CHAPTER 6

PROOF OF CONCEPT IMPLEMENTATION

This chapter discusses the Proof of Concept Implementation (POC) of all the proposed schemes discussed in chapters 3, 4 & 5. The POC is named as STARS. The objective behind this implementation is twofold. Firstly, the survey of current research in Smart Card Based Two Factor Authentication without Verifier Table has not seen any implementations that could evaluate the practicality, usability and security of such schemes. Secondly, most of the existing Graphical Password Based Authentication implementations are single-factor and requires verifier table. Therefore, it is inevitable that an authentication platform be provided which shall prove how Two-Factor Authentication (be it graphical password based or text password based) is accomplished without maintaining a verification table at the server. Moreover, once the concept is successfully implemented the usability and security can be easily verified.

STARS authentication offers five variants of it using the proposed three schemes i.e. textual password, graphical password and text-o-graphic password. The graphical password and text-o-graphic password schemes are implemented with two variations each i.e. I-27, I-16 and S-27, S-16 respectively. STARS provide users with the flexibility of freely choosing any of the password schemes to set his password. He can later change the scheme by simply switching to another scheme and setting
his password. Hence, the user, once registered with a scheme (say textual password) can later switch to another scheme (say graphical password) using password change phase.

STARS is implemented using various Java technologies to provide high end security both at client and server. The client side scheme computations are implemented using Java Script and Applets, where as for computations on server and secure exchange of messages and data between pages, client & server, Servlet, JSP and EJB are used. JBOSS 4.2.2.GA is used as application server and PostgreSQL 8.3 as database. ‘STARS’ is platform independent and therefore can run on any OS, but as part of this research the deployment is done on MS Windows server.

Registration Process: Note that as per the proposal, the registration process of all the authentication schemes run in secure environment such as HTTPS. Moreover, as all the proposed schemes require smart card as the second factor; each system should have the card reader attached to it for successful authentication. But since smart card readers cannot be found on every system, this additional factor becomes a constraint for the deployment of these schemes for wide range of applications.

To overcome this limitation and to achieve usability, it is decided to develop STARS with an aim of providing the following options to users:

1. If a user generally accesses his account from different systems or if he wishes to have any external hardware based second factor then
he can use any USB / external device as a replacement of smart
card to store the credentials. For this, he has to choose ‘Public’
option during registration, so that the server provides a
downloadable link of the virtual smart card file to be stored in
external device. This is depicted in Figure 6.2.

2. If the user always accesses his account from his personal system
then he can choose the ‘Personal’ option which creates a virtual
smart card after successful registration. The virtual smart card is a
kind of soft token (a text file in this POC) which the server creates
in order to store the user credentials which would otherwise be
written on the smart card. This would be stored at some secret
location which is not even known to the user.

3. If the user wishes to register from Public PC and does not provide
the external device, then STARS will create a cookie containing the
user credentials and stores it in temporary location for 6 hours.
After the time limit, it will be automatically deleted from the
system.

One may ask that if the cookie is deleted (deliberately /
accidentally) then how a user can access his account as he does
not possess the second factor in any form. To answer this, the POC
provides a backup mechanism at the server which maintains
copies of all the smart cards issued to users. Thus, if a registered
user wants to login but does not possess the virtual smart card
then the system detects it automatically, and throws challenge
questions to be answered. These are the questions which were asked during the user registration phase. If the verification of answers is successful, then the server sends to client, a copy of the virtual smart card.

Figure 6.1 shows the screen shot of first step of STARS-27 registration process. Here, the user has to first choose the type of authentication scheme from the given three options i.e. Textual and Image (Text-o-graphic), Only Image (Graphical) and Only Text (Textual) password based schemes.

![Figure 6.1 Screen shot of STARS-27 Registration](image1)

Once the login type is selected, the user now has to choose the type of second factor through the given two options i.e. Personal and
Public. After selecting the appropriate options, user will now check the availability of the entered user ID by clicking the “Check Availability” button. This feature is implemented using AJAX to reduce the time for communicating with server.

*Login & Authentication Process:* Whenever a user wants to login, he sends ‘request for login’ to server by clicking on the login link provided on the home page. The server responds to the request and checks if the virtual smart card is stored in the secret location; if it finds, it proceeds further by asking the user to enter his ID. The screen shot of this is shown in figure 6.3. If the server could not locate the virtual smart card, then it assumes that the user possesses a USB or external device, and accordingly it asks the user to locate the USB device shown in figure 6.4.

