# A Text based Authentication Scheme for Improving Security of Textual Passwords

Shah Zaman Nizamani
Department of Information Technology
Quaid-e-Awam University of Engineering, Science & Technology, Pakistan

Tariq Jamil Khanzada
Department of Computer Systems Engineering
Mehran University of Engineering & Technology, Pakistan

Syed Raheel Hassan
Department of Computer Systems Engineering
Quaid-e-Awam University of Engineering, Science & Technology, Pakistan

Mohd Zalisham Jali
Faculty of Science and Technology
Universiti Sains Islam (USIM), Malaysia

***Abstract*—User authentication through textual passwords is very common in computer systems due to its ease of use. However textual passwords are vulnerable to different kinds of security attacks, such as spyware and dictionary attacks. In order to overcome the deficiencies of textual password scheme, many graphical password schemes have been proposed. The proposed schemes could not fully replace textual passwords, due to usability and security issues. In this paper a text based user authentication scheme is proposed which improves the security of textual password scheme by modifying the password input method and adding a password transformation layer. In the proposed scheme alphanumeric password characters are represented by random decimal numbers which resist online security attacks such as shoulder surfing and key logger attacks. In the registration process password string is converted into a completely new string of symbols or characters before encryption. This strategy improves password security against offline attacks such as brute-force and dictionary attacks. In the proposed scheme passwords consist of alphanumeric characters therefore users are not required to remember any new kind of passwords such as used in graphical authentication. Hence password memorability burden has been minimized. However mean authentication time of the proposed scheme is higher than the textual password scheme due to the security measures taken for the online attacks.**

***Keywords*—Password security; security; usability; alphanumeric passwords; authentication**

## I. Introduction

Despite of many weaknesses user authentication through textual passwords is widely used since long time. In textual password scheme credentials are directly inserted into login fields, which results in easy capture of password through spyware attack, and shoulder Surfing attack [1]. Other problem with textual password scheme is that users tend to set short and easy to remember passwords, such passwords are easy to break through brute force or dictionary attack [2]. Therefore users are restricted to add numbers or special characters in their passwords but such policies make the passwords hard to remember.

By recognizing the memorability and security issues in textual passwords, researchers proposed different graphical password techniques. In this category of authentication passwords are consist of some pictures, lines or x, y coordinates inside a picture. Generally graphical passwords have memorability advantage over textual passwords because visual information is easy to remember and recall than alphanumeric characters [3] [4]. While security and usability of graphical password techniques varies from one scheme to another.

Graphical password technique was first proposed by Blonder [5] in 1996, since then many graphical password techniques are proposed but none has replaced textual password scheme. Shoulder surfing and spyware attacks are common threat to different graphical password schemes. Android unlock scheme [6], is the only graphical password scheme being largely used in smart phones because the scheme is easy to use. Although this technique has many security weaknesses such as shoulder surfing attack but due to nature of the device, attackers have very little access to launch security attacks. Due to security weaknesses Android unlock scheme is not used in online systems for authentication. Secure graphical password schemes have timing and adoptability issues. Such schemes require large amount of physical and mental work to do for authentication and users have to remember different kinds of passwords that is why many usability issues arises.

User authentication can be made secure by biometric or token based authentication techniques but they require special hardware for processing. The other easy to use authentication option remains the knowledge based technique. Authentication through this technique is improved by two approaches. In first, different graphical password schemes have been proposed, while in second approach schemes are suggested by enhancing or mixing text based and graphical password techniques. In this paper second approach has been taken for improving the security of traditional textual passwords. Proposed scheme provides enhancements in the login screen and the way passwords are stored into the database. In the login screen every time user inserts a new set of numbers which

represent the password, therefore proposed scheme provides resistance from spyware and man-in-the middle attacks. In the proposed password storage technique, alphanumeric characters of a user's password are transformed to different alphanumeric characters and symbols and then stored into database. This password transformation makes harder to apply dictionary and brute force attacks.

The remaining paper is divided into six sections. In section 2 literature review is given regarding the field of user authentication. Proposed authentication scheme along with technique to store passwords are explained in section 3. In section 4 analysis of the proposed scheme is given with respect to security, usability and memorability. Proposed scheme is compared with famous authentication techniques in section 5. Finally conclusion is given in section 6.

## II. Literature Review

User authentication works on the basis of something user knows (Knowledge-based), something user has (token-based) or something user are (Biometric). Focus of this research is to design an efficient user authentication scheme under the category of knowledge-based authentication. Therefore literature review targets knowledge-based authentication. This section has been divided into two parts in first, different user authentication schemes are discussed which are related with the research work. In the second part, problems in user authentication schemes are briefly discussed.

### A. Related work

Zhao and Li [7] proposed some changes in textual password scheme for adding resistance to shoulder surfing attack and called the scheme as S3PAS. In this scheme registration process is same as textual password scheme but the login process is different. In the login screen alphanumeric characters are randomly shown in the image format and a user has to click on the logical triangles formed by the password elements or type characters which belong to each password triangle. S3PAS scheme provides resistance from shoulder surfing, keystroke logger and mouse logger attacks. Searching password triangles is time consuming task, therefore this scheme is very difficult to use. The scheme is also vulnerable to dictionary and brute force attacks.

