considerable drawbacks. Although graphical password is great idea it's not implemented very well. Password management tools work well; however, general users doubt its security and thus feel uncomfortable about using it. Furthermore, they have trouble using these tools due to the lack of security knowledge. Phishing is the most common and efficient password stealing attack. Some researches focus on three-factor authentication rather than password-based authentication to provide more reliable user authentication. Three-factor authentication depends on what you know (e.g., password), what you have (e.g., token), and who you are (e.g., biometric).

To pass the authentication, the user must input a password and provide a pass code generated by the token (e.g., RSA SecureID and scan her biometric features (e.g., fingerprint or pupil). Three-factor authentication is a comprehensive defense mechanism against password stealing attacks, but it requires comparative high cost.

Thus, two-factor authentication is more attractive and practical than three-factor authentication. Users have to memorize another four-digit PIN code to work together with the token, for example RSA Secure ID. In addition, users easily forget to bring the token. So the new proposal is a user authentication protocol named oPass which leverages a user's cell phone and short message service (SMS) to prevent password stealing and password reuse attacks.

Motivation of This Novel Design and Implementation

Now a day's organization makes use of huge amount of information. In order to keep the information secure some sort of password management strategy is needed. With the advancements in technology, hackers are hacking the passwords so as to steal the important information. So securing the passwords is another important issue.

An organization's competitiveness is increased when the data it maintained is secured. So many kinds of passwords have been used. There are many drawbacks exhibited during usage of those. Hotspots are more for attackers so that they can guess passwords and make the gain of important data available.

And these days phishing has become more common so as to steal password and gain access to data. So in order to reduce these types of attacks, there is a need of some mechanism which eliminates entering genuine passwords in the websites. So this idea has been developed for that purpose. Here most of phishing attacks can be avoided.

Objectives

To provide authenticated environment to users
To manipulate privacy for sensitive data which is valuable and confidential?

Problem Definition

There are many challenges faced by users in securing passwords. Some passwords are user friendly and doesn't provide proper security. Some are difficult to remember and some exhibits more hotspots for attackers. So securing passwords is a great challenge.

Problem Solution

In this system, android mobile will be receiving the one time password for accessing website. Authenticated users will have their unique mobiles so that password will be unique. Here hackers cannot gain hotspots or cannot gain passwords through plug-ins.

Literature Review

In [ref1] there is a study on the impact of selected parameters on the size of the password space for "Draw-A Secret" (DAS) graphical passwords. They examine the role of and relationships between the number of composite strokes, grid dimensions, and password length in the DAS password space. They show that a very significant proportion of the DAS password space depends on the assumption that users will choose long passwords with many composite strokes. If users choose passwords having 4 or fewer strokes, with passwords of length 12 or less on a 5 × 5 grid, instead of up to the maximum 12 possible strokes, the size of the DAS password space is reduced from 58 to 40 bits. Additionally, they found a similar reduction when users choose no strokes of length 1. To strengthen security, they propose a technique and describe a representative system that may gain up to 16 more bits of security with an expected negligible increase in input time. Their results can be directly applied to determine secure design choices, graphical password parameter guidelines, and in deciding which parameters deserve focus in graphical password user studies.

In [ref2] there is an introduction and evaluation on various methods for purely automated attacks against Pass Points style graphical passwords. For generating these attacks, they introduce a graph-based algorithm to efficiently create dictionaries based on heuristics such as click-order patterns (e.g., 5 points all along a line). Some of their methods combine click-order heuristics with focus of- attention scan-paths generated from a computational model of visual attention, yielding significantly better automated attacks than previous work. One resulting automated attack finds 7-16% of passwords for two representative images using dictionaries of approximately 226 entries (where the full password space is 243).

Relaxing click-order patterns substantially increased the attack efficacy albeit with larger dictionaries of approximately 235 entries, allowing attacks that guessed 48-54% of passwords (compared to previous results of 1% and 9% on the same dataset for two images with 235 guesses). Their results show that automated attacks, which are easier to arrange than human-seeded attacks and are more scalable to systems that use multiple images, pose a significant threat to basic Pass Points-style graphical passwords.