![Figure 6.3 Login Page displayed if Virtual Smart Card is found](image1)

![Figure 6.4 Login Page displayed if the second factor is external device](image2)
Note that the Registration and Authentication process discussed above is same for all the variants of the proposed authentication schemes. Therefore, in the subsequent sections the process of registration and authentication in respective schemes with personal option will be discussed.

6.1 TEXTUAL PASSWORD BASED AUTHENTICATION

The POC implementation of Textual Password based Scheme discussed in chapter 3 will be discussed here in this section.

6.1.1 Registration

The initial steps of registration are same as discussed earlier. Once the user ID is accepted by the server, the user sets his password on the password page as shown in figure 6.5.

![Set Password page](image)

**Figure 6.5** Set Password page

Now the user has to click on two buttons i.e. Submit and Next. When he clicks submit button, the applet (grey color component in figure 6.2) computes hash of password and when Next is clicked the client sends it to the server for server side computations of registration phase.
Upon receipt of this, the server asks for the registration details from the user in a registration form shown in figure 6.6. These details include user personal information and few secret questions which are to be answered to recover the smart card file or to reset the password, the details of which will be discussed later.

**Figure 6.6** User Profile page

Once the registration computations are completed, the server stores the virtual smart card in a secret location known only to server. The server then displays the registration successful message as shown in figure 6.7.

**Figure 6.7** Registration Success Page
6.1.2 Authentication and Key Agreement

Recall that the Textual Password scheme employs the DHKE to securely share the random number required for computations at client and server. So, when the user submits request for login, the server, after checking the availability of virtual smart card, sends the login page along with the public values as per DHKE to the client system. Once the user enters the ID and clicks submit button, the client generates its own public parameters and sends it to the server. Upon receipt of public values from client, the server sends the password page to the client as shown in figure 6.5 of section 6.1.1.

Once the user submits password, the server validates the user and upon successful authentication, both client and server generates a dynamic session key which will be used for further communication. The dynamism of this key is based on the randomness of the numbers being exchanged between client and server using DHKE. The figure 6.8 shows the screen shot of successful authentication displaying the message along with the session key.

![Figure 6.8](image)

Figure 6.8 Authentication & Session Key Generation page
6.2 GRAPHICAL PASSWORD BASED AUTHENTICATION

In this section, the POC implementation of Graphical Password based Scheme proposed in chapter 4 is presented. It will cover the implementation of two of the variants of graphical passwords which are named as iP27 and iP16. Note that, the scheme will be same for both variants except a few differences in the way the password is computed and the way in which the images are displayed. Also, the initial steps required in choosing the second factor type will be same for both the variants in registration and login phase.

6.2.1 Registration

In iP27, after the availability of user ID is checked, the server has to send 27 images to the client which are to be displayed on three 3x3 grids. For this, the server proceeds as follows:

- Randomly picks 27 images from the database
- Creates temporary Image IDs’ ranging from 1 to 27.
- Maps the original image IDs with the temporary IDs
- Displays the images on client in such a way that ID’s from 1 to 9 are displayed on Grid 1, 10 to 18 are displayed on Grid 2 and 19 to 27 are displayed on Grid 3.

In case of iP16 the server randomly picks 16 images which are mapped to temporary ID’s from 1 to 16 and are displayed on a single grid. The screen shots of iP27 and iP16 password set page are shown in figures 6.9 and 6.10.
Now the user chooses three images as password in both iP27 and iP16. The only difference is that, in iP27 the user has to pick one image from each grid making the count to 3. Once the user submits his password selection and the user profile form (figure 6.6), the client...
system computes user’s password by performing XOR of message digest of the selected three images and proceeds differently as follows:

In I-27, the client randomly picks six images in addition to the selected one from each grid and sends the image ID’s of these 21 images along with the computed password $PW_i$ to the server. Upon receipt of this, the server first computes the user credentials and personalizes the smart card with these credentials. It then creates a user profile which stores the $h(ID_i)$ and the received 21 image ID’s mapped to the original ones. The other registration details such as personal information and secret question and answers are also stored in user profile in message digest form.

