Minimum Space and Huge Security in 3D Password Scheme

Prof. Sonkar S.K
Research Scholar JJTU
India

Dr.Ghungrad.S.B
(Principal)
St. Xavier Technical Institute,
Mahim, Mumbai, India

ABSTRACT

The 3-D password is a multifactor authentication scheme. For the authentication, it is require to presents a 3-D virtual environment where the user navigates and interacts with various objects. The sequence of actions and interactions toward the objects inside the 3-D environment constructs the user's 3-D password. The 3-D password can combine most existing authentication schemes such as textual passwords, graphical passwords, and various types of biometrics into a 3-D virtual environment. The design of the 3-D virtual environment and the type of objects selected determine the 3-D password key space. As per the reference[1] the author tells that the resulting 3D password space is very huge. In this paper we shows that the space is reduced and increase the security.

Keywords— Authentication, biometrics, graphical passwords, textual passwords, 3-D passwords, 3-D virtual environment.

1 .INTRODUCTION

1.1 History of authentication system

Textual passwords are commonly used, but users do not follow their requirements. Users tend to choose meaningful words from dictionaries, which make textual passwords easy to break and vulnerable to dictionary or brute force attacks. [2]

Graphical passwords are based on the idea that users can recall and recognize pictures better than words. On the other hand, some of the graphical password schemes require a long time to be performed. Moreover, most of the graphical passwords can be easily observed or recorded while the legitimate user is performing the graphical password; thus, it is vulnerable to shoulder surfing attacks. Currently, most graphical passwords are still in their research phase and require more enhancements and usability studies to deploy them in the market. Many available graphical passwords have a password space that is less than or equal to the textual password space.

Many authentication systems, particularly in banking, require not only what the user knows but also what the user possesses (token-based systems). However, it is observe that tokens are vulnerable to fraud, loss, or theft by using simple techniques.

Numerous biometric schemes have been proposed, fingerprints, palmprints, hand geometry, face recognition, voice recognition, iris recognition, and retina recognition are all different biometric schemes. Each biometric recognition scheme has its advantages and disadvantages based on several factors such as consistency, uniqueness, and acceptability. One of the main drawbacks of applying biometrics is its intrusiveness upon a user's personal characteristic. Moreover, retina biometrical recognition schemes require the user to willingly subject their eyes to a low-intensity infrared light. In addition, most biometric systems require a special scanning device to authenticate users, which is not applicable for remote and Internet users.

1.2 3-D Password Scheme

The 3-D password is a multifactor authentication scheme. For the authentication, it is require to presents a 3-D virtual environment where the user navigates and interacts with various objects. The sequence of actions and interactions toward the objects inside the 3-D environment constructs the user's 3-D password. The 3-D password can combine most existing authentication schemes such as textual passwords, graphical passwords, and various types of biometrics into a 3-D virtual environment. The design of the 3-D virtual environment and the type of objects selected determine the 3-D password key space. *[1]*

This new authentication scheme has the following requirements:

1) This authentication scheme should not be either recall based or recognition based only. Instead, the scheme should be a combination of recall-, recognition-, biometrics-, and token-based authentication schemes.

2) Users should have the freedom to select whether the 3-D password will be exclusively recall-, biometrics-, recognition- or token-based, or a combination of two schemes or more. This freedom of selection is necessary because users are different and they have different requirements. Some users do not like to carry cards. Some users do not like to provide biometrical data, and some users have poor memories. Therefore, to ensure high user acceptability, the user's freedom of selection is important.

3) This new scheme should provide secrets that are easy to remember and very difficult for intruders to guess.

4) This new scheme should provide secrets that are not easy to write down on paper. Moreover, the scheme secrets should be difficult to share with others.

5) This authentication scheme should provide secrets that can be easily revoked or changed.