In [ref3], given the widespread use of password authentication in on- line correspondence, subscription services, and shopping, there is growing concern about identity theft. When people reuse their passwords across multiple accounts, they in- crease their vulnerability; compromising one password can help an attacker take over several accounts. They sometimes failed to realize that personalized passwords such as phone numbers can be cracked given a large enough dictionary and enough tries. They discuss how current systems support poor password practices. They also present potential changes in website authentication systems and password managers.

In [ref4], they report the results of a large scale study of password use and password re-use habits. The study involved half a million users over a three month period. A client component on users' machines recorded a variety of password strength, usage and frequency metrics. This allows us to measure or estimate such quantities as the average number of pass- words and average number of accounts each user has, how many passwords she types per day, how often passwords are shared among sites, and how often they are forgotten. They get extremely detailed data on password strength, the types and lengths of passwords chosen, and how they vary by site. The data is the large scale study of its kind, and yields numerous other insights into the role the passwords play in users' online experience.

In [ref5], while password theft is a threat to the system from which the passwords were stolen, the network password vulnerability also threatens other systems. If users have many password-protected accounts and they reuse a password across more than one account, a hacker gaining access to one account may be able to gain access to others. If, for example, a hacker gains access to a weakly defended departmental file server and those passwords are stolen, those passwords could be used to gain access to a more secure corporate system.

Such attacks are difficult to defend against. In many systems, if a hacker tool repeatedly attempts to use incorrect passwords to access a particular account, the system will, usually after several tries, shut down the account in defense. In this scenario where variations on both the user-ID and password are involved, no such defense will be effective and the hacker will be free to run long lists of such pairs. Once the user-ID/password pairs have been tested, a hacker could add database entries capturing identities of systems accessible by a particular pair. This database would then be a powerful tool for systems infiltration, with coordinated attacks arranged to maximize the damage before the problem is discovered.

In [ref6], they address an important issue in the usable security of user authentication software: the memorability of multiple passwords. Authentication software supports legitimate users in gaining access to systems or resources by verifying their credentials. They focus on passwords, the most common form of credentials. The problem with passwords is making them easy for legitimate users to remember, but difficult for attackers to guess. Alternatives to passwords include physical tokens or biometrics; these also have their problems, such as cost, management, and privacy, which we will not address in the paper. As passwords are the most common method of authentication, the password problem is important, and is made worse by the increasing number of users and the number of different systems they access. In particular, users now need to remember not just one password, but many. This places a significant memory load on users, leading them to choose (and reuse) simple passwords that are easy for attackers to guess.

In such systems, the user does not enter a text password using a keyboard, but instead clicks on particular points on an image. Such graphical passwords are intended to take advantage of the human ability to more easily recognize and recall images than textual information. They wished to study whether this approach would have advantages over ordinary text passwords when multiple distinct passwords were necessary, since there is currently little research on this topic. They were particularly concerned about the potential for multiple password interference, where remembering a password for one system might affect the user's memory of a password for another system.

In [ref7] , current security systems suffer from the fact that they fail to account for human factors.

This paper considers two human limitations: First, people are slow and unreliable when comparing meaningless strings; and second, people have difficulties in remembering strong passwords or PINs. They identify two applications where these human factors negatively affect security: Validation of root keys in public-key infrastructures, and user authentication. Their approach to improve the security of these systems is to use hash visualization, a technique which replaces meaningless strings with structured images. They examine the requirements of such a system and propose the prototypical solution Random Art. They also show how to apply hash visualization to improve the real-world security of root key validation and user authentication .

II. METHODOLOGY

Introduction

Special consideration is required to design usable, understandable, and manageable security features. At first glance, it seems like applying standard usability and HCI principles should suffice, but there are security constraints that 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.

Methodology

In this system,

1. User enters the username, userid and server no to the server.
2. Then based on the mobile which the user is using the TSP traces the phone number of user and long term password is sent to him via mobile.
3. Based on the long term password and the server number one time password will be generated.
4. The one-time password generated is unique for every session.
5. Finally in tomcat server's website, username along with url and otp should be entered so as to redirect to original website.

