Journal of Theoretical and Applied Information Technology
15th April 2025. Vol.103. No.7

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

TrusPass: DYNAMIC PASSWORD GENERATION FOR USER AUTHENTICATION USING GRAPHICAL KEYWORD

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

User authentication is the gateway for entry into any system. Attacks on user authentication can cause irreperable loss for the individuals and organizations. Password based authentication is the primary method of user authentication. But they are vulnerable to many attacks. Imposing strong password policies to choose complex passwords causes difficulty in remembering them. People have the habit of using the same password across multiple accounts. This leads to cascading password attacks. Attacks on passwords commence from the moment they are being entered on the keyboard. In this paper, we have proposed a graphical keyboard based dynamic password generation scheme that facilitates inconspicuous entry of passwords. Users need to use a token for entering their passwords on the proposed keyboard. By aligning the token with the characters on the keyboard, they can unobstrusively enter their passwords. User's password gets transformed into a high entropy dynamic password with the help of an algorithm. For the same password, different dynamic passwords are geberated each time. The scheme improves password strength without burdening users to remember complex passwords. The proposed scheme averts many attacks on passwords. The usability of the scheme is ascertained through a user study
Keywords: *User authentication, Password attacks, Internet banking, Shoulder-surfing, Form grabbing, Keylogging, Virtual keyboard, Cyber security*

1. INTRODUCTION

In today's world everybody is required to maintain an average of 25 online accounts for different services. Each account is associated with a user id and a password [1]. It is a common practice for people to choose simple passwords for easy remembrance. It has been discovered that most people frame their passwords with the help of commonly used words, popular terms, places related to them, dates such as date of birth or wedding day and names of their dear ones to overcome the difficulty of remembering complex passwords. This makes the passwords vulnerable to dictionary and guessing attacks [2]. Another common mistake people make in password selection is choosing the same password for several accounts to reduce the burden of remembering too many passwords. Imposition of stringent rules in password selection policies further increases the difficulty in remembering passwords.

Attacks on passwords could happen on three potential locations – at the client's device, on the communication channel or at the server. Cryptographic encryption schemes limit the chances of communication and server attacks. By employing encryption algorithms and hash functions, passwords are concealed from attackers during transmissions

Attacks at the client device are the most difficult ones to prevent because password entry is generally a direct keying process using the keypads. During the entry, passwords could be captured through keylogging, form grabbing or shoulder surfing attacks (SSA) [3]. Form grabbing is a type of Man-in-the-Browser attack in which the HTML form entries are captured before they are submitted [4]. Since the data are grabbed from the form inside the browser, the cryptographic protocols cannot prevent this attack. Using this attack user's credentials can be stolen when they are typed on HTML form controls. SSAs are either carried out manually or with the help of sophisticated recording tools [5]. There are several virtual keypad schemes proposed to resist the Human shoulder surfing attacks. But it is very challenging to resist the recorded shoulder surfing attack. The attacker has the convenience of replaying the password entry

multiple times until he figures out the actual password.

There are numerous secured virtual keypad schemes proposed to defend against recorded SSAs. But many of them are poor in usability since they introduce immense complexity in the entry process. The proposed schemes are also specific to tackling SSA. The research question that we attempt to answer in our proposed scheme is "whether it is possible to design a password scheme that is both resistant to multiple password attacks and remains user-friendly?".

In this paper we present a novel virtual keyboard scheme named as TrusPass with an on-the-fly password transformation algorithm for converting the actual passwords into dynamic passwords. Since the dynamic password is different during each password entry, caputuring it to impersonate the user is impossible. Our scheme mitigates many well known password attacks. The proposed scheme balances security and usability.