I-16 differs from I-27 in the process of choosing the decoy images, herein the client randomly picks 14 images ID’s and sends these along with the password. The rest of the computations and profile creation process is same as discussed above.

The registration session is completed with the registration successful message from the server being displayed at client

### 6.2.2 Authentication & Key Agreement

When the user requests for login, the server sends the login page along with its public key as per the scheme. User then enters his ID which is locally validated with the parameters stored in smart card. Here the login is designed such that unless the user ID is verified with the
contents of smart card, he cannot proceed further to get the password page. After the successful verification of user ID, the client generates a random number using the random number generation function which takes the seed value as the current timestamp. This number is used to compute $R_i$ which is then encrypted along with $P_i$ and $h(Y_i)$ using the public key of the server. The public key encryption is accomplished using RSA. The client then sends this to the server.

Upon receipt of the request for password page, the server first decrypts it using its private key and after validating the user ID and creating a challenge it picks images from the user profile and database. So, in I-27, the server picks the 21 images as per the image ID's stored in user profile and the remaining six images are randomly picked from the image database. In I-16, fourteen images are taken from the user profile and the remaining two are picked from the database. All the images are again mapped to temporary ID's before sending it to the client.

Upon receipt of the images in encrypted form, the client first decrypts it using the pre-shared key discussed in scheme and checks the freshness of the random number so that the replay and phishing attack is resisted. If the random number does not match the computed one then the client will not display the images and thus terminates the session. During the period of validating the random number, the images are temporarily stored at client. If the validation is successful, then the client displays the images on the grid(s) as shown in figures 6.9 and 6.10.
Once the user submits password, the client locally verifies it with the contents of the smart card and then creates $C_i$ which is sent to the server as challenge. Upon receipt, the server validates $C_i$ and if successful, both client and server generates a dynamic session key which will be used for further communication. Figure 6.8 in section 6.1.2 shows the screen shot of successful authentication, displaying the message and the session key.

6.3 **TEXT-O-GRAPHIC PASSWORD BASED AUTHENTICATION**

This section presents the POC implementation of the Text-o-Graphic Password based Scheme proposed in chapter 5. The proposed Text-o-Graphic password is based on the concept of combining text and image where in a user has to write any description about the chosen image. The concatenation of message digests of text and image becomes user’s password. The proposed text-o-graphic password does not fall directly into the categories of Recall based and Recognition based Graphical Password techniques but it could be a type of a recall based recognition technique.

The proposal has two variants of the text-o-graphic password which are named S-27 and S-16. In S-27, user chooses one image from each grid and associates a description with that image. This is done for the displayed 3 image grids. Therefore, the password will be the hash of concatenation of text and image message digests from all the three grids. In S-16, the user has to choose one image from the given sixteen images
and associate a description with that image. The hash of XOR of the text and image message digests becomes the user password.

6.3.1 Registration

The registration process for picking the images and displaying it at client system in S-27 and S-16 is almost same as I-27 and I-16 discussed in section 6.2.1 except the method of setting the password. Herein the user has to write description along with the image. The screen shots of password page for S-27 and S-16 are shown in figures 6.11 and 6.12.

Once the password is set by the user, the client computes the password as discussed in section 5.1.1 and 5.2 for S-27 and S-16 respectively. In S-27, the client picks 21 images and in S-16, the client picks 14 images randomly and sends these along with the computed password for further registration computations at server.

After successful registration, the server issues the smart card to the user and displays a registration successful message on the client as shown in figure 6.7 of section 6.1.1.
6.3.2 **Authentication & Key Agreement**

Recall that S-27 employs the DHKE protocol for secure passing of the random secret where as S-16 is based on PKI where in the user and the server needs to have digital certificates for secure exchange of
messages. Both the schemes will be discussed separately in the following paragraphs:

_S-27:_ When a user requests for login to S-27, the server, after checking the availability of virtual smart card, sends the login page along with the public values as per DHKE to the client system. Once the user enters the ID and clicks submit button, the client generates its own public parameters and sends it to the server along with $R_i$ as challenge. Upon receipt of $R_i$ and $B_1$, the server verifies the validity of $R_i$ and on successful verification, it picks the 21 images based on the user profile image IDs and 6 randomly from the image database and maps them to temporary IDs. The server also computes $T_i$ and a new set of public parameters as per DHKE and sends it to the client.