Feasibility study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of a proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below:

Technical Feasibility

Technical feasibility assesses whether the current technical resources are sufficient for the new system. If they are not available, can they be upgraded to provide the level of technology necessary for the new system. It checks whether the proposed system can be implemented in the present system without supporting the existing hardware. This system can be implemented in the present system. It need not be upgraded to run in this system.

Economic Feasibility

Economic feasibility determines whether the time and money are available to develop the system. It also includes the purchase of new equipment, hardware, and software. A software product must be cost effective in the development, on maintenance and in the use. Since the hardware and resources are already available with the organization and the organization can afford to allocate the required resources. Software needed to run this system can be downloaded from internet for free. Extra hardware like GSM modem and USB to serial connector is needed to implement this system but it doesn't incur much cost.

Operational feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. The resources that are required to implement or install are already available with the organization. The persons of the organization need no exposure to computer but have to be trained to use this particular software. A few of them will be trained. Further, training is very less. The management will also be convinced

that the project is optimally feasible. For this system, there is no need of any explicit training for users to run this. .apk files are needed to be installed in phone. By opening them we can simply run the project if we know what actually it is doing. There is no additional training required to run this.

III. SYSTEM DESIGN ARCHITECTURE

System Design

Software design is the process through which the requirements are translated into representation of software. Design is the technical kernel of the software engineering. During design, progressive refinements of data structure, program architecture, interfaces and procedural detail are developed, reviewed and documented. Design results in representations of software that can be assessed for quality.

Architectural Design

Architectural design involves identifying the software components, decoupling and decomposing them into processing modules and conceptual data structures and specifying the interconnections among the components.

Software architecture alludes to the overall structure of the software and the ways in which that structure provide conceptual integrity for a system. In its simplest form, architecture is the hierarchical structure of the program components (modules), the manner in which these components interact, and the structure of the data that are used by the components. The primary objective of architectural design is to develop a modular program structure and represent the control relationships between modules. In addition, architectural design melds program structure and data structure, defining interfaces that enable data to flow throughout the program.

Here is such an architecture of opass system.

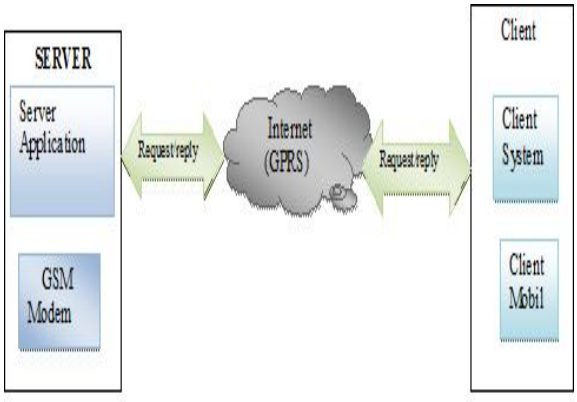

Fig 1. System architecture

The system can be separated into various modules in the different layers which perform the respective processes. Now we have a detailed look of modules.

Registration Module

The aim of this phase is to allow a user and a server to negotiate a shared secret to authenticate succeeding logins for this user. The user begins by opening the oPass program installed on her cell phone she enters IDu (account id she prefers) and IDs (usually the website url or domain name) to the program. The mobile program sends account id and url to the Telecommunication Service Provider (TSP) through a 3G connection to make a request of registration. Once the TSP received the account id and the url, it can trace the user's phone number based on user's SIM card.

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-CBC. The TSP and the server will establish an SSL tunnel to protect the communication. Then the TSP forwards account id, and 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 phone number. The TSP then forwards id, and a shared key to the user's cell phone. Once reception of the response is finished, the user continues to setup a long-term password with her cell phone.

Login Module

The login phase begins when the user sends a request to the server through an untrusted browser (on a kiosk). The user uses her cell phone to produce a one-time password, e.g., and deliver necessary information encrypted with to server via an SMS message. Based on pre shared secret credential, server can verify and authenticate user. The detail flows of the login phase. The protocol starts when user wishes to log into her favorite web server (already registered). However, begins the login procedure by accessing the desired website via a browser on an untrusted kiosk. The browser sends a request to with account IDs. Next, server supplies the ID and a fresh nonce to the browser.