2. MATERIALS AND METHODS:

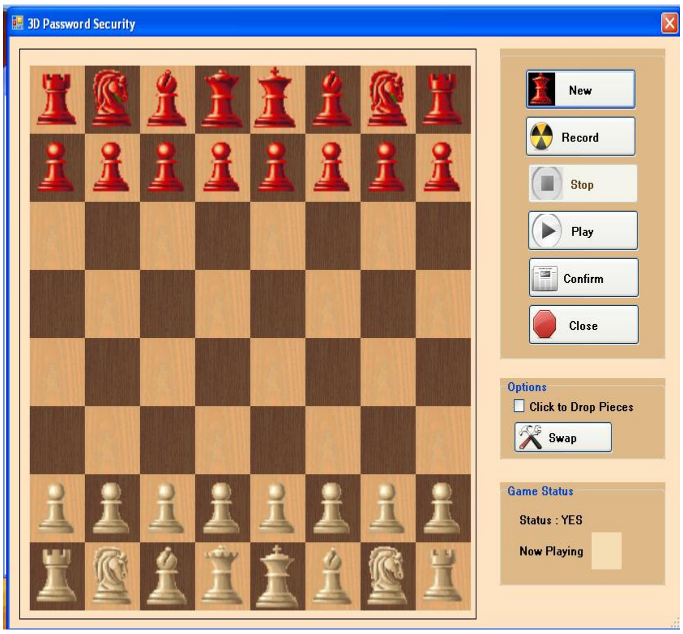
Here we design the two 3D environment with second environment is having the two options.

2.1. Environment 1 – Chess:-

First is Chess games in which user construct the 3D password by moving the chess pieces in valid places on chessboard. Second environment is Cube in which user construct the 3D password by moving the Cube left, right, up, down and by rotating the Cube along with option of placing the input images on each side of Cube. In second environment the Cube without no image input in which space required to store 3D password is very less which is advantage for reducing the password space and increase in security.

In the registration form after filling all the details of new user, user must have to click on environment1 button to select this environment. The figure 4.1.2 shows that the snapshot of Enviornment1.This particular environment is for chess game. In which it has the total 32 objects – 16 White objects and 16 Red objects. And also it includes total seven buttons such as – New button, Record button, Stop button, Play button, Confirm button, Close button and Swap button, and one checkbox option.

The working of each button is as follows:



□ Click to Drop Pieces

Status : YES

Figure.1: Enviornment1 (Chess)

New button: Whenever the user is click this button then the all the objects (white and red) are initialized. Before clicking on this button the environment is empty.

Swap button: If user wants to change the position of the Red and White objects. This can be done by using this button. In other words it simply swaps the location of White and Red objects respectively.

Record button: when user wants to create his/her 3D password it is compulsory to click this button and then all the sequence of actions and interactions are stored as 3D password as a string. If the user is not clicked on the record button then the nothing is recorded and the error is occurred when the stop button is clicked.

Stop button: For ending the sequence of actions and interactions user must have to click on this button. This button stops the recording action and the recorded sequence of action and interaction are saved as a 3-D password in the form of string.

Play button: After clicking the stop button, if the user wants to check what he/she has performed as an action and interaction then for that user is require to click on play button. After clicking the play button then user can see the sequence of action and interaction that he/she has recorded as a 3-D password.

Confirm button: Once the user click on this button he/she can not change his/her 3-D password. It means that user can change his/her 3D password before clicking on this button by using new button.

Close button: After clicking on this button the environment is close and control pass to the registration form.

2.2 Environment 2 – Cube:-

In this dissertation the second environment is a cube. Fig. 4.1.3 shows the snapshot of enviornment2. Whenever user is selecting the enviornment2 then the cube is at initial position which is already setted at (400, 240, 0) co-ordinates with respect to X, Y, Z axis. And one more point that setted in this environment in the form of camera position. This camera position is set at co-ordinates (400, 240,-500) on X, Y, Z axis and acts as a reference point and from this point user can observe the action and interaction that are performed on the cube. This environment has four main actions; each main action has six sub actions and also having the one particular Input action as load image on each side of cube

The detail of the four main actions is as follows:

Move Cube: This particular main move cube action having the six different sub actions that are- Left, Right, Up, Down, In, Out. Whenever the user is single click on these buttons then the cube moves by 45 co-ordinates with respective to which button is click. The maximum click on each button is six. When a user is clicked on any particular button at seventh time then he/she got the error message as “you have reached the maximum limit”.