Ziran et.al [8] proposed a text-based password scheme. In this scheme a user set password by drawing a shape inside a registration screen. In the login screen a grid filed with 0s and 1s are randomly shown, a user is required to insert a list of 0s and 1s, such that they form the shape of password. Proposed scheme provides resistance from spyware attacks but the scheme is vulnerable to brute force, dictionary and shoulder surfing attacks.

Chen et al. [9] proposed a mixed textual and graphical password scheme for resisting shoulder surfing attack. In this scheme passwords consist of some characters and numbers along with a colour. In the login screen characters and numbers are shown in circular format. Password is entered by rotating password characters in front of the colour chosen during registration. Proposed scheme does not contain symbols therefore it has small password space and password entry process requires physically efforts.

Rao and Yalamanchili [10] proposed an authentication scheme known as Pair Pass Char (PPC), in this scheme registration process is same as ordinary textual password scheme. In the login screen all alphanumeric characters are shown in 10 * 10 grid. For password entry a user has to search logical rectangles, formed by different pairs of password characters and then click on the corner characters of the rectangles. The scheme contain different rules for rectangle searching therefore the scheme is difficult to learn. Average authentication time for 6 characters password is 47.4 seconds which is quite high.

First graphical password scheme was proposed by Blonder [5]. He proposed a scheme where a password consists of certain points inside a password picture. Blonder's scheme has many security issues such as shoulder surfing attack and mouse logger attack. Wiedenbeck [11] proposed "PassPoint" scheme based upon Blonder's scheme. In PassPoint scheme users have freedom to click on any point inside the password picture, this freedom was not available in Blonders scheme. Passpoint scheme is better than Blonders scheme with respect to brute force and dictionary attacks but it is not resilient to shoulder surfing and spyware attacks.

Wiedenbeck et al. [12] proposed a shoulder surfing resilient graphical password scheme known as CHC (Convex Hull Click). In this scheme users are given multiple challenges for authentication. In each challenge users have to find out three password images and then need to click inside an invisible triangle formed by the password images. This scheme provides resistant from shoulder surfing attack but authentication time is 71.66 seconds which is quite high.

Lopez et al. [13] suggested a challenge response based shoulder surfing and spyware attack resilient graphical password scheme. In this scheme three images per row are shown in the login screen. A user has to identify whether number of password images are even or odd in different rows. The scheme is weak with respect to brute force attack because small number of images are used in this scheme. Combined screen scrapper and key logger attack become successful after multiple rounds of recordings.

Weinshall [14] proposed a recognition based graphical password scheme known as cognitive authentication scheme. It provides resistance form key logger and mouse logger spyware attacks. In the the scheme, 80 icons are presented into 8 * 10 grid based login screen. Password icons are selected by computing a path generated by the icons. Learnability and high authentication time are the issues with this scheme.

Google introduced android unlock scheme, in which nine points are given into a 3 * 3 grid based login screen. Password of the scheme consists of some lines inside the grid. This scheme is very easy to use but the passwords can be captured by shoulder surfing attack and the scheme also provides low password space [6]. Microsoft introduced a graphical password scheme in windows 8, in which passwords consist of some points, lines or circles inside a picture. This scheme is also very easy to use but it has Hot-Spot and shoulder surfing issues [15].

Akpulat et al. [16] proposed a hybrid graphical password scheme known as T&C. In hybrid schemes multiple user authentication schemes are combined into single scheme. In this scheme passwords are consist of alphanumeric characters

and a location inside a picture. Users enter alphanumeric part of a password in text field through keyboard while location is identified through mouse. Usability is not a big issues in T&C scheme but passwords can be captured by online attacks, because they are directly inserted into the login screen. Another hybrid graphical password scheme was proposed by Alsaiari et al. [17], the scheme is known as Gotpass. GOTPass scheme is designed by combining properties of Android unlick, Deja Vu and textual password schemes. For authentication a user has to draw password lines and insert some codes which represent different password images. The scheme provide resistance from key logger, mouse logger and dictionary attacks but combined screen scrapper and key-mouse logger attacks can reveal passwords. This scheme has many usability issues such as high error rate and authentication time. This scheme also requires large amount of information to memorize.

## III. Proposed Authentication Scheme

In this research a user authentication scheme is proposed which reduces the security weaknesses of textual password scheme. The proposed scheme has two common authentication phases which are registration and login. Registration phase is same as ordinary textual password scheme but passwords are saved with different methodology. In the login phase changes are made in password entry screen and password verification process. Both phases are explained here.

### A. Registration Phase

In this phase authentication information of a new user is registered. In the proposed scheme registration information is taken in same way as in ordinary textual password scheme. Therefore registration process is required to be executed in a secure machine and environment, where no one should be able to monitor the process. A secure channel should be used during registration time such as SSL/TLS [18] [19] for collecting password from a user. Generally registration phase is consist of three layers, which are password collection, password encryption and password storage into the database. In order to improve the password security from offline guessing attacks, transformation layer is added into the registration phase. The transformation layer is described here.