Meanwhile, this message is forwarded to the cell phone through GSM Modem. After reception of the message, the cell phone inquiries related information from its database via IDs, 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 regenerate by inputting the correct on the cell phone. The one-time password for current login is recomputed If the received equals the previously generated, the user is legitimate; otherwise, the server will reject this login request. Upon successful verification, the server sends back a success message through the Internet, if the user is successfully log in to the server.

Recovery Module

Recovery phase is designated for some specific conditions; for example, a user may lose her cell phone. The protocol is able to recover oPass setting on her new cell phone assuming she still uses the same phone number (apply a new SIM card with old phone number). Once user installs the oPass program on her new cell phone, she can launch the program to send a recovery request with her account ID and requested server ID to predefined TSP through a 3G connection.

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 and the 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.

This message Procedure of recovery phase. 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 cell phone encrypts the secret credential and server nonce to a cipher text. The recovery SMS message is delivered back to the server for checking. Similarly, the server computers and decrypts this message to ensure that user is already recovered. At this point, her new cell phone is recovered and ready to perform further logins. For the next login, one-time password will be used for user authentication

GSM Modem Implementation Module

GSM modem is a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. From the mobile operator perspective, a GSM modem looks just like a mobile phone. Importing the comm Driver and connecting the Modem to the PC with serial port.

IV. Novel Design Model

Analysis models:

Figure 2 shows use case diagram of the proposed model. Figure 3 shows the sequence diagram of the proposed model. Figure 4 shows the activity diagram of the proposed model. Figure 5 shows the class diagram of the proposed model.

