Journal of Theoretical and Applied Information Technology
15th November 2024. Vol.102. No. 21
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

JUMBLEDKEYS: TWO FACTOR USER AUTHENTICATION SCHEME USING PARTITIONED VIRTUAL KEYBOARD

S. RAJARAJAN[1], PLK. PRIYADARSINI[2*]

[11]Assistant Professor, SASTRA Deemed University, Tamilnadu, India
[2] Senior Assistant Professor, SASTRA Deemed University, Tamilnadu, India
E-mail: [1]srajarajan@cse.sastra.edu, [2]priya.ayyagari@it.sastra.edu

ABSTRACT

Passwords are the oldest and most widely used method of user authentication. While password based authentication is cost-effective, easier, faster and flexible, it is also susceptible to attacks. Particularly at the client device, passwords are most vulnerable when users enter their passwords. Before the password is encrypted and forwarded to the server, it can be captured by the attackers by exploiting the loopholes at the client systems that users use. In this paper, we introduce a secure virtual keyboard scheme designed to shield passwords from attackers during entry. The keyboard is partitioned into four groups and the keys in each group are randomized after each password character entry. Instead of directly clicking on the actual password characters, users will be clicking on the designated target keys as per a key-transfer scheme. The key-transfer scheme will be communicated to users through a SMS to their registered mobile number every time they attempt to login, effectively making the mobile phone a second factor of authentication. The jumbledKeys keyboard then generates a dynamic password based on the keys clicked by users. Only this dynamic password is stored at the client's form and sent to the server. Attackers cannot trace the actual password without the knowledge of the key transfer scheme and the position of keys on the keyboard. This protects passwords against the shoulder-surfing, form grabbing, public wifi and man-in-the-middle attacks. Our user survey results proved that our scheme has not compromised on usability while elevating security

Keywords: *User authentication, Password attacks, Internet banking, Shoulder-surfing, Form grabbing, Keylogging, Virtual keyboard, Cyber security*

1. INTRODUCTION

User authentication is an essential component of any computer application that serves customers. Accurately recognizing the users helps in customizing the services, restricting the access rights, protecting the resources and preventing illegal operations [1]. Generally, some secret information that is known only to the user or possessed by the user is verified to determine the authenticity of the users [2]. The secret information is either remembered by users or carried by users or integral part of the user. There are three categories of authentication schemes, based on what kind of secret information is used for authentication [3]. They are broadly categorized as Knowledge-based , Token-based and Biometrics based authentication schemes.

Knowledge-based schemes are more popular and are widely used among the three categories of schemes, Passwords and PIN numbers are the two of the commonly used knowledge-based authentication schemes. At the same time, they are frequently attacked by adversaries to steak the passwords and PIN numbers since they are usually made out of the limited character set comprising alphabets, digits and symbols. The three categories of authentications are summarized in Figure 1.

There are two important challenges to password-based authentication. First, attackers have several ways to predict the passwords without having the knowledge of the actual passwords. Users have the tendency to use common words, personal details, popular terms, and important dates in their passwords in order to remember them easily. This leads to dictionary attacks [4]. It was discovered in a survey that the word "password" has been included in the large percentage of passwords chosen by people [5].

Second challenge in password based authentication is remembering several passwords for different accounts. On an average, a person has to remember passwords for ten different accounts.

Ideally each account must be assigned a unique password. Otherwise, successful attack of one account will lead to a cascading effect of breaching multiple accounts of the same user. Unfortunately, a large percentage of users reuse their passwords for multiple accounts to avoid the burden of remembering too many passwords [6][4]. To mitigate dictionary attacks, users are forced to choose complex passwords consisting of the combination of upper case, lower case, numerical digit and symbols. Periodic resetting of passwords is also imposed in certain more critical accounts like Internet banking accounts. This puts a huge burden on the users to remember all the passwords and also remember for which accounts what passwords are to be entered. This leads to frequent forgetting of passwords and password reset requests. So there is a need for stronger as well as easily memorable passwords [8].

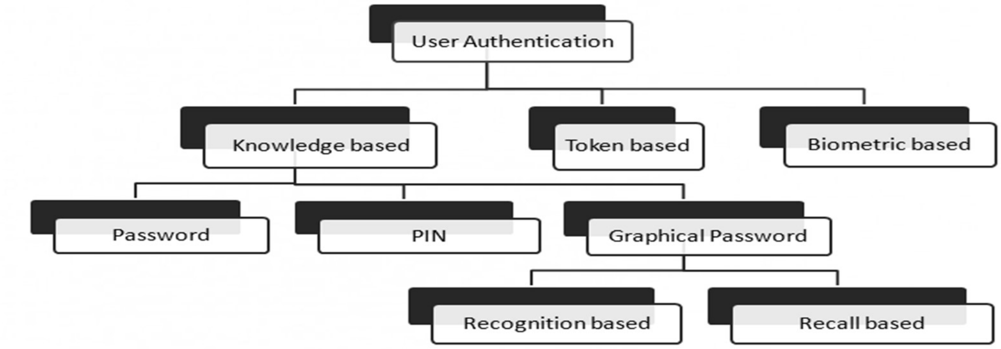


Figure 1.Category of user authentication schemes

There are well known attacks that are carried out against the user authentication to capture the credentials of the users. Password attacks are of two categories [4]:

- Attacks that attempt to steal the passwords
- Attacks that try to predict the passwords

The first category of attacks is more common, since predicting the passwords is very hard as the entropy of passwords is high entropy due to the stringent rules imposed by the agencies on password selection. So directly stealing the password from the user makes the job of the attacker much easier and quicker. Preventing those attacks is also relatively difficult. Based on the location of the attacks, password attacks can be classified into three types:

- Client-side attacks
- Communication attacks
- Server-side attacks

In client-side attacks, attackers target the users' devices to steal their credentials [7]. The device could be a computer or a mobile phone. Attackers either employ special hardware to execute the attack or install malware into the device to secretly monitor and grab the user details. In communication attacks, an attacker intercepts the traffic between client and server and acquires the credentials being transmitted. In a server attack, an attacker hacks the server and steals the database consisting of users account details. Several incidents of attacks on bank servers leading to customer details getting leaked have happened [9][10].

Attacks at the communication channel and server side are being greatly mitigated with the help of strong encryption algorithms and secured transmission protocols. But the client systems continue to remain vulnerable. Potential attacks at client systems commences when the credentials are entered using keyboard and till the credentials are submitted. Some of the client-side attacks are Shoulder surfing [11] [12] [13], Key-logging [14] [15] [16] [17], Form grabbing [18] [19] [20], Evil twin [21] [22] [23], Web-skimming [24] [25], Cross-frame Scripting (XPS) [26] [27] [28]. A common characteristic of these attacks is that they all capture the user's credentials before they are submitted to the application layer protocol. Some of the common attacks are summarized in Table 1.

Virtual keyboards are software components that can be used as an alternative for hardware keyboards and they can be operated with the help of a mouse. Though they were introduced as an option for the people with disabilities who cannot operate keyboards, they are also used to provide security against keylogging attacks [15][19][29]. But virtual keyboards heighten the chances of shoulder surfing attacks since the keyboard entry is clearly visible on the screen.

Designing covert keyboard entry schemes to prevent shoulder surfing has been the research objective of several researchers. Virtual keyboards are the default option in mobile phones. Mobile keyboard apps can potentially steal confidential information typed from its users [30]. So developing security mechanisms for secured entry of credentials for user accounts is important. Table 1.1 summarizes the various password attacks that are carried out at the locations of password entry, transmission and storage.

Table 1. Summary of attacks

Name of the attack	Attack location	Description
Phishing attack	Client system	Attacker sends a malicious link through email by masquerading it as a legitimate site
Man-in-the-Middle attack	Transmission channel	Attacker captures the credentials shared between client and server during login and uses them latter to masquerade user
Brute force attack	Server	Attacker generates all possible combinations of passwords and tries them against the user accounts
Dictionary attack	Server	Attacker tries the common passwords that are used by large number of people
Credential stuffing	Server	Attackers gather leaked or stolen credentials to gain access into user accounts
Keylogger	Client	Keyloggers are malwares secretly installed at the client systems to collect the keyboard entries and forward to the attacker
Shoulder surfing attack	Client	Users keyboard entries are observed or recorded to learn the passwords
Password spraying	Server	A large collection of commonly used passwords are attempted on a small number of user accounts
SQL Injection	Server	Attackers exploit the vulnerability in applications to insert malicious SQL queries
Form grabbing attack	Client	It is a malware based attack which obtains the login credentials from HTML forms where user types
Cross frame scripting	Client	After tricking an user to click a link, attacker places an Iframe that contains a legitimate site
Evil twin	Client	Users are tricked to connect to a fake Wi-Fi that mimics a legitimate Wi-Fi..
Web skimming	Client	It is an attack in which the attacker inserts malicious code inside the legitimate e-commerce web sites

2. RELATED WORKS

Passwords are susceptible to attacks. The easiest attacks on passwords are the shoulder surfing and keylogging attacks. With these attacks the attacker is able to learn the password before it gets encrypted and forwarded to the server. Protecting passwords against SSAs is an active research domain with plenty of new schemes areproposed. A major issue with these schemes is their usability. A blank keypad is displayed on the screen in [15]. Then the users have to capture an encrypted

QRCode shown along with the keypad, decrypt it using his private key that is stored in the smart phone to uncover the keypad layout and the user types on the blank keypad referring to the keypad layout. To conceal the actual password characters, random camouflage characters are typed along with the actual characters [31]. To differentiate the actual characters and camouflage characters master keys are used to enable and disable entry of camouflage characters. These keys are chosen by users as any combinations of alphabets, numbers or special characters but they should not be part of the actual password. The server will discard the camouflage characters and extract the remaining characters as the password.