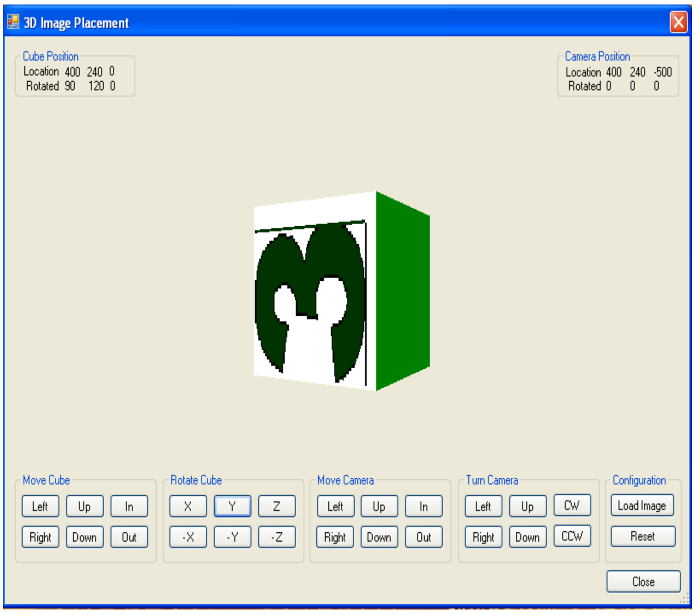
Rotate Cube: The next main action is rotate cube with sub actions that are – rotate cube x-direction, y-direction, z-direction and –x -direction, -y-direction, -z-direction. Whenever the user single clicks on these buttons then the cube rotate in 45° direction with respective to which button is click. The maximum click on each button is six. When a user is clicked on any particular button at seventh time then he/she got the error message as “you have reached the maximum limit”.

Move Camera: Move camera action also having different subaction that is - Left, Right, Up, Down, In, Out. When the user is single click on these buttons then the camera or reference point moves 45 co-ordinates with respective to which button is click. The maximum click on each button is six. When a user is clicked on any particular button at seventh time then he/she got the error message as “you have reached the maximum limit”.

Turn Camera : Turn camera action with different subaction as to rotate camera Left, Right, Up, Down, CW(Clockwise), CCW(Counter clock -Wise) direction. When the user make single click on these buttons then the camera rotate by 45° in direction with respective to which button is click. The maximum click on each button is six. When a user is clicked on any particular button at seventh time then he/she got the error message as "you have reached the maximum limit".

Load Image: This action is used to load image on each side of cube. This will make user 3D password stronger.

User can perform any number of action and interaction on the cube and at the end for to save these action and interaction as a 3-D password, user is require to click on Close button.



3D Image Placement

Cube Position
Location 400 240 0
Rotated 90 120 0

Camera Position
Location 400 240 -500
Rotated 0 0 0

Figure 2 : Snapshot of Enviornment2 – Cube

3. RESULTS & DISCUSSION

3.1 3-D password space size:

To determine the 3-D password space it required to count all possible 3-D password that have certain number of action interaction and inputs towards all objects existing in environment. Now we are calculating password space for different environment for this dissertation.

3.1.1 Enviornment-1(Chess)

Now proposed scheme are calculating the password space considering that user wants to move a single object when the environment is appeared. Let's say you start with a chess board set up for the start of a game. Each player has 16 pieces. Let's say that white starts first. White has 20 possible moves:

1. The white player can move any pawn forward one or two positions.
2. The white player can move either knight in two different ways.

The white player chooses one of those 20 moves and makes it. The equation for calculating the password space

$$\Pi\,(L_{max}, G) = \sum_{n=1}^{n=Lmax} (m + g\,(AC))^{n}.$$

Here,

m → All possible actions and interaction towards all existing objects. For our consideration as discussed above the value of m is 20.

g(AC) → Count of total number of action, input towards the environment. For our consideration as discussed above action is only one i.e. moving the object and the interactions are 3(moving pawn forward, moving either knight in two different ways). So the value of g(AC) is 3.

G → *(G ×G×G)* Number of actions, interactions and inputs, for consideration action is only one i.e. move, interactions are 3 and inputs are nil. So the value of G is 3.

Lmax → The maximum length of password Here L_{max} =17. Then the possible password space for our consideration is

$$\Pi\,(17, 3) = \sum_{n=1}^{n=17} (20 + 3)^{n}$$

$$= 3.7714\times10^{43}$$

The above value gives total number of space (in byte) required to store the password for environment-1.

3.1.2 Enviornment-2(Cube):-

In environment-2 proposed scheme create password by moving, rotating, zooming the cube. For creating password there are four different actions i.e. moving cube, rotating cube, moving camera, rotating camera along the x, y, z axis. And for each action user can perform the six different interactions.