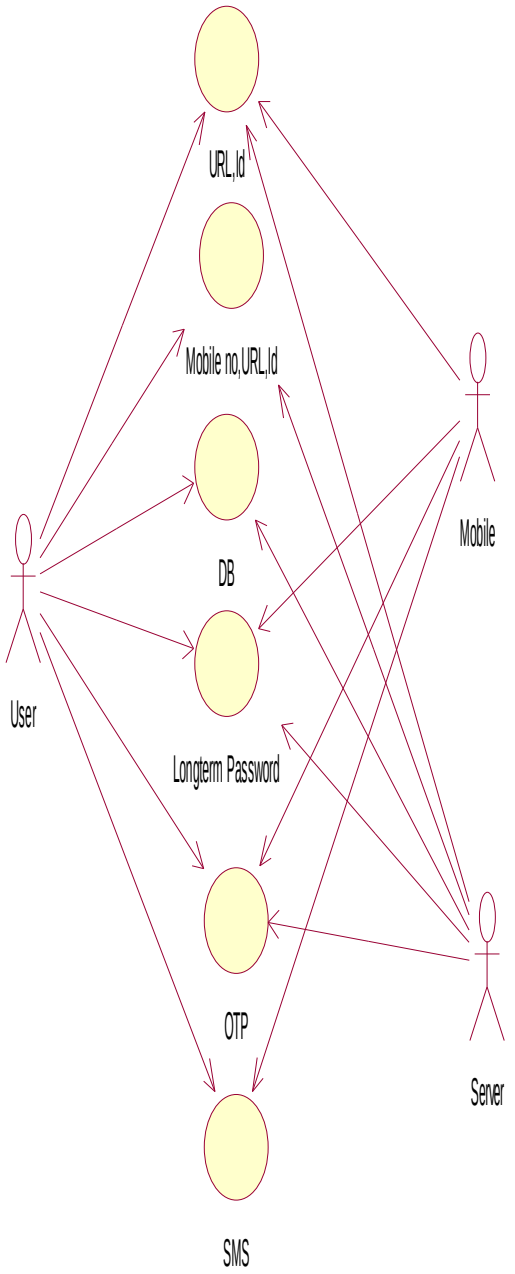

Fig 2 : Usecase diagram

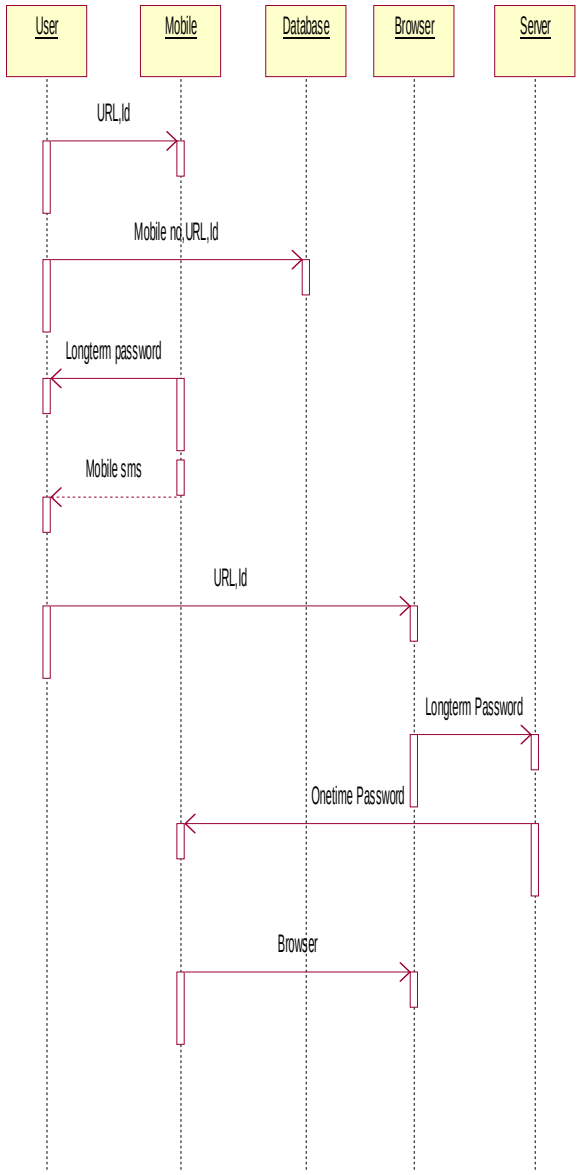

Fig 3: Sequence diagram

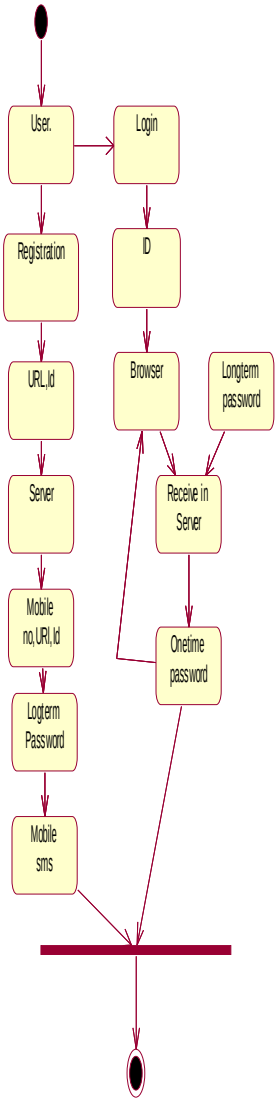

Fig 4: Activity diagram

V. IMPLEMENTATION

Step–by–Step Procedure

The step by step procedural implementation of the opass system is illustrated here. The input for the overall system is the username, user id,url and the apk files that are needed to be loaded into the android phone for sending sms and all the details are to be stored in database and to be extracted when needed. The input for every iteration of the system-run includes the long term password that should be given as input to server so that everytime one one-time password will be generated for access of webpage. The output that is produced by the system is the respective page that is requested by user. This system has been developed using MySQL database, android java application and eclipse. It comprises of two phases, one for generating long term password, and another for the generation of one-time password for access of webpage.

Database Loading

1. Get the username, url, uid, serverno that should be sent to server which is taken as input feed.
2. Then long term password will be received to phone and this will be stored in database.
3. Then long term password and serverno should be entered in apk file so that otp will be generated and stored in database for that session.
4. Load all the data in the database according to the given DB schema.

Database Analysis

1. Get the user name, uid and url of the user who wants the access of web page.
2. Based on given details users phone no will be traced and long term password will be generated to users phone.
3. This long term password and the server number should be provided to apk file so that otp will be generated. During this process that long term password corresponding to the phone number provided is valid or not will be checked from database.

4. If the provided otp is valid for the obtained long term password then that password is valid one. This should be thoroughly checked with database for each iteration.
5. Generate the display of the desired webpage.