A password creation policy based on drawing-to-text is proposed in [32]. Under this policy users have to choose and memorize keypad line drawings that form different shapes.

The keys that are part of the line drawings are the password keypads. A visual keypad scheme that comprises emoticons is proposed in [29]. Each key on the keypad is assigned a unique emoticon. During registration the user must choose a key as his starting keypad key.This key is encrypted and stored in the smart phone's memory. Using the navigation keys left, right, up and down users may move the emoticons across the keypad. In order to enter the password character, users must identify the emoticon at the position of the first starting keypad key and move it towards his password character with the help of the navigation keys. When the emoticon is placed at the password character user presses Enter key. Now all the emoticons are randomized on the screen and users must repeat this step for the remaining password characters.

Diksha Shukla *et. al.* [33] have demonstrated how passwords can be stolen by recording the hand movements of users when they are typing their passwords. The use of a thermal camera for capturing the change in keypad temperature after it was used for typing the password and thereby detecting the password was demonstrated by Duo Li *et. al* [17]. Shukun Yang *et. al* [6] have developed a scheme called DPPG which generates a dynamic password policy for users to choose their passwords so that passwords chosen are stronger against attacks. There are some schemes that try to generate a dynamic password each time. The dynamic passwords differ for each login. In the scheme proposed by Xiao, Yang, *et al* [34], users can generate a virtual password dynamically by applying some function over their actual password. This scheme involves some amount of human computation. A dynamic password generation protocol that combines a static password with dynamically changing password is proposed in [35]. This scheme is meant for preventing attacks carried out on the Internet.

3. PROBLEM STATEMENT

With the increasing usage of online accounts by people, both for less critical accounts like social media accounts to highly critical accounts like banking accounts, the importance of protecting passwords against several types of password attacks has become very important. Though, several contributions are made by researchers by designing secure password entry schemes, many of many of them still fall short in terms of usability. Some schemes provide limited security while focusing on better usability. Therefore, our research challenge is to design a password entry system that combines strong security with high usability. We wish to specifically fulfill the following requirements in our proposed scheme:

- The proposed scheme should secure password against multiple attacks so that it will be a comprehensive solution for password security.
- The proposed scheme must be a two factor authentication scheme since they offer considerably better security over single factor authentication schemes
- The proposed scheme should not burden the user for entering passwords by requiring them to have special skills, to learn new techniques, to buy additional tools, to spend more time and to repeat password entry multiple times by making mistakes.

4. PROPOSED SCHEME

4.1 Overview

The proposed scheme consists of a new keyboard interface for password entry. It is called the JumbledKeys keyboard since the keys on the keyboard are randomly positioned unlike the fixed layout of normal keyboards. Users can enter their password with the help of a key-transfer scheme that is communicated to them through a SMS to the registered mobile number. In the following sections, the design of the keyboard and the methodology to enter passwords through the keyboard is explained.

4.2 Design of Virtual Keyboard

The keyboard of the proposed approach consists of 68 keys with 26 alphabets (a-z), 10 digits (0-9) and 32 special characters ($, #, ? etc.). The keys in the JumbledKeys virtual keyboard are logically divided into four groups- G1, G2, G3 and

G4. Each group consists of 16 keys. G1 contains the alphabets from 1 to p. G2 contains the alphabets from q to z, numbers from 0 to 5 and a key for CAPS lock. G3 contains the digits from 6 to 9 along with 12 symbols. G4 contains 16 symbols. When the keyboard is shown to the user, the keys are randomized such that the keys are not in any fixed positions. But to reduce the burden of users in locating the keys on the keyboard, the randomization is confined at the group level and not at the entire keyboard level. To easily distinguish the four groups of keys, different colors are used for each group. On top of every key, there is a label containing an index number starting from 1 to 16 called key-id is shown. These key-ids will not be randomized and will be in the same positions permanently. The key-ids play an important role in the password entry process. The keyboard model is provided in Figure 2.