The terms to calculate password space for environment -2 are: *G – (G ×G×G)* → number of actions, interactions and inputs. Actions – 4 (.moving cube, rotating cube, moving camera, rotating camera), Interactions – 6

Input- 6 (Placing an image on each side of cube)
So, G = G×G×G = 4×6×6 = 144

m → All possible actions, interactions towards all existing objects in environment.

For Proposed scheme environment is, for each action we have total 36 interactions so total possible interactions are
m = 1679616.

Lmax → Maximum length of password, for this environment by taking the input i.e. the images on each side of cube having the name six characters wide then the value for Lmax is 111.

g (AC) → Count of total number actions and interactions towards virtual environment.

For this environment it is 24 (6 × 4) Now, the password space for this environment is

$$\Pi\,(L_{max}, G) = \sum_{n=1}^{n=Lmax} (m + g\,(AC))^{n}.$$

After placing the values

$$\Pi\,(111, 144) = \sum_{n=1}^{n=111} (1679616 + 24)^{n}.$$

The value of equation 4 gives the total number of space (in byte) required to store passwords for environment-2.

3.1.3 Enviornment-2(Cube without no image input):-

Now we are calculating the password space without taking the input.

Therefore,

G → (G ×G×G) – number of actions, interactions and inputs. Actions – 4 (.moving cube, rotating cube, moving camera, rotating camera)

Interactions – 6 . Input- Nil

So, G = G×G = 4×6 = 24

m → All possible actions, interactions towards all existing objects in environment. For our environment is, for each action we have total 36 interactions so total possible interactions are,

m = 1679616

Lmax → maximum length of password, for this environment, Lmax is 8

g (AC)→ count of total number actions and interactions towards virtual environment. For this environment it is 24 (6 × 4).

Now, the password space for this environment is

$$\Pi (L\max, G) = \sum_{n=1}^{n=L\max} (m + g\ (AC))^{n}$$

After placing the values

$$\Pi (8, 24) = \sum_{n=1}^{n=8} (1679616 + 24)^{n.}$$

$$=1.47446\times10^{20}$$

The above value gives the total number of space (in byte) required to store the 3D password for environment-2 without input.

3.2 Comparison: Text and 3D password

Here it had compared the text password and proposed scheme. The comparison is in between the length of text password and the action and interactions with 3-D objects in virtual environment. Firstly, it have taken the text password length as one character and a single action on 3-D object which is present in enviornment1 and enviornment2. The following table shows Comparison between the length of text and 3-D password for virtual environment1 and enviornment2.

Table 1. Comparison between Text and 3-D Password

No, of Action/ Charac-ter	Encrypt-ed 3D Password Size of Env-1 in Byte	Encrypted-d Text Password size in Byte	Encrypted 3D Password Size of Env-2 in Byte	Encrypted 3D Password Size of Env-2 with no image in Byte
1	23	2	18	8
2	39	3	19	8
3	55	4	22	8
4	71	6	23	8
5	87	7	24	8
6	103	8	26	8
7	119	10	28	8
8	135	12	31	8
9	151	13	34	8
10	167	14	36	8
11	188	15	39	8
12	202	16	42	8
13	218	18	44	8
14	236	19	39	8
15	247	20	43	8
16	263	22	47	8
17	283	23	51	8
18	300	24	55	8
19	319	26	59	8
20	340	27	63	8

No, of Action/ Charac-ter	Encrypt-ed 3D Password Size of Env-1 in Byte	Encrypted-d Text Password size in Byte	Encrypted 3D Password Size of Env-2 in Byte	Encrypted 3D Password Size of Env-2 with no image in Byte
21	354	28	50	8
22	370	30	55	8
23	391	32	60	8
24	402	33	66	8
25	418	34	71	8
26	428	35	76	8
27	439	36	82	8
28	468	38	60	8
29	476	39	67	8
30	487	40	74	8
31	522	42	80	8
32	530	43	87	8
33	546	44	94	8
34	572	46	100	8
35	604	47	71	8
36	612	49	79	8
37	620	50	87	8
38	628	52	95	8
39	652	53	103	8
40	660	54	111	8

Table 1 shows the comparison of password space required for text and 3-D password for enviornment1, enviornment2 and environment2 with no images. Proposed scheme compared 40 different records by taking length of text password from one character to fourty characters and single action to fourty actions on 3-D objects in environment1 and enviornment2.