### B. Password Transformation

In this layer alphanumeric characters of a password are converted into different alphanumeric character or symbols. Password transformation helps in resisting from brute force and dictionary attacks. For resisting brute force attack, theoretical password space and effective password space need to be high. Theoretical password space is the total number of passwords available in an authentication scheme, while effective password space is the total number of passwords being used by the users inside a scheme. Theoretical Password space and effective password space are increased by adding password transformation layer into the proposed scheme. Standard keyboard contains 94 alphanumeric characters excluding space key, therefore theoretical password space of textual password can be described with equation 1.

$$\sum_{i=1}^{94} 94^i \tag{1}$$

Majority of the users create password from less than 13 alphanumeric characters [20], therefore effective password space can be described with equation 2.

$$\sum_{i=1}^{12} 94^i \tag{2}$$

In order to decrypt a password, attackers need to check all the passwords belong to effective password space or in special case theoretical password space. Password transformation layer helps in increasing the size of theoretical and effective password space by adding symbols along with 94 alphanumeric characters.

Password transformation can be static or dynamic. In static transformation, same password of different users generate same transformed string. While in dynamic transformation, different transformed strings are generated from same password of different users. Password transformation can be carried out with many techniques, for example one strategy for static transformation is described using the following steps.

(i) Create a list of alphanumeric characters as shown in Table I. The table contains all 94 alphanumeric characters.
(ii) Create a combined alphanumeric characters and symbols list as shown in Table II. The list may be consist of more than two hundred elements.
(iii) Find out the index number in Table I, which belongs to first character of a password.
(iv) Get an element from Table II, which has same index number generated from previous step. The element would be transformed character or symbol.
(v) Fetch index number of next character of the password from Table I.
(vi) Sum previous index and current index of the elements, generated from Table I.
(vii) Fetch an element from Table II, which has the index number generated after summation in step vi.
(viii) Step v to vii will continue until all password characters are transformed.

TABLE I. List of alphanumeric characters

| index | character |
|---|---|
| 1 | a |
| 2 | b |
| 3 | c |
| 4 | d |
| 5 | e |
| 6 | f |
| 7 | g |
| ... | ... |
| 94 | 9 |

With the above transformation method the password "bdg" will be transformed to "$\beta$Y$\sigma$" through the following steps, if the alphanumeric characters are stored in the form of Table I and symbols are is stored in the form of Table II.

(i) System picks the index of first password character 'b' from Table I. The index of 'b' is '2'.
(ii) System gets an element from Table II which has index '2' . In this case the element is '$\beta$'.
(iii) System fetches index of second password character 'd' from Table I. The index of 'd' is '4'.

TABLE II. LIST OF SYMBOLS AND ALPHANUMERIC CHARACTERS

| index | character |
|---|---|
| 1 | $\alpha$ |
| 2 | $\beta$ |
| 3 | X |
| 4 | $\delta$ |
| 5 | b |
| 6 | Y |
| 7 | $\lambda$ |
| 8 | $\Theta$ |
| 9 | $\chi$ |
| 10 | g |
| 11 | $\sigma$ |
| ... | ... |

(iv) System generates new index '6' by adding current index '4' with previous index '2'.
(v) System fetches an element from Table II which has the index '6'. In this case the element is 'Y'.
(vi) System picks the index of last password character 'g' from Table I. Here the index of 'g' is '7'.
(vii) System generates index "11" by adding current index '7' with previous index '4'.
(viii) System fetches an element which has index "11" in Table II. In this case the element is '$\sigma$'.

Dynamic password transformation is also achieved by different methods, one of the method is password concatenation. In this method before applying password transformation steps, some characters are added into the password of a user. For example first three characters of user's email address can be concatenated with the password. Every user has different email address, therefore same password of two users will have different transformed string.

### C. Login Phase

Authentication process of the proposed scheme is different from ordinary textual password scheme. In the password field users need to enter decimal numbers which represent the alphanumeric character of their password. For authentication, decimal numbers entered by a user are mapped into alphanumeric characters and then the characters are matched against stored password. Login phase is further divided into three parts, which are login screen generation, password entry and password matching.

*1) Login Screen Generation:* Login screen is a medium through which authentication information is collected and sent to a server. Login screen of the proposed scheme contains all alphanumeric characters along with some numbers as shown in Figure 1.

The alphanumeric characters are represented by decimal numbers from 0 to 9 (total 10 numbers). Each decimal number is assigned to 9 or 10 alphanumeric characters, because 94 alphanumeric characters are shown in the login screen and they are represented by 10 decimal numbers.

Each time a user opens the login page, the decimal numbers are randomly assigned to the alphanumeric characters. For example characters (g m x F G P X ) > ) are represented by decimal number '4' in the login screen as shown in Figure 1. While in another session the alphanumeric characters (f h r y O X = [ ]) are assigned to the same decimal number '4' as shown in Figure 2.

Through algorithm 1, every alphanumeric character is assigned a random decimal number within the range of 0 to 9. All alphanumeric characters and their corresponding decimal numbers are saved into session variable for password matching.