The client verifies the validity of $T_i$ and if valid displays the received images as shown in figure 6.1.1. At each login attempt, the images are randomly displayed in grids so as to reduce the social engineering related threats. The user submits the password by identifying his password image and enters the image number and its description in the respective fields. This is repeated for all the three grids. Upon submission of the password, the client computes the password as discussed in section 5.1.1. It then computes $C_i$ and public parameters as per DHKE and sends it to server for mutual authentication. The server verifies the validity of received $C_i$ using the public parameters and its secret key, and if validation is successful, both client and server generate a session key to be used for encrypted communication.
S-16: Upon receipt of the user’s login request, the server sends user, the digital certificate and the login page. The client creates a random secret $P_i$ encrypts it with the public key of server and then encrypts $R_i$ and the above result with user’s private key. If the private key is protected with pin then the client asks the user for the pin. After the computation, the encrypted message is sent to the server along with user’s public key. The server decrypts it and verifies the validity of user ID and accordingly retrieves 14 images from user profile and two from the image database. It also computes $T_i$, encrypts $T_i$ and images with user’s public key and sends it to client. The client checks the freshness of $T_i$ and displays the images in a single grid as shown in figure 6.12.

The user submits the password by identifying his password image and enters the image number and its description in the respective fields. Upon submission of the password, the client computes the password as discussed in section 5.2. It then computes $D_i$ and $C_i$ as per the scheme and sends it to server for mutual authentication. The server verifies the validity of received $C_i$ using $D_i$ the, and if validation is successful, both client and server generates a session key to be used for encrypted communication.

6.4 CHANGE PASSWORD AND FORGET PASSWORD

This section presents the change password phase of all the proposed schemes that provides the facility of changing the password freely in a secure way. In the implementation, the flexibility of choosing
any of the authentication schemes along with the login type ‘personal’ or ‘public’ is provided to user. For example, a user who was earlier registered with textual password based scheme with external device as the second factor wants to switch to Graphical password based scheme with virtual smart card as second factor can do so with the change password option. This flexibility is extended to the cases wherein the user forgets his password, but here the user first needs to answer the secret questions.

6.4.1 Change Password

The password change phase is invoked when a user requests for change of password by clicking the ‘Change Password’ button. Note that password change phase of all the schemes is same except the client side computations of $N_i$ or $V_i$ parameters. Moreover, all the schemes are designed such that the change password resultant parameters to be updated in the smart card can be done at the client side itself. Thus, allowing the user to freely change password anytime anywhere by simply connecting his external device. Also recall that the change of password is possible only after successful login which means that this phase runs in a secure way.

When a user requests for change of password and submits his request, the client generates a random number which is encrypted with the authenticated session key and sends to the server. Upon receipt, the server decrypts it and checks the freshness of nonce. It then sends the password change option page along with nonce in encrypted form. When
it is received, the client, after checking the freshness of nonce displays the option page as shown in figure 6.13

![Password Change option page](image)

**Figure 6.13** Password Change option page

The user now chooses the option and submits it to the server. Based on the option selected, the server follows the process of setting the password. Once the password is submitted by the user, the client performs password change computations as proposed in the respective schemes. The client then updates the smart card with the newly computed parameters.

If the user had selected the Graphical or Text-o-Graphic password option, then after updating the smart card the client randomly picks the images (21 or 14 based on user’s selection of S/I-27 or S/I-16 respectively). These images are encrypted along with the random number using the authenticated session key and send it to the server. Upon receipt, the server decrypts it and after checking the freshness of nonce, updates the user profile with the new set of profile image set.
6.4.2  Forget Password

The feature of change password is extended to forget password as well to allow the user to freely choose any of the authentication schemes when resetting the password.

This phase is invoked when the user request to reset the password by clicking the forget password link. When the user forgets the password and requests for password reset, the system asks him to enter the userID as shown in figure 6.14. Upon receipt of this request, the server randomly picks three of the five secret questions based on userID which were asked to him at registration time. These questions are given to the user as a challenge as shown in figure 6.15.