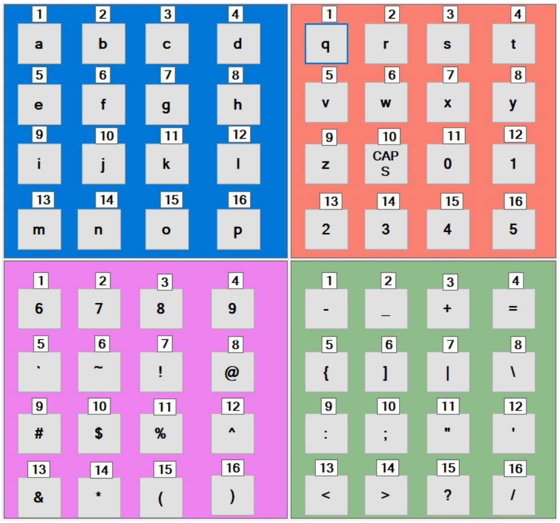


Figure 2. Design of keyboard

4.3 Methodology of the Scheme

The proposed JumbledKeys virtual keyboard is split into four logical groups. Each group consists of 16 keys. To easily differentiate the groups, different colors are applied to each group. Each key is displayed with a key-id number which are sequential numbers starting from 1. A typical password will have its characters scattered in more than one group or even among all the groups. The scheme makes use of five key-transfer schemes which cause the keys in a group to be transferred to another group based on the pattern of the key-transfer scheme. The key-transfer schemes are explained in section 3.5.During the login process, users will be communicated the key-transfer scheme they need to use by a SMS message. While entering password, before each password character entry the keys will be transferred as per the key-transfer scheme. But these key transfers are not displayed on the keyboard, but they are made only internally inside the keyboard application on the arrays maintained for the four groups. So on the screen no key transfer takes place. After pressing the key-transfer button, users must predict the group to which the key group containing their password character would have been transferred as per the key-transfer scheme. They need to locate the key with the key id of their password character, in the target group and click that key. Once user clicks on a key, the keys on all the four groups are randomized within the groups before user enter the next password character. Finally, the characters of the actual keys pressed by the user are combined into a dynamic password *dwd* and sent to the server. Server can regenerate the actual password *pwd* out of the *dwd* with the help of the key-transfer scheme and the seed value used for the key randomization by the JumbledKeys application.

4.4 Password Entry Procedure

Registration

Step 1: Users create a new login by choosing a user id and password

User login

Step 1: Users enter their user id and date of birth and press KTS button

Step 2: A key-transfer scheme name is received in user's mobile phone

Step 3: Users recall their password

Step 4: Locate the first password character P_1 in G1 or G2 or G3 or G4 as per the KTS and note down its key-id displayed on top of the key.

Step 5: Press PROCEED button to virtually transfer keys among the groups

Step 6: Users identify the destination group as per the key-transfer scheme and the key-id of the target key where P_1 would have been transferred

Step 7: Press the target key on the destination group

Step 8: Users follow step 3 to 6 to enter remaining password characters

Step 9: Users press Submit button to submit the dynamic password *dwd* generated

4.5 Randomize Keys at the Client System

In the JumbledKeys scheme, client system performs the key transfer among the four groups as per the key-transfer scheme. In addition to that, it also randomizes the keys after each password character entry. This is done to increase the security against observation attacks. It follows a pseudo random technique along with a seed value provided by the server to randomize the keys. The algorithm for randomizing the keys is given below.

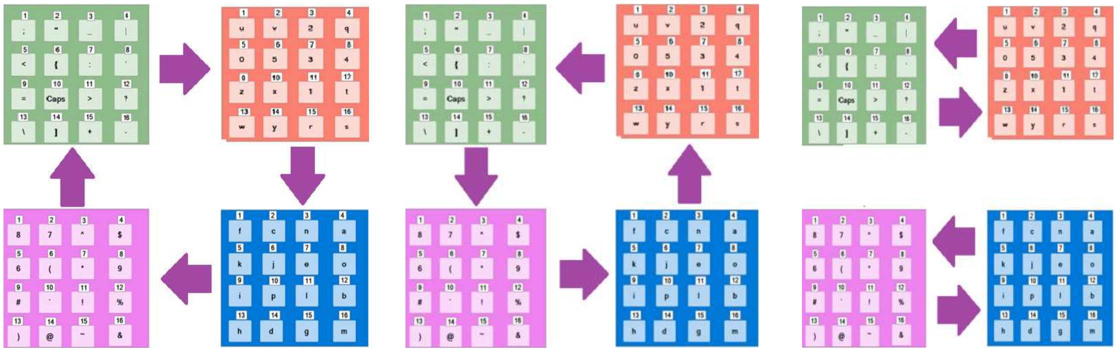
Algorithm randomizeKeys(char keys[], int n)

```
Input: a -seed number received from server
       Keys-Array of keys
       n-  Number of keys in a group
Output:Keys -Randomized keys array
       srand(a)
       for i = n – 1 to 1
               j = rand() Mod (i+1)
       swap(keys[i], keys[j])
```

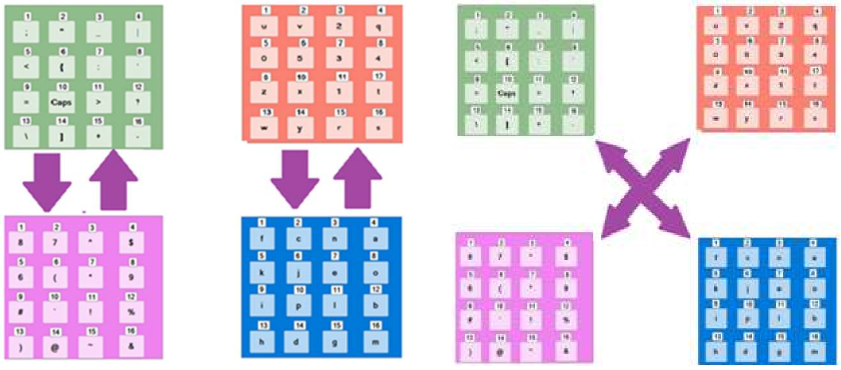
4.6 Key-transfer Schemes

In our scheme, users need to remember a key-transfer scheme in addition to their user id and password. Every user must choose a key swap scheme at the time of registration. There are five possible key-transfer schemes for the users to choose from. Once a particular scheme has been chosen by the user, this has to be remembered by the users and it is also recorded by the server. It remains a secret between the user and the server. The idea behind the key-transfer scheme is to swap or exchange the keys among the four groups before users enter their password characters. The key swapping is completed before the groups are displayed on the screen. JumbledKeys software maintains four arrays of the key groups and carries out the swapping of the arrays. Different options for the methods of key transfer are given below.