**Algorithm 1** Numbers to characters mapping

```
alphaNum = List of alphanumeric characters
counter = 0
comment: Each decimal number is stored 10 times
for i = 0 to 9 do
    for j = 0 to 9 do
        numbers[counter]=i
        numbers++
    end for
end for
CLength = 100
comment: Rearranges elements of numbers array
for i = 0 to 100 do
    tempElement = Null
comment: A random index is generated
    ind = Random(0,cLength)
    tempElement = numbers[ind]
    numbers[ind] = numbers[cLength]
    numbers[cLength] = tempElement
    cLength = cLength -1
end for
for i = 0 to 94 do
    alphaNum[i][0] = numbers[i]
end for
```

*2) Password Entry:* Passwords of the proposed scheme consists of alphanumeric characters but in the password field some decimal numbers are entered, which represent the password characters. For example if a user's password is "bdg", then the user has to enter "724" in the password field, if the login screen is same as shown In Figure 1. Here first digit '7' represents the password character 'b' next digit '2' represents the character 'd' and last digit '4' represents the password character 'g'.

*3) Password Matching:* In the password matching phase, authentication information provided by a user is compared against the stored authentication information. Password matching is further divided into two steps, password re-transformation and password numbers matching. Whole process of password matching is given in the flowchart as shown in Figure 3.

*a) Password re-transformation:* In this step a password is restored into its original form based upon the username provided by the user. A password is first decrypted if encrypted value is stored into a database and then the password is re-transformed by the following steps.

(i) Index of first symbol or element is fetched from Table II.
(ii) A password character is fetched from Table I, which contains the index number generated from step i.
(iii) System fetches index of next element from Table II.
(iv) An index number is generated by subtracting the index of Table I (belongs to previous password character) from the index generated in step iii.