![Figure 6.14 Forget Password request page](image)

![Figure 6.15 Secret Questions page](image)
When the questions are answered and submitted by the user to server, it verifies the validity of it. If the validation is successful, then the server provides the user with the password reset page consisting of all those options which are give to user at registration time. This is shown in figure 6.16

![Password Reset page](image)

**Figure 6.16** Password Reset page

Once the user resets his password and submits it, the client performs password change computations as proposed in the respective schemes. The client then updates the smart card with the newly computed parameters.

### 6.4.3 Loss of Smart Card / Deletion of Virtual Smart Card

In the real world, if an ATM card or credit card is lost, the owner of the card has to immediately report the incident to bank to deactivate the lost card and request for issuance of new card. In web world also, the smart card may be lost or the virtual smart card may be deleted accidentally or deliberately by the user or others. Hence, STARS is designed such that, if the user looses the smart card or accidentally
deletes it, then he cannot access his account. Therefore, there should be a mechanism to recover the lost card thus recovering the access of the user to his account.

To deal with such situations, STARS provides a backup mechanism which maintains a backup of all the smart cards issued to every user. Each backup file is named after the user ID.

If a user has lost the smart card or his virtual smart card is deleted, the server asks him to enter his ID. When the user submits the ID the server presents the user with three challenge questions out of five that were asked at registration time as shown in figure 6.15. The user submits the answers to the server, which after successful validation, sends a copy of smart card backup to the user as shown in figure 6.17. The user can now proceed with normal login process to access his account.

![Figure 6.17](image)

In case of roaming users who does not wish to maintain a hardware based second factor, the above discussed feature will help
them but will be bit inconvenient to go through secret question validation every time before the actual login process. But here it is suggested that such users make sure that cookies and temp files are deleted from the public system once their session is completed.

6.5 SECURITY ANALYSIS OF STARS

The security analysis of the proposed schemes is presented in Security Analysis sections of the respective chapters. In this section, the security analysis of STARS against possible threats on such implementations is presented.

6.5.1 Storage of Registration Data

One of the research objectives of all the proposed schemes is to design a Two-Factor Authentication scheme which does not maintain any verifier table at server. In all variants of STARS a user profile is maintained, which stores user IDs, secret questions, image portfolio data (not applicable in textual password based scheme) and other registration details. Among these, the user ID, answers to secret questions are stored in message digest form so that even if the attacker gets access to the database he cannot figure out the answers of the secret questions the user has set. The image portfolio stores the pointers to 21 and 14 images in the database for S-27 and S-16 respectively, making the password images camouflaged with decoy images. If the attacker gets access to these portfolio images then it is difficult for him to correctly guess the
password which eventually is combination of three images. In case of text-o-graphic password, if the attacker gets access to the image portfolio then it is more difficult to crack than graphical password because the user password is the combination of text and password. Above all, to perform unauthorized login successfully, the attacker must have the smart card of the user in addition to successful guessing of the password.

6.5.2 Security of Password Computation

In graphical and Text-o-Graphic password computations during registration phase, when the user submits his password, the client computes the XOR of message digest of all the images. The result is then subjected to hash function to get the final message digest. Now the client sends this message digest to the server along with the randomly chosen image IDs. Since the computations are done at client side, STARS provides security to such computations by implementing it using applets which do not reveal the client side computation code and intermediate values. Therefore, this prevents the attacker from getting the intermediate values which could be used by him to perform any attack. All the computations at client side in the remaining phases also are implemented using applets, EJB and JSP so as to prevent the attacker from unauthorized access to the protocol computations.
6.5.3 Security of Cookie

Recall that if a user selects ‘Public’ and registers to STARS without providing any external device, then the server delivers the smart card in the form of a cookie in that system. It may be noted that, the user might have chosen Public because he might be registering from public PC such as cyber café system. Therefore, in such cases, the server sets the cookie validity time as three hours after which the cookie automatically gets expired and hence deleted.

One may ask that during the three hour period an attacker may access the cookie after the registered user left the system without deleting the cookie. In such a case, the attacker may get all the parameters such as $V_i$, $Q_i$ etc. stored in it. But since the $V_i$, $Q_i$ etc values are message digests, the attacker could not get the reverse of it. Moreover, the computation of $V_i$ requires the attacker to have the knowledge of server’s master secret along with other secret values.