- **Clockwise Circular transfer:** In this option, the 18 keys in each key group are transferred to their next key groups in a clockwise manner. But the column positions of the keys in the transferred groups remain unchanged. So the user only needs to predict the new group number of the target key after the transfer, in order to select it. This is shown in Figure 3(a).
- **Anti-Clockwise Circular transfer:** In this option, the key transfers just happen in the anticlockwise order. Figure 3(b) indicates this transfer method.
- **Vertical transfer:** In this scheme, the two groups in the same row are interchanged with each other. Figure 3(c) pertains to this way of keys transfer.
- **Horizontal transfer:** In this scheme, the two groups in the same column are interchanged with each other. Figure 3(d) pertains to this way of keys transfer.
- **Cross-X transfer:** Here, the key transfers are happening in a cross wise manner. So the 1st and the 4th group keys are swapped with each other. Similarly the keys in the 2nd and 3rd groups are swapped with each other. This is presented in Figure 3(e).



(a) (b) (c)



(d) (e)

Figure 3. Key Transfer Schemes (a) Clockwise transfer (b) Anti-clockwise transfer (c) Horizontal transfer (d) Vertical transfer (e) Cross X transfer

4.7 Password Verification

The algorithm for the conversion of dynamic password into the actual password is done using an algorithm. It is discussed in this section. Once the *dwd* is received at the server, its characters are extracted one by one and mapped to their respective characters as per pwd. First step is identifying the index d of the character on the JumbledKeys keyboard that has 64 keys. With the help of d, the group G_i to which the key belongs to, is found. Next task is predicting the source key for which user would have clicked this target key. That requires knowledge of the key-swap scheme T. This scheme T registered by the user is retrieved from the server's database and used in the algorithm. Each character of *dwd* is converted into characters of *pwd* and matched against the stored *pwd.*

Algorithm Password Verify

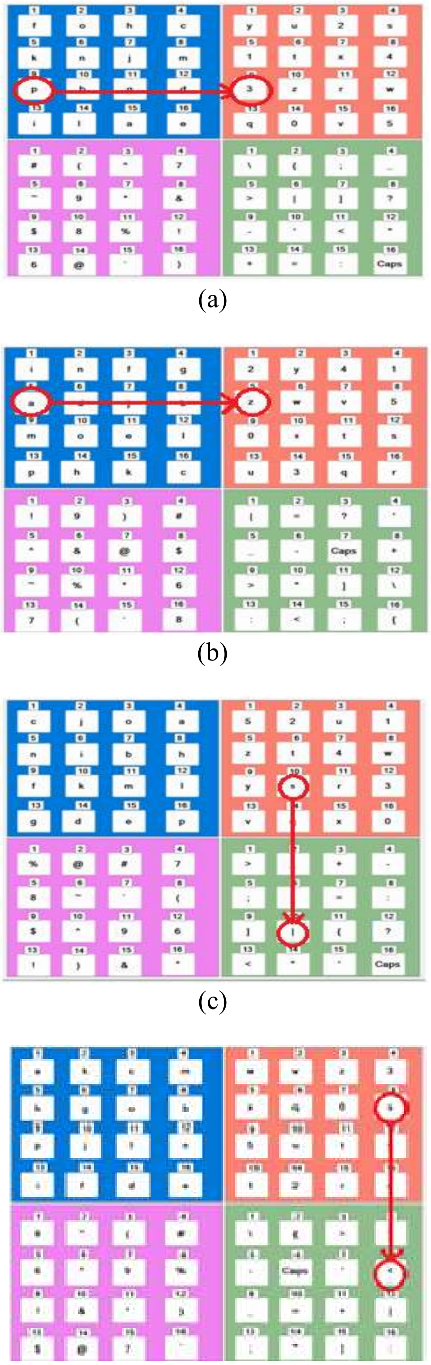
```
Input: dwd – dynamic password
       T – Key transfer scheme of user
Output: pwd – actual password of user
for i=0 to dwd.length
        d= findIndex(dwd[i])
        g = findGroup(d)
        k=findActualKey(dwd[i],g,d, T)
        pwd[i]=k
return pwd

findIndex(k, G)
Input: k – Dynamic password character
       G – Array of 64 keys of the keyboard
Output: i- Index of k in array G
i=0
while G[i] ≠ k
    i=i+1
return i

findGroup(i)
Input: i-Dynamic password character
Output: g-Group to which
if i<17 then g=0
else if i< 33 then g=1
else if i<49 then g=2
else g=3
return g
findActualKey(i,g,d,T)
switch(T)
    case 1: if g = 3 then k = G[d-48]
        else k= G[d+16]
    case 2: if g= 0 then k=G[d+48]
        else k=G[d-16]
    case 3: if g= 0 Or g=1then k=G[d+32]
        else if g= 2 Or g= 3 then k=G[d-32]
    case 4: if g= 0 or g=2 then k= G[d+16]
        else if g=1 or g=3 then k= G[d-16]
    case 5: if g= 0 then k=G[d+48]
        else if g=1 then k=G[d-48]
        else if g=2 then k=G[d+16]
        else if g=3 then k=G[d-16]
return
```