Implementation of overall system

The class diagram is shown to show the classes that will implement the major functionalities.

The list of classes here are

1. User
2. Mobile
3. Server
4. Database

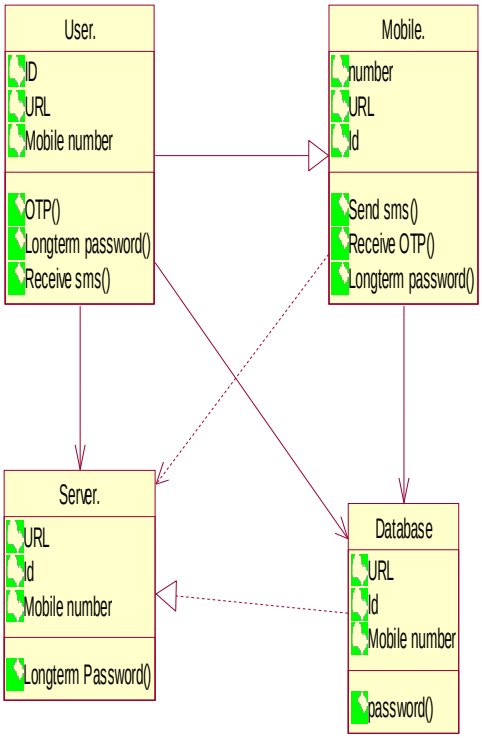

Fig 5: Implementation of system using class diagram

The functionalities of each of these classes are:

User: User gives userid, url and desired server number to the server. Then server traces the users phone number and sends the long term password to the users phone. That should be saved in database. After receiving this sms user will provide that long term password and server number in the apk file to generate the otp. After he gets otp he will enter that along with userid and url in tomacat servers website so as to redirect to the original website.

Mobile: when user enters the userid, url and server no the mobile number will be tracked. The long term password will be generated to the respective phone number. And after that apk file will be installed in the mobile itself and from that file otp will be generated which is unique for every session.

Server: Server after receiving the provided details of user, it traces the phone number and sends the long term password to users mobile. Based on long term password generated otp will be generated uniquely for every session.

Database: database will store all the data which is entered by the user and retrieve the same when needed.

VI. RESULTS AND DISCUSSIONS

The system was evaluated using several test cases which have all been completed successfully. Each of these test cases were run for different users each time with different one time passwords and evaluated for the expected behavior. Some of these cases are shown in below table:

Table 1: Test Case Description

	TESTCASE DESCRIPTION	EXPECTED BEHAVIOUR
1	Valid server number	Long term password will be geneated
2	Invalid server number	No password will be generated
3	Provide valid full length url	Redirects to valid webpage
4	Provide short length url	Doesn't redirect to webpage
5	Valid long term password	Valid otp will be generated
6	Invalid long term password	Otp will not be generated
7	Valid sequence of username and url	Requests for long term password
8	Invalid sequence of username and url	Shows error message
9	Valid otp	Redirects to desired webpage
10	Invalid otp	Shows error message

CONCLUSION

In this paper, an user authentication protocol called OPass has been implemented for secure passwords. The design principle of opass is to eliminate the negative influence of human factors as much as possible. Through opass, each user only needs to remember a long-term password which has been used to protect her cellphone. Users are free from typing any passwords into untrusted computers for login on all websites. Compared with previous schemes, opass is the first user authentication protocol to prevent password stealing (i.e., phishing) and password reuse attacks simultaneously. The reason is that opass adopts the one-time password approach to ensure independence between each login. So this system reduces hacking of passwords and provides much security. Future Enhancements includes: Should be implemented for other mobiles other than android, An alternative password recovery scheme should be implemented when users lose their cellphones before reissual of them.

REFERENCES

[1] Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin, oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks, IEEE Transactions of Information Forensics and Security, Vol 7 No 2, April 2012, pp. 651-663

[2] J. Thorpe and P. van Oorschot, "Towards secure design choices for implementinggraphical passwords," presented at the 20th. Annu. Computer Security Application Conf., 2004.

[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] 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.

[5] 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.

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

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

[8] Adrian perrig, Diana song, "Hash visualization: A new technique to improve real world security, computer science department, Carnegie mellon university, 2010.