3.3 Performance result in Graphs*:*

In the fig. 5.1 shows that the Blue line shows the password space required for 3-D Env-1 and the Yellow line password space for Env-2 with images and the Green line shows the password space required for Env-2 with no images whereas the Pink line shows the password length for Text Password.

4. CONCLUSION

In 3D password scheme as number of sequence of action and interaction in 3D virtual environment increases then the length of the password is also increases. Therefore to store 3D password space required to store in memory is very large as compared with textual password. In this paper shows that two different 3D virtual environments are analyzed to minimize the space required to store the 3D Password. First environment is Chess game in which user construct the 3D password by moving the chess pieces in valid places on chessboard. Second environment is Cube in which user construct the 3D password by moving the Cube left, right, up, down and by rotating the Cube along with option of placing the input images on each side of Cube. In second environment Cube without no image input in which user can perform more number of action and interaction as compared with first environment and it is observed that space required to store the 3D password is comparatively very less, and the password created is very strong. And security is also increases. Therefore for to minimize the space Required to store the 3D password depends upon well designed 3D virtual environments.

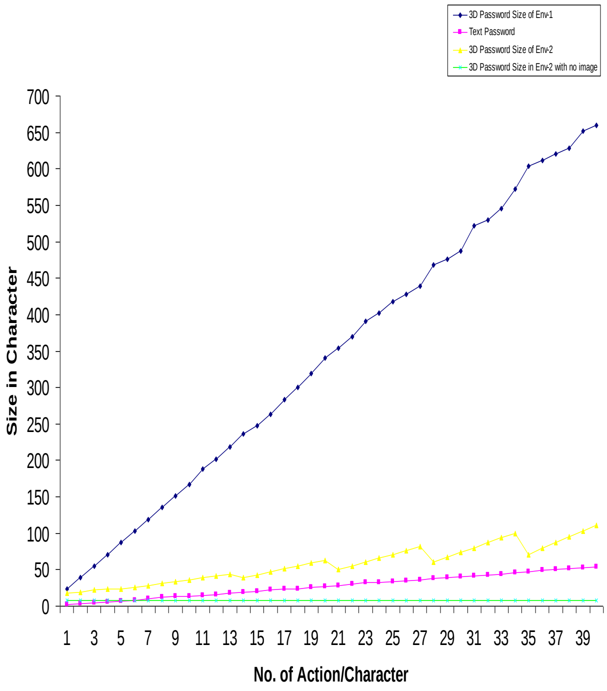


Figure 5.1: comparison of Text and 3-D password for Env-1 & Env-2.

5. REFERENCES

[1] Alsulaiman, F.A.; El Saddik, A., "Three- for Secure," IEEE Transactions on Instrumentation and measurement, vol.57, no.9, pp 1929-1938.Sept. 2008

[2] D. V. Klein, “Foiling the cracker: A survey of, and to passwords security,” in Proc. USENIX Security, , pp.–14

[3] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, “The design and analysis of graphical passwords,” in Proc. 8th USENIX Security Symp,Washington DC, Aug.1999, pp.1-14.

[4] X. Suo, Y. Zhu, and G. S. Owen, “Graphical passwords: A survey,” in Proc. 21st Annual. Computer Security Appl. Conf., Dec. 5–9, 2005, pp. 463–472.

[5] D. Weinshall and S. Kirkpatrick, "Passwords You’ll Never Forget, but Can’t Recall," in Proceedings of Conference on Human Factors in Computing Systems (CHI). Vienna, Austria: ACM, 2004, pp. 1399-1402.

[6] L. Sobrado and J.-C. Birget, "Graphical passwords," The Rutgers Scholar, An Electronic Bulletin for Undergraduate Research, vol. 4, 2002.

[7] D. Hong, S. Man, B. Hawes, and M. Mathews, "A password scheme strongly resistant to spyware," in Proceedings of International conference on security and management. Las Vergas, NV, 2004.

[8] S. Man, D. Hong, and M. Mathews, "A shouldersurfing resistant graphical password scheme," in Proceedings of International conference on security and management. Las Vegas, NV, 2003.

[9] Two Factor Authentication for the Enterprise, http://realuser.com/realuser.