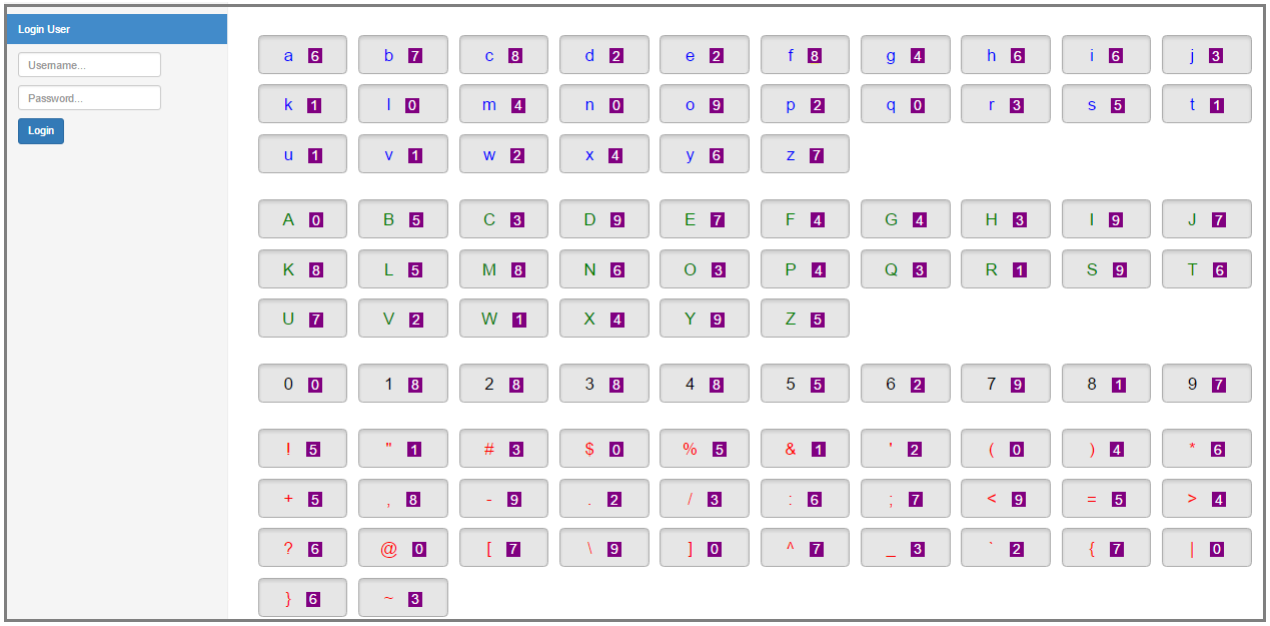


Fig. 1. Login screen

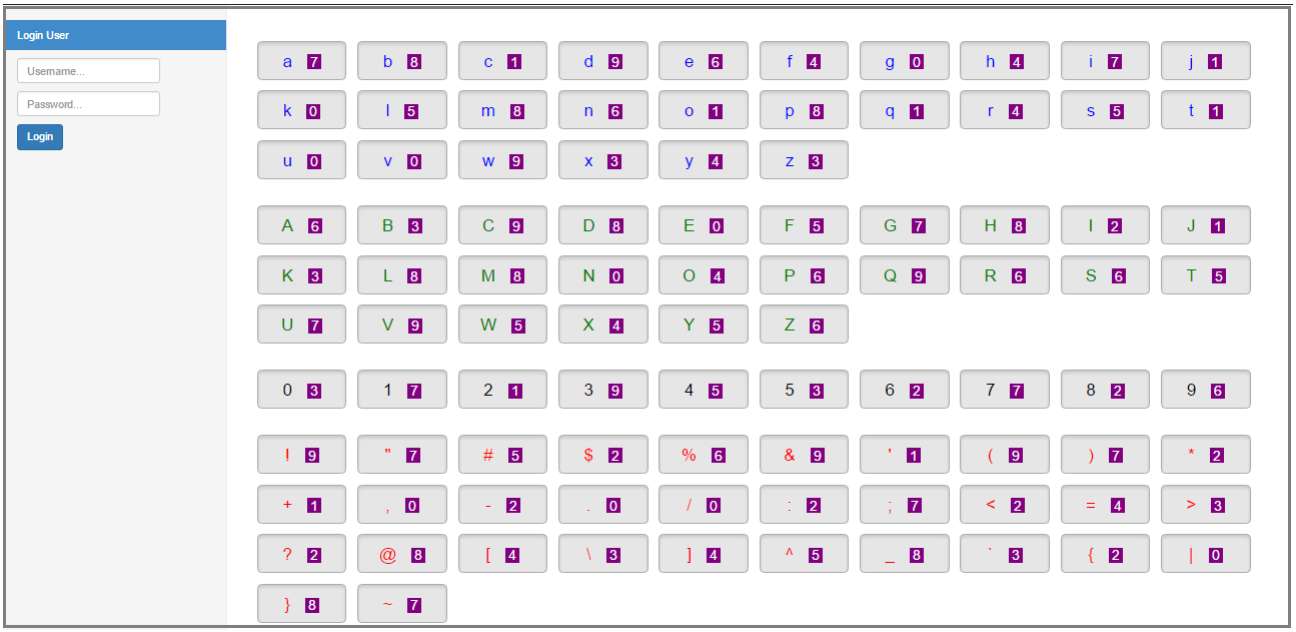


Fig. 2. Login screen 2

(v) A password character is fetched from Table I which belongs to the index number generated in step iv.
(vi) Steps iii to v will continue until all the password elements are fetched.

In the current scenario saved symbols "$\beta$Y$\sigma$", are converted to "bdg" through the following steps.

(i) System fetches the index of first symbol '$\beta$' from Table II, in this case the index is '2'.
(ii) System fetches a character from Table I, which has same index '2' . In this case the system fetches character 'b', which is the first element of the password.
(iii) System then fetches the index of next symbol which is 'Y' from Table II. The index of 'Y' is '6' in the table.
(iv) System subtracts the index of 'b' which is '2' from the index of 'Y' which is '6'. The new index becomes '4'.
(v) System fetches a character from Table I which has index position '4'. In this case the element is 'd' which is the second character of the password.
(vi) System then fetches the index of '$\sigma$' from Table II, the index is "11".
(vii) An index number is generated by subtracting the index of 'd' from Table I, with the symbol '$\sigma$' from Table II. In this case the index of 'd' is '4' and index of '$\sigma$' is '11', therefore a new index '7' is generated.

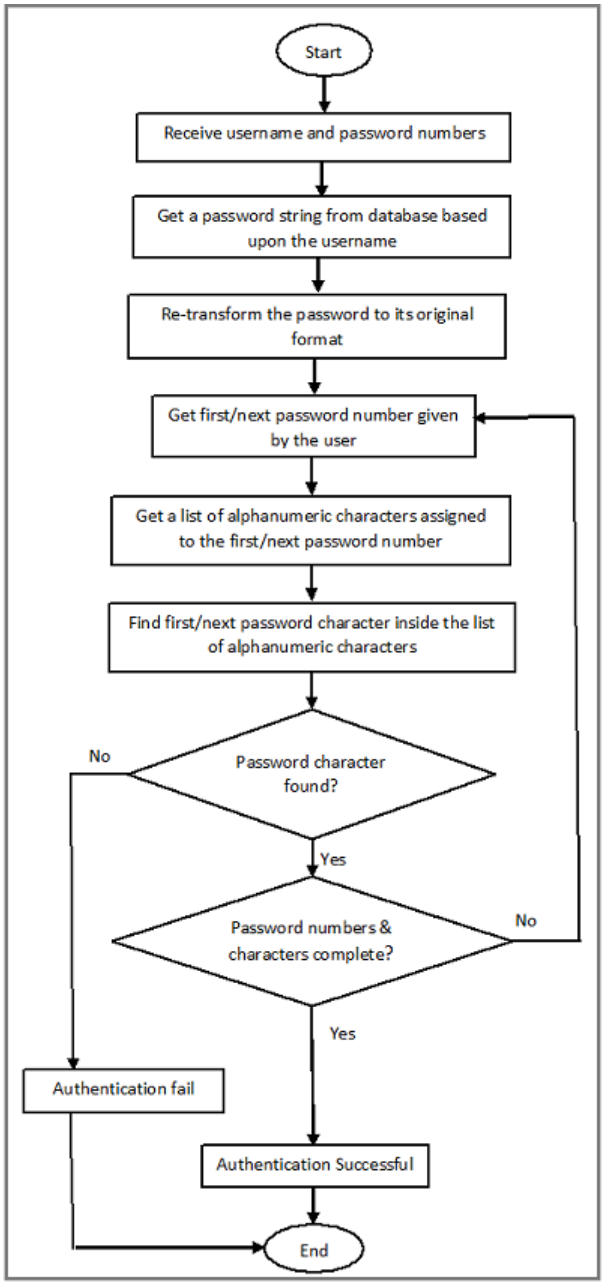


Fig. 3. Flowchart of password matching

(viii) System fetches a character which has index number '7' in the Table I. In this case the character is 'g', which is the last character of the password.