Table 2. Dynamic Password

Actual password pwd	Pass
Dynamic password	3z|<



(a)

(b)

(c)

(d)

Figure 4 : Sample Password Entry (a) Entry of 'p' (b) Entry of 'a' (c) Entering 's' (d) Entry of 's'

4.8 Sample Password Entry

In this section, we are presenting the demonstration of the proposed scheme using the screen shots of the model we implemented. Assume that the password to be entered is "pass" and the key-swap scheme is Clockwise KT. Figure 4(a) contains the initial keyboard layout. The first password character 'p' is present in group 1 as per its alphabetical order. As per the randomization happened on the keyboard, p is at placed at the index 9 and so 9 is the Key-id. As per Clockwise KT, keys of G_1 will get transferred to G_2. So user should click the key having the same key-id of p in $G_{2.}$ Number 3 is at the key-id 9 in G2. Immediately the keyboard is randomized again and it is as per Figure 4(b). Now the user locates 'a' in G1 at index 5. So the user clicks z in G2. Figure 4(c) and Figure 4(d) show the keyboard instances during the entry of 's' and 's'. So the *dwd* generated for the *pwd* "pass" is "3z|<". This is presented in Table 2.

5. SECURITY ANALYSIS

JumbledKeys scheme is capable of defending against multiple attacks carried out at the client device, communication channel and the server. It never allows the actual password to get revealed in the client system. So it prevents malware attacks. Since the password being forwarded to server is a collection of random characters clicked by users based on their key-transfer scheme and also the keys are randomized each time based on a seed value, it acts as a one-time password varying for each authentication. The resistance of the proposed scheme against different attacks is analyzed below.

5.1 Shoulder Surfing Attacks

Since in this scheme users never click their password characters, anybody who watches the keyboard entry will not find the actual password being entered. If the keys within the groups are not randomized for every character entry, then the attacker may note down the keys being pressed by users as per the key-swap scheme and simply press the same keys without requiring the knowledge of the actual password. But since the keys are randomized, every time different sets of keys are clicked by the users. Attacker's only chance of learning the password is by recording a video or taking a photo snap of the keyboard entry and later analyzing it to construct the five possible passwords according to the five key-swap schemes. Our scheme offers stronger resistance against human SSA and only meager resistance against recorded SSA.

5.2 Client Side Attacks

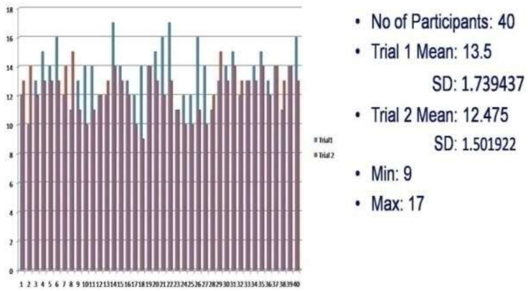
Because of the key-swap scheme, users press different keys each time. If attackers are able to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

acquire the user entered password value through form grabbing or web skimming or cross frame scripting attacks, they will only obtain the dynamic password. It is impossible to recover the real password from the dynamic password without the knowledge of the key-swap scheme and the keyboard layout during the character entry. So our scheme resists multiple attacks carried out at the client systems. Probability recovering actual password $P_R = 1 / 64^n$ for the password length of n.

6. USER STUDY

To ascertain the usability of our proposed, scheme we decided to conduct a user. We developed a prototype of the scheme in VB.net. The prototype has all the functionalities of the proposed scheme. We inducted 40 participants from our institution who were willing to help us for the usability analysis. After thoroughly explaining to them the scheme's benefit and its procedure, we requested them to try entering the same password five times in a week at any time of their choice. We installed our virtual keyboard on five computers in our lab. They were linked with a database holding the User ID and Passwords of the 40 participants. The keyboard tool recorded the duration of the password entry and number of wrong entries for each participant. Finally we collected the data after a week and prepared the chart based on that. It has been presented in Figure 5. It proves that JumbledKeys scheme has sufficient usability characteristics to be used in applications like Internet banking, e-banking etc.