In summary, proposed scheme offers the following security features:

- Comprehensive resistance against attacks of keylogging, shoulder surfing, form grabbing, evil twin and web skimming.
- A key transformation scheme that converts actual password into dynamic password comprising of random characters. So actual passwords are never revealed to attackers from the moment of password entry to password verification.
- Conversion of weak passwords into high entropy strong passwords which are distinct for each time of log in.
- Introduction of voice assisted keyboard. This eliminates the possibilities of shoulder surfing attacks.
- Even if there is no secured channel available between client and server, password security is ensured.
- A novel graphical keypad that incorporates human friendly cartoon images to improve the usability of the keypad.

The rest of the paper is organized as follows. Section 2 explores the related works proposed. In section 3 we introduce and discuss the proposed scheme. The security analysis is provided in section 4 and user study is elaborated in section 5. A discussion on the strengths and limitations are described in section 6. Section 7 contains the conclusion of the paper.

1.1 Problem Statement

The goal of our research is to develop innovative and secure user authentication schemes that enhance password-based authentication while resisting various password attacks. Our approach prioritizes both security and user-friendliness, ensuring that users can authenticate safely without fear of their accounts being compromised.

2. RELATED WORKS

The vulnerability of passwords against stealing while they are being entered has been recognized by security experts. The existing research works on secured virtual keyboard schemes are falling into two categories - schemes that are aimed at resisting PIN entry attacks and schemes that prevent attacks on password entry.

2.1 PIN Entry Schemes

A PIN is a secret number used for authentication. Usually PIN numbers are made of four digits. They are generally associated with internet banking accounts, credit and debit cards and mobile banking apps. The user is required to remember and correctly enter it to prove their authenticity to do transactions. Many researchers have made use of invisible cues and counting as a technique to allow the users enter their PIN numbers secretly. A PIN entry method based on audio or haptic cues is proposed in the name of SpinLock[6]. Users have to listen to the audio or haptic clues and rotate on a dial for alternate directions of left and right for the number times equivalent to their PIN digit value. To overcome the observation attack the spatial distance to be travelled between cues is randomized. Two variants of PIN entry schemes called LIN4 and LIN5 are proposed in [7]. In this scheme, the user has to arrange his PIN digits against the session key which is the symbol displayed against the first PIN digit in round 1. In LIN4 the first PIN digit is selected by default but in LIN5 user must explicitly enter the first digit also. This scheme prevents the HSSA attacks but fails against RSSA. A PIN entry scheme that was specifically designed for ATMs is IFTTPIN [8]. The four PIN digits are associated with three colours. At the time of PIN entry the user has to recognize the alphabet under his PIN digit which is displayed in the colour associated with that

digit and enter it through the keypad. In the scheme called phoneLock, users are shown a blank wheel comprising of 10 targets [9]. The PIN entry is facilitated by audio or haptic cues regarding the positions of digits on the keypad. A similar PIN entry interface is provided in [10]. Under this scheme the ten digits from 0 to 9 are shown along with ten alphabets. The user has to listen to the audio channel for an alphabet conveyed to him by the system. This alphabet has to be positioned below the PIN digit of that iteration. In this way he has to position four alphabets for four digits of his PIN numbers. A PIN entry scheme that introduces camouflage characters into actual password characters to confuse attackers is presented in [11]. The proposed keypad contains activator and deactivator keys. While the user is typing his password, he may suddenly disable the keypad by entering the deactivator key. Then the keys pressed by the user are all ignored by the system and are not included into the password. He may reactivate the keypad by typing the activator key. SpinPad is a PIN entry scheme with a circular keypad [12]. It has two circles. The inner circle contains ten alphabets in random order. The outer circle comprise of the ten digits from 0 to 9. Before entering the PIN number, user must know his token alphabet for PIN entry. This is conveyed by the system through the headphone or earphone. Using the token user must enter his PIN digits by aligning the token in the same columns of his PIN digits. In the scheme of [13], for each PIN digit an audio challenge in the form of an alphabet is conveyed to the user through the headphone. Now the user has to align his PIN digit with that alphabet. For each PIN digit different alphabets will be conveyed as challenge. Under the UTP scheme proposed in [14], users can transform their PIN number into a one-time random number with the help of a Visual Pattern that they are provided at the time of registration. This scheme protects the PIN from various attacks.

2.2 Password Entry Schemes

A password is a alphanumeric value. Passwords are relatively stronger against attacks than PIN numbers. But they are still susceptible to attacks. The easiest attacks on passwords are the shoulder surfing and keylogging attacks. With these attacks the attacker is able 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 proposed. A major issue with these schemes is their usability. A blank keypad is displayed on the screen in [15]. Then the user 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 the keypad layout. To conceal the actual password characters, random camouflage characters are typed along with the actual characters [16]. 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 [17]. 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 secured virtual keypad based on keypad randomization is proposed in [18]. The keys on the virtual keypad are divided into four groups. User has to select a key-transfer pattern out of four possible options at the time of enrollment. Then during password entries, to enter each password character user has to first observe the position of his password character and should press rotate button. Now the keypad character groups will be shifted as per the key-transfer scheme and the keys are hidden. Now the user must recall the transfer scheme he had selected and identify the index of his password character in the target key group where it should have been shifted. User should press the button which is supposed to be the location of his password character. In this way user must enter all the password characters. A visual keypad scheme that comprise of emoticons is proposed in [19]. Each key on the keypad is assigned a unique emoticon. During registration 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 user may move the emoticons across the keypad. In order to enter the password character, user 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 user must repeat this step for the remaining password characters. Diksha Shukla *et. al*. [20] have demonstrated how

passwords can be stolen by recording the hand movements of users when they are typing their passwords.

2.3 Dynamic Password Generation

There are some schemes that try to generate a dynamic password each time. The dynamic passwords differ for each login. Kumar, B. P et. al.[21] have proposed a scheme for the generation of passwords based on multiple secret paramenets from the users. They also demonstrated that the cracking time of their passwords is longer than the traditiobnal passwords. Shukun Yang et. al [22] 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 is another such scheme for dynamically generating password policies based on the frequency of password characters proposed by Anuraj Singh et. al.[23]. In the scheme named B-Dash, a dynamic graphical password generation scheme for the mobile devices [24]. In the scheme proposed by Xiao, Yang, et al [25], users can generate a virtual password dynamically by applying some function over their actual password. This scheme involves some amount of human computation.

Through the literature survey, we identified the following limitations in the existing schemes which our proposed scheme is expected to overcome.

1). Schemes that we reviewed only focus on tackling shoulder surfing attacks. They do not protect against other types of password attacks. So in order to mitigate different attacks multiple solutions needs to be implemented. We need to design an integrated solution that prevents several password attacks.
2). Although the existing schemes are resistive against human shoulder surfing attacks, many of them fail against recorded-shoulder surfing attacks. For example, the scheme of [19] [7] offer protection against single recording based attacks. But if an attacker is able to record the password entry for few times, then he could trace the password. The proposed scheme should withstand shoulder surfing attacks based on multiple recordings of the password entry.
3). Despite the improved security achieved by password entry schemes, passwords are still vulnerable to attacks. So a two factor authentication promises stronger security over a single factor authentication [26].
4). Many schemes have disregarded the fact that passwords are usually stored in hash forms. Schemes such as [7] require that the plain text of the password should be available at the server to verify the password. The proposed password scheme should be compatible with the hashed form of the password.

We analyzed and compared many existing schemes with our proposed scheme. Result of the analysis is summarized in Table 1. Our scheme provides security against several password attacks. But other schemes only offers immunity against shoulder surfing attacks..

3. PROPOSED SCHEME

3.1 System Model

There are three parties involved in our TrusPass scheme: users (represented as *U*), a client computer through which users submit their passwords (represented as *C*), and the server (represented as *S*) which verifies users' passwords. We make the following assumptions:

- Headphone or earphone available with the users to hear the voice messages from the keypad application
- Changes could be made at the server application to implement our proposed scheme
- Users should be willing to spend additional time for entering their passwords
- Users are willing to expend additional time for entering their passwords considering the added security of their accounts.

The roles of the user, client device and the server in the proposed framework have been summarized in the system model provided in Figure 1.

Figure 1: System model

3.2 Step-by step Procedure of Password Entry

The following steps summarize the password entry process:

Step 1: User enter the user id and press the button captioned as "Enter Password"

Step2: Server receives the user id, checks

whether it exists and generates T, constructs a virtual

keyboard with 54 keys, loads 54 images randomly on the keys and returns keyboard to *C*

Step 3: TrusPass graphical keyboard with 54 keys and 54 images appear on the computer screen of *U*. The keys also contain 54 numbers from 1 to 54 referred to as Position Indicators (*P*) under the images

Step 4: User is conveyed the value of *T* by voice through the headphone or speaker

Step 5: User locates the key that has *T* as its *P* and notes down the image present on that key as the image token Δ for that transaction.

Step 6: User clicks start button to enter the password

Step7: Now the 54 keys on the keyboard are loaded with the actual characters with the 54 pictures which are now randomized

Step 8: Password entry with the help of Δ commences

Step 9: User locates the key with Δ and recalls the first character of their password

Step 10: User uses the appropriate navigation buttons and listens to navigation hints through the headphone to trace the virtual movement of Δ for moving Δ towards the first password character pwd_1

Step 11: User clicks the Enter button once Δ aligned with the key containing pwd_1

Step 12: *P* on the keyboard are randomized again

Step 13: User have to repeat Step 3 and 4 to enter the remaining password characters.

Step 14: After entering all the characters of *pwd*, *U* press the Submit button

Step 15: *dwd* value gets generated based on *pwd* and it is forwarded to *S*

The above steps will be repeated every time user wishes to authenticate using their registered password.

Table 1: Comparative Evaluation of various schemes

	Usability					Security				
Scheme	Memory-wise Burden	User-friendly Interface	Need of Additional Resources	Wrong Password entry rates	PIN \ Password Entry Time	Multiple PIN\ Password Attacks	HSSA and RSSA Resistance	Two factor authentication	Compliance with Existing Password	Transforming Weak passwords to Strong
VisualAuthentication [15]	L	M	Y	L	L	Y	M	N	Y	N
SSA Resistant Virtual Keypad [18]	M	M	N	H	H	N	L	N	Y	N
Optiwords[17]	H	H	N	H	L	N	L	N	Y	N
oPass[35]	L	H	N	L	M	N	M	N	N	N

SSSL[31]	M	M	Y	L	M	N	M	N	Y	N
DragPIN[36]	L	H	N	L	M	Y	H	N	Y	N
IFTT[8]	H	H	N	H	M	Y	M	N	N	Y
SwitchPIN[37]	M	H	N	M	M	Y	M	N	N	Y
Spinlock[6]	M	M	Y	H	H	Y	Y	N	N	N
RotaryPIN[38]	L	H	N	L	M	N	L	N	N	N
TrusPass[PROPOSED]	M	H	Y	M	M	Y	H	Y	Y	Y

Legends: L- Low, M-Moderate, H–High, N- No, Y- Yes

4 DESIGN OF TRUSPASS KEYBOARD

Proposed graphical keyboard is used for entering alphabets, numbers and symbols. It has 54 characters comprising of 26 alphabets, 10 digits and 17 symbols and a CAPS lock. The keyboard is displayed in two versions. The initial version is meant for choosing the image token Δ as per value of *T*. The second version of the keyboard is the one to be used for password entry. In the initial keyboard, each key contains a number along with a picture ☺ that users can remember easily. So, all the 54 keys on the keyboard have 54 unique pictures. Users will have to identify the ☺ based on the token number T conveyed by the system as their Image token Δ for that password entry. In our sample keyboard layout, we have used cartoon images of animals. There are four arrow buttons representing left, right, top and bottom to move Δ to the keys of the password characters on the keyboard. Once Δ is identified, user should press the "Image Token" button to indicate that Δ selection is completed. Now the second version of the keyboard is loaded and displayed. Figure 2(a) presents the initial keyboard of TrusPass scheme

(a)

(b)

Figure 2. (a) Initial keyboard (b) During password entry

Now the user is ready to enter their password through the second version of the keyboard. The keys now contain the character set (0 to 9, a to z, symbols) along with the same ☺ pictures. Figure 4(b) shows a sample keyboard during password entry. By recalling the Δ selected in the first stage, users should complete the entry of their passwords. The methodology of password entry on the keyboard is described in the coming section.

5 PASSWORD ENTRY

The overall process involved in the proposed scheme is explained in this section.

5.1 Overview of Password Entry

During registration, users need to submit their preferred password *pwd* through the TrusPass graphical keyboard to store it at the server. To do that, , first users have to enter the user id and press the Password button. When server receives the user id, it generates a random number T, constructs the initial version of the TrusPass keyboard by randomly aligning ☺ on the keys of the keyboard and sends them to the client system. Now the TrusPass keyboard is displayed on the screen. The keys contain numbers between 1 to 54 with a unique ☺ on each key. Users are expected to wear

headphones as the password entry involves audio messages. But password entry can also be done with the help of computer speakers but with less security. Now the number T is conveyed by voice to the users. It is a number in the range of 1 to 54. User locates the number of T on the TrusPass keyboard and identifies ☺ that is on the key that contains T. It is treated as Δ for that login. With the help of Δ, users have to enter the password *pwd*. For each character of pwd_i of *pwd*, users have to move Δ to the key that contains that pwd_i character. But the movement of Δ happens without any visual movement on the keyboard with the help of audio navigation hints. This causes the password entry to be totally invisible and secured against any kind of observation attacks. As *pwd* gets entered, it is transformed into *dwd* as per a conversion algorithm to be presented latter. After entering all the *pwd* characters, users click the submit button. Then the *dwd* is sent to the server. Server extracts *pwd* from *dwd*, computes the hash of the password, H(*pwd*), and compares it with stored H(*pwd*) for authentication. The following section explains the system model of the scheme.

5.2 Dynamic Password Generation

In this section, the procedure for converting *pwd* into *dwd* at the client side is explained. A list of 54 random numbers from 1 to 54 are generated and stored in array E. These numbers are regenerated after user presses the ENTER key after entering a character pwd_i of the password. Server has the same seed for regenerating the same random list of numbers while verifying the password. During password character conversion, the initial position of Δ in the image vector *M* is referred to as J_i. Then the value of *O* is nothing but $E[J_i]$. After users virtually move Δ to their respective password character and press the Enter button, the current position of Δ in *M* is identified as *Jc*. Then the value of E[*Jc*] is taken as the target value *R*. Using *O* and *R*, the distance *D* can be computed as follows:

$$D = ((54 - O) + R) \; Mod \; 54.$$

The value of *D* is then indexed into the key vector *V*. The character stored at *V[D]* is retrieved. It becomes the dwd_i character replacing pwd_i. Immediately after pwd_i is entered, E array is replaced with the new list of random numbers before users enter the next password character. Due to the use of E in the calculation of dynamic password, even if the passwords contain same characters multiple times, the dynamic password will have different characters for each of its occurrence. Finally client system forwards *dwd* to server for verification.

Table 2. Keys Vector

1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	I	J
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
k	l	m	n	o	p	q	r	S	t
21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
u	v	w	x	y	z	0	1	2	3
31	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**
4	5	6	7	8	9	~	!	@	$
41	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**
%	^	&	*	(	)	-	_	+	=
51	**52**	**53**	**54**						
;	:	“	\						

For a sample password conversion, consider the TrusPass keypad in Fig. Assume that *I* of the user is the “dinosaur” image which is at the 47h key. Then *P* at the 47h key is 12. Assume that the invisible *E* value at that key is 34.Then 34 is the value of *O*. If the password character being entered by the user is “s’ then 14 is the *P* value of *T*. Let us assume that *E* value at the *R* position is 50. Then 50 is the value of *R*. Then the distance *D* is computed as *D* = ((54 – 34) + 50) Mod 54. Then the distance *D* is 16. The alphabet “p” is found at V[16]. So “p” is inserted into the dynamic password *dwd* as the replacement character of “s” in *pwd*. If “s” occurs again in the same password, since *E* value changes after each password character entry, the resultant character for *dwd* is different now. This characteristic prevents the attacker from learning the repetition of password characters even if they gain access to *dwd*. In Table 3, dynamic passwords generated for the entry of the password “pass1234” during different attempts are provided. It is evident from the table that the weak password “pass1234” was turned into stronger passwords by our scheme. Along with the passwords, the approximate times needed to crack those passwords are also given. These times were taken out of the online password strength checker tool provided by my1login.com [27].

Algorithm pwd-to-dwd

Input: d – Array of random numbers used for each password character entry
T– Token number sent the client
I - Array of Image ids of the images displayed on the keyboard

C – Array of 54 characters comprising 26 letters, 10 digits and 14 symbols

Output: *dwd* – Dynamic password generated

```
1   j=0
2   i= I[T]
3   repeat
4           l=Locate-Image-Token(i)
5           O= d[l]
6           l=Locate-Image-Token(i)
7           R= d[l]
8           G = ((54 - O ) + R) mod 54
9           dwd[j] = C[G]
10          j=j+1
11          shuffle-numbers(d)
12  until Submit = false
13  return dwd
```

Table 3: Dynamic Passwords Generated for "pass1234"

Dynamic Password	Time to crack
zs@jpm0+	61 years
33gmcn+i	24 years
@3f71/2^	7 years
!/3xd$p9	47 ears

5.3 Password Verification

When *dwd* is received from client, server retrieves *K* pertaining to the user. The *dwd* is converted to the actual password characters of *pwd* with the help of the following formula.

$$R = (D + O) \text{ Mod } 54.$$

The algorithm PasswordVerify given below is used to retrieve *pwd* from *dwd*. *T* represents the token value that was sent by *S* to *C*. *M* is the vector of image ids that are displayed on the keyboard for that authentication. The image id present at the index value of *T* in M is taken as the image token *I* of the user. *E* vector consist of the pseudo random numbers shared between *C* and *S*. E_1 represents the *E* vector before the user moved *I* to the target and key E_2 represents the *E* vector after the user moved *I* to his password character. *V* is the key vector. D_i is the distance value equivalent to the character of dwd_i. If the user has pressed the CAPS lock then *R* will be 0. No character is inserted into *dwd* but the subsequent character is expected to be an alphabet to which the CAPS lock will be applied. If it is not an alphabet then it is reported as a wrong entry.

Similarly if CAPS lock entered twice consecutively then that is also reported as wrong entry. Generally passwords are stored in hash format as a precaution against server side attacks [28]. Hashing algorithms such as SHA, MD5 or Bcrypt are used to hash the passwords and they are stored in the databases of the servers. After *dwd* is converted into pwd, the *h(pwd)* is found and it is matched with the stored hash. If they are the same then authentication is successful otherwise authentication has failed.

Algorithm PasswodVerify

Input: I - Array of Image ids
C- Array of 54 characters
dwd- dynamic password submitted by user

Output: *pwd* – Password of the user

```
1   i=0
2   j=0
3   k=0
4   l= I[T]

5   while i<>pwd.length
6           whileC[j] ≠dwd[i]
7                   j=j+1
8     while d[k] ≠ j
9                   k=k+1
10    G=d[k]
11    l=Locate-Image-Token(i)
12    O=d[l]
13  R = (G +O) Mod 54
```

5.4 Dynamic Password Proof