*b) Password numbers matching:* In this step password numbers given by the users and re-transformed password are matched for authentication. Password numbers are matched with the following steps.

(i) Server receive username and decimal numbers for authentication.
(ii) Based upon username, password is re-transformed as described in password re-transformation process.
(iii) System fetches a list of 9 or 10 alphanumeric characters from the session variable, which belong to first password number provided by the user.
(iv) System searches first character of re-transformed or original password within the list of 9 or 10 alphanumeric characters.
(v) If system successfully finds the password element in the list of alphanumeric characters, then system will repeat step iii & iv for all password numbers.
(vi) If all re-transformed password characters successfully matches with the corresponding list of password characters, then system will allow login, otherwise user will not be authenticated.

*4) Illustration of password matching:* If password of a user is "bdg" and the password numbers entered by the user are "724". Based upon the login screen shown in Figure 1 the system matches the password with the following steps.

(i) System will fetch first decimal number, which is '7'.
(ii) Corresponding alphanumeric characters are fetched from the session variable. In this case the characters belong to decimal number '7' are (b z E G U 9 ; [ ^ {) see login screen as shown in Figure 1.
(iii) System successfully finds first character of the password, which is 'b' within the set of ten elements (b z E G U 9 ; [ ^ {).
(iv) System will fetch next decimal number, which is '2' and its corresponding alphanumeric characters, which are (d e p w V ' 6 . ´).
(v) System successfully finds second password character 'd' within the set (d e p w V ' 6 . ´).
(vi) Finally the system will fetch last decimal number, which is '4' and its corresponding alphanumeric characters, which are (g m x F G P X ) >).
(vii) Password character 'g' is also present in the set (g m x F G P X ) >), therefore user will be authenticated.

## IV. Analysis of the Proposed Scheme

This section consists of two parts, in first proposed scheme is analyzed with respect to security. In second part, usability analysis of the scheme is given.

### A. Security Analysis

Proposed scheme improves the security of textual password scheme, by taking measures in password entry and database storage. Many security attacks are applied for password theft, in this section different security attacks are discussed with respect to the proposed scheme.

*1) Brute Force Attack:* In this attack all possible passwords exist in an authentication scheme are matched with the encrypted password value, in order to get the original password. Although proposed scheme uses same alphanumeric characters for password creation, but the theoretical and effective password space of the scheme is increased by using the transformation layer. Therefore brute force attack becomes difficult to apply.

*2) Dictionary Attack:* This is the efficient form of brute force attack, in which small number of passwords are tried to guess a password. In this attack a list of passwords are generated which is called password dictionary and the passwords are cracked by comparing every password of the dictionary

with a user's password. Password dictionary creation is very difficult in the proposed scheme, because the sequence of symbols generated after password transformation may not be same within different implementations of the scheme.

*3) Shoulder Surfing Attack:* In shoulder surfing attack, authentication information is revealed through camera recording or by directly observing login activity. This attack is not possible in the proposed scheme, because user does not directly enter the password characters but they enter some decimal numbers which represent multiple alphanumeric characters.

*4) Random Guessing Attack:* In this attack an attacker blindly try some passwords for authentication. The probability of successful random guessing attack in the proposed scheme is given by the equation 3

$$P(S) = (1/10)^N \tag{3}$$

Here n is the length of a password and the number 10 shows the range of decimal numbers used inside the scheme. If the length of a password is 8 alphanumeric characters then the probability of successful random guessing attack is 0.00000001. This probability is very low that is why attacker will not rely on this attack to break the password.

*5) keystroke/mouse logger attack:* keystroke/mouse logger programs send information to attackers without the consent of users. In textual password scheme keystroke logger can easily send the password of a user. While in the proposed scheme keystroke logger are not helpful for attacker, because they only get some random numbers instead of a password.

*6) Man-in-the-middle Attack:* There are many form of main-in-the-middle attack such as phishing and replay attacks. In phishing attack a legitimate user is redirected to a fake website, where the user enters the authentication information. From the fake website attacker gets the authentication information. In replay attack, password is recorded from a communication medium and then the recorded information is used later for authentication. Both attacks are not possible in the proposed scheme, because an attacker does not get exact password characters.

*7) Multiple Recording Attack:* Passwords in knowledge based authentication schemes can be captured by recording information of multiple login sessions. The information may be in the form of camera recordings or spyware data. Proposed scheme also suffer from this attack but it requires recordings of multiple login sessions. Equation 4 shows the condition in which password of the scheme is captured by multiple recording attack.

$$X \cap Y = Z / |Z| = 1 \tag{4}$$

Here X and Y are the set of alphanumeric characters of two login sessions belongs to a password character and Z is the intersection result. A password character is captured when cardinality of set Z comes to 1 or single alphanumeric character is generated after intersection. If cardinality of Z is not equals to 1 then recursively intersection work has to be performed until single element remains in Z. The recursion process is given by the equation 5.

$$Z_{i-1} \cap Y_i = Z_i \tag{5}$$

Here i is the count of login sessions, $Z_{i-1}$ shows the intersection result of previous two sets and $Y_i$ is the set of alphanumeric characters related with current login session.