- No of Participants: 40
- Trial 1 Mean: 13.5
  SD: 1.739437
- Trial 2 Mean: 12.475
  SD: 1.501922
- Min: 9
- Max: 17

Figure 5. User survey result

7. DISCUSSION

Securing password based user authentication by designing a user friendly virtual keyboard scheme was identified as the objective of the proposed scheme based on the literature survey. Accordingly, we have designed the JumbledKeys keyboard and thoroughly analyzed it both for its security and usability. Our scheme offers better security over the existing schemes that we studied due to the following reasons:

1. The proposed scheme is inherently two factor authentication by utilizing the registered mobile number for communication of key-transfer scheme
2. While existing schemes are capable defending against only one of the password attacks, our scheme can withstand multiple password attacks

Though our scheme offers adequate security against recorded shoulder surfing attack, the security can be breached by recording multiple password entries of a user. The password entry time is also prolonged under our scheme. In future, we should aim at reducing the duration of the password entry while averting multiple recording based attacks.

8. CONCLUSIONAND FUTURE WORKS

Despite the various security measures employed for preventing attacks on user authentications, the incidents of attacks are on the rise. Although new cryptographic schemes are incorporated into password transmission, their vulnerability during password entry by the users remains unaddressed. In this paper, we have presented a new password entry scheme through a virtual keyboard. It mitigates many attacks generally made over passwords. It facilitates the stealthy entry of passwords with the help of a key transfer scheme known only to the user and the server.

A dynamic password based on the user's actual password, key transfer scheme and the random position of keys on the keyboard is generated by the virtual keyboard and transmitted to the server. Even if an attacker extrats the dynamic password, it is not possible to learn the actual password from it. So in our scheme the passwords are secured from the time of entry by user to verification at the server. Our user surveys prove that the proposed scheme is usable.

The proposed scheme requires users to apply reasoning skills to correctly predict the target key based on the position of their password characters and the key-transfer method. While this may not pose a significant challenge for most users, some individuals may find it difficult to complete successfully. Older adults and people with certain memory-related conditions might face challenges using this system. Unlike traditional password entry, the proposed approach demands additional patience and time from users to complete the authentication process.

Even though the proposed scheme has attained satisfactory levels of security and usability, there is still scope for improving them. We need to develop schemes that can be utilized by all age groups and people with disabilities also.

REFERENCES:

[1] Kruzikova, A., Knapova, L., Smahel, D., Dedkova, L., & Matyas, V. Usable and secure? User perception of four authentication methods for mobile banking. *Computers & Security*, 115, 102603. (2022).

[2] Rajarajan, S., M. Prabhu, S. Palanivel, and M. P. Karthikeyan. "GRAMAP: Three stage graphical password authentication scheme." *Journal Of Theoretical & Applied Information Technology* 61, no. 2 (2014).

[3] Ali, M., Baloch, A., Waheed, A., Zareei, M., Manzoor, R., Sajid, H., & Alanazi, F. A simple and secure reformation-based password scheme. *IEEE Access*, 9, 11655-11674. (2021).

[4] Sun, H. M., Chen, Y. H., & Lin, Y. H. oPass: A user authentication protocol resistant to password stealing and password reuse attacks. *IEEE transactions on information forensics and security*, 7(2), 651-663. (2011).

[5] NordPas, "Top 200 most common passwords" URL: https://nordpass.com/most-common-passwords-list/, last accessed: 6 Aug 2023

[6] Yang, S., Ji, S., & Beyah, R. DPPG: A dynamic password policy generation system. *IEEE Transactions on Information Forensics and Security*, 13(3), 545-558. (2017).

[7] Bošnjak, L., and Bostjan Brumen. "Examining security and usability aspects of knowledge-based authentication methods." *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, (2019)

[8] Sun, Hung-Min, Shiuan-Tung Chen, Jyh-Haw Yeh, and Chia-Yun Cheng. "A shoulder surfing resistant graphical authentication system." *IEEE Transactions on Dependable and Secure Computing* 15, no. 2: 180-193. (2016)

[9] Zack Whittaker, "India's largest bank SBI leaked account data on millions of customers", URL: https://techcrunch.com/2019/01/30/state-bank-india-data-leak/, last accessed: 6 Aug 2023

[10] PolicyBazaar.com, "Biggest cyber breaches in India " ,URL: https://www.policybazaar.com/corporate-insurance/articles/biggest-cyber-breaches-in-india/, last accessed: 6 Aug 2023

[11] Bošnjak, Leon, and Boštjan Brumen. "Shoulder surfing experiments: A systematic literature review." *Computers & Security* 99: 102023. (2020)

[12] Aviv, Adam J., John T. Davin, Flynn Wolf, and Ravi Kuber. "Towards baselines for shoulder surfing on mobile authentication." In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pp. 486-498. (2017)

[13] Khan, Hassan, Urs Hengartner, and Daniel Vogel. "Evaluating attack and defense strategies for smartphone pin shoulder surfing." Proceedings of the *2018 CHI Conference on Human Factors in Computing Systems*. (2018)

[14] Damopoulos, Dimitrios, Georgios Kambourakis, and Stefanos Gritzalis. "From keyloggers to touchloggers: Take the rough with the smooth." Computers & security 32: 102-114. (2013)

[15] Nyang, DaeHun, Aziz Mohaisen, and Jeonil Kang. "Keylogging-resistant visual authentication protocols." *IEEE Transactions on Mobile Computing* 13.11: 2566-2579. (2014)

[16] Kwon, Taekyoung, Sooyeon Shin, and Sarang Na. "Covert attentional shoulder surfing: Human adversaries are more powerful than expected." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.6: 716-727. (2013)

[17] Li, Duo, Xiao-Ping Zhang, Menghan Hu, Guangtao Zhai, and Xiaokang Yang. "Physical password breaking via thermal sequence analysis." *IEEE Transactions on Information Forensics and Security* 14, no. 5: 1142-1154. (2018)

[18] Sood, Aditya K., Sherali Zeadally, and Richard J. Enbody. "An empirical study of HTTP-based financial botnets." *IEEE Transactions on Dependable and Secure Computing* 13.2: 236-251. (2014)

[19] Al-Hammadi, Yousof, and Uwe Aickelin. "Detecting bots based on keylogging

activities." 2008 *Third International Conference on Availability, Reliability and Security*. IEEE, (2008).

[20] Nelson, Tjada, Cory Nance, and Cherie Noteboom. "Web Injection and Banking Trojan Malware-A Systematic Literature Review." *2023 6th International Conference on Information and Computer Technologies (ICICT)*. IEEE, (2023).

[21] Shrivastava, Pragati, Mohd Saalim Jamal, and Kotaro Kataoka. "EvilScout: Detection and mitigation of evil twin attack in SDN enabled WiFi." *IEEE Transactions on Network and Service Management* 17.1: 89-102. (2020).

[22] Yeboah-Ofori, Abel, and Aden Hawsh. "Evil twin attacks on smart home IoT devices for visually impaired users." *2023 IEEE International Smart Cities Conference (ISC2)*. IEEE, (2023).

[23] Sushant, A. A., et al. "EvilSpot: Detection and Mitigation in Multi Channel." *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*. IEEE, (2023).

[24] Dastidar, Kanishka Ghosh, Olivier Caelen, and Michael Granitzer. "Machine Learning Methods for Credit Card Fraud Detection: A Survey." *IEEE Access* (2024).

[25] Sharma, Neeraj A., Arjun Pillay, and Mohammed Farik. "A Review Of Recent Cyber-Attacks In Fiji." *International Journal of Scientific & Technology Research 5.11*: 110-115. (2016)

[26] Chen, Shuo. Light-Weight Transparent Defense Against Browser Cross-Frame Attacks Using Script Accenting. Technical Report—MSR-TR-2007-29, Mar. 14, 2007 http://ftp. research. microsoft. com/pub/tr/TR-2007-29. pdf. Last accessed Oct. 5, 2007.

[27] Shashidhara, R., Kantharaj, V., Bhavya, K. R., & Lingareddy, S. C. Cross Channel Scripting Attacks (XCS) in Web Applications. In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021, Volume 1 (pp. 387-397). *Springer Singapore*. (2022).

[28] Kaur, Manjit, Manish Raj, and Heung-No Lee. "Cross Channel Scripting and Code Injection Attacks on Web and Cloud-Based Applications: A Comprehensive Review." *Sensors* 22.5 1959. (2022).

[29] Khedr, Walid I. "Improved keylogging and shoulder-surfing resistant visual two-factor authentication protocol." *Journal of Information Security and Applications* 39: 41-57. (2018).

[30] Corbett, Matthew, et al. "ShouldAR: Detecting Shoulder Surfing Attacks Using Multimodal Eye Tracking and Augmented Reality." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8.3: 1-23. (2024).

[31] Alsuhibany, Suliman A., and Saad G. Almutairi. "Making PIN and password entry secure against shoulder surfing using camouflage characters." *International Journal of Computer Science and Information Security* 14.7: 328. (2016).

[32] Guo, Yimin, Zhenfeng Zhang, and Yajun Guo. "Optiwords: A new password policy for creating memorable and strong passwords." *Computers & Security* 85: 423-435. (2019).

[33] Shukla, D., & Phoha, V. V.. Stealing Passwords by Observing Hands Movement. *IEEE Transactions on Information Forensics and Security*, 14(12), 3086-3101. (2019)

[34] Xiao, Yang, Chung-Chih Li, Ming Lei, and Susan V. Vrbsky. "Differentiated virtual passwords, secret little functions, and codebooks for protecting users from password theft." *IEEE Systems Journal* 8, no. 2: 406-416. (2012).

[35] Channabasava, H., and S. Kanthimathi. "Dynamic password protocol for user authentication." *Intelligent Computing-Proceedings of the Computing Conference. Springer*, Cham, (2019).