In the current scenario the password "bdg" is captured by the following steps.

(i) Attacker gets the information of first login session, which consists of password numbers "724" given by the user and screen-shot of the login screen appeared in-front of the user as shown in Figure 1.
(ii) Attacker gets the information of second login session, which consists of password numbers "894" and screen-shot of the login screen as shown in Figure 2.
(iii) Attacker gets the alphanumeric characters of first login session, which are related with first password digit. The password digit is '7' and its corresponding characters are (b z E G U 9 ; [ ^ {), see Figure 1.
(iv) Attacker gets the alphanumeric characters of second login session, which are related with first password digit. The password digit is '8' and its corresponding characters are (b m p D H L M @ _), see Figure 2.
(v) After intersection of set (b z E G U 9 ; [ ^ {) and (b m p D H L M @ _), the attacker captures the character 'b' of the password.
(vi) Same process will be repeated for password characters 'd' and 'g' by utilizing second and third password digits given by the user.

It is not necessary that a password character is cracked after recording information of two login sessions. Multiple login sessions may be required for capturing a password character, because password numbers are randomly assigned to nine or ten password characters.

### B. Usability Analysis

Usability study was conducted in order to analyze password entry time and input accuracy of the proposed scheme. A web based application was developed in order to perform the usability tests. The testing application was created through PhP programming language and MySQL database. In the application users performed registration and login activities. Different processes were created inside the application for storing password entropy and password entry time inside the database. Information of successful and failure login attempts were also stored inside the database for analysing the input accuracy of the proposed scheme.

*1) Procedure:* In the experiment users were asked to register once and make three successful login attempts. Maximum three attempts were allowed for a successful login. Each user performed registration and login activities in single session because memorability evaluation was not the objective of the experiment. Users performed registration activity on a traditional registration page, because password input method in the registration screen of the proposed scheme is same as textual password scheme. While graphical interface of the login page was similar to Figure 1. Before starting the test, the purpose of testing and how to perform registration and login activities were explained to the users.

*2) Participant:* Application was tested by 30 volunteer users belong to Quaid-e-Awam University. From volunteers, 8 users were faculty members while remaining 22 users were students of different departments. Both male and female users participated in the experiment.

*3) Experiment Results:* Experiment data was collected from the database of the application in order to analyse the authentication time and input accuracy. Total 30 users made 123 login attempts, which were both fail and successful login attempts. Average password length of the thirty users was 7.83 alphanumeric characters and average password entropy was 38.28 bits.

*a) Password entry time:* Table III shows the password entry timings of the proposed scheme. Mean authentication time of the proposed scheme is 16.73 seconds which is larger than textual password scheme because users require extra time for searching the alphanumeric password characters in the login screen.

TABLE III. PASSWORD ENTRY TIMINGS OF THE PROPOSED SCHEME IN SECONDS

| Mean | Lowest time | Largest Time | Standard Deviation |
|---|---|---|---|
| 16.73 | 9 | 34.79 | 7.54 |

*b) Input accuracy:* Input accuracy of the proposed scheme with respect to first login attempt and within three login attempts is given in the table IV.

TABLE IV. INPUT ACCURACY

| Accuracy | Percentage |
|---|---|
| First Attempts | 73.17 % |
| Within three attempts | 100% |

Results show that the number of input errors are high in first login attempt, which may be due to a new method of password insertion. The users made less number of input errors when they become familiar with the graphical interface of the login screen.

## V. SECURITY COMPARISON OF THE PROPOSED SCHEME

Many authentication schemes are proposed which have different advantages and disadvantages. In this paper proposed scheme is compared with three commercially used authentication schemes along with traditional textual password scheme. Commercially used schemes are Android unlock pattern scheme, Passface scheme and Picture Gesture Authentication (PGA) scheme used by Microsoft in windows 8 operating system. Security comparison with respect to different attacks is given in table V. For comparison three values are used depending upon the level of resistance provide by the schemes against a particular security attack. The values are "Strong", "Moderate" and "Weak". The value "Strong" shows that the scheme provides high level resistance to the particular attack while the value "Moderate" shows that mid level resistance and the values "Weak" shows the scheme is weak or not resilient to the particular attack.

Table V shows that proposed scheme improves the security of traditional textual passwords against different type of attacks, only multiple recording attacks is a threat with the scheme. Proposed scheme is resilient to brute force and dictionary attacks due to password conversion process presented in the paper. Proposed scheme also provides better security in comparison to different commercial authentication schemes.

## VI. CONCLUSION

The idea of proposed scheme is not to replace textual password scheme, but to enhance the scheme for improving the security aspect. Users can easily shift towards the proposed scheme from textual password scheme, because old textual passwords can be used inside the scheme. The proposed scheme is also easy to learn because a very simple approach is used for login process.

Proposed scheme uses alphanumeric character based passwords, therefore memorability results would be same as textual password scheme. The problem with alphanumeric passwords is that easy to remember passwords are easy to guess through dictionary attack [21]. However in the proposed scheme easy to remember passwords are not easy to guess due to password transformation layer. Proposed password transformation layer can also be used for other knowledge based authentication schemes for improving the security against dictionary and brute force attacks.

Many security threats can be resisted when passwords are indirectly inserted into an authentication scheme, but this approach has usability cost [1]. Users may require more mental or physical work to do in order to indirectly insert the passwords. That is why login time of the proposed scheme is higher than traditional textual password scheme, but this usability disadvantage is less than security advantages provides by the proposed scheme.

## REFERENCES

[1] Q. Yan, J. Han, Y. Li, H. DENG *et al.*, "On limitations of designing usable leakage-resilient password systems: Attacks, principles and usability," 2012.

[2] A. Adams and M. A. Sasse, "Users are not the enemy," *Communications of the ACM*, vol. 42, no. 12, pp. 40–46, 1999.

[3] D. Davis, F. Monrose, and M. K. Reiter, "On user choice in graphical password schemes." in *USENIX Security Symposium*, vol. 13, 2004, pp. 11–11.

[4] I. Jermyn, A. J. Mayer, F. Monrose, M. K. Reiter, A. D. Rubin *et al.*, "The design and analysis of graphical passwords." in *Usenix Security*, 1999, pp. 1–14.

[5] G. E. Blonder, "Graphical password," Sep. 24 1996, uS Patent 5,559,961.

[6] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz, "Quantifying the security of graphical passwords: the case of android unlock patterns," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 161–172.

[7] H. Zhao and X. Li, "S3pas: A scalable shoulder-surfing resistant textual-graphical password authentication scheme," in *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, vol. 2. IEEE, 2007, pp. 467–472.

[8] Z. Zheng, X. Liu, L. Yin, and Z. Liu, "A stroke-based textual password authentication scheme," in *Education Technology and Computer Science, 2009. ETCS'09. First International Workshop on*, vol. 3. IEEE, 2009, pp. 90–95.

[9] Y.-L. Chen, W.-C. Ku, Y.-C. Yeh, and D.-M. Liao, "A simple text-based shoulder surfing resistant graphical password scheme," in *Next-Generation Electronics (ISNE), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 161–164.

[10] K. Rao and S. Yalamanchili, "Novel shoulder-surfing resistant authentication schemes using text-graphical passwords," *International Journal of Information and Network Security*, vol. 1, no. 3, p. 163, 2012.

TABLE V. SECURITY COMPARISON

| Scheme | Brute Force attack | Dictionary Attack | Shoulder Surfing | Random guessing attack | Keystroke/Mouse logger Attack | Man-in-the-Middle Attack | Multiple recording attack |
|---|---|---|---|---|---|---|---|
| Android unlock | Weak | Weak | Weak | Moderate | Moderate | Weak | Weak |
| PassFaces | Weak | Weak | Weak | Moderate | Weak | Strong | Weak |
| PGA | Strong | Strong | Weak | Moderate | Moderate | Moderate | Weak |
| Textual passwords scheme | Moderate | Moderate | Moderate | Strong | Weak | Weak | Weak |
| Proposed scheme | Strong | Strong | Strong | Strong | Strong | Strong | Moderate |

[11] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "Passpoints: Design and longitudinal evaluation of a graphical password system," *International journal of human-computer studies*, vol. 63, no. 1, pp. 102–127, 2005.

[12] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, "Design and evaluation of a shoulder-surfing resistant graphical password scheme," in *Proceedings of the working conference on Advanced visual interfaces*. ACM, 2006, pp. 177–184.

[13] N. Lopez, M. Rodriguez, C. Fellegi, D. Long, and T. Schwarz, "Even or odd: A simple graphical authentication system," *IEEE Latin America Transactions*, vol. 13, no. 3, pp. 804–809, 2015.

[14] D. Weinshall, "Cognitive authentication schemes safe against spyware," in *Security and Privacy, 2006 IEEE Symposium on*. IEEE, 2006, pp. 6–pp.

[15] H. Gao, W. Jia, N. Liu, and K. Li, "The hot-spots problem in windows 8 graphical password scheme," in *Cyberspace Safety and Security*. Springer, 2013, pp. 349–362.

[16] M. Akpulat, K. Bicakci, and U. Cil, "Revisiting graphical passwords for augmenting, not replacing, text passwords," in *Proceedings of the 29th Annual Computer Security Applications Conference*. ACM, 2013, pp. 119–128.

[17] H. Alsaiari, M. Papadaki, P. Dowland, and S. Furnell, "Graphical one-time password (gotpass): a usability evaluation," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 94–108, 2016.

[18] T. Dierks, "The transport layer security (tls) protocol version 1.2," 2008.

[19] A. Freier, P. Karlton, and P. Kocher, "The secure sockets layer (ssl) protocol version 3.0," 2011.

[20] Z. Liu, Y. Hong, and D. Pi, "A large-scale study of web password habits of chinese network users." *JSW*, vol. 9, no. 2, pp. 293–297, 2014.

[21] X. Suo, Y. Zhu, and G. S. Owen, "Graphical passwords: A survey," in *Computer security applications conference, 21st annual*. IEEE, 2005, pp. 10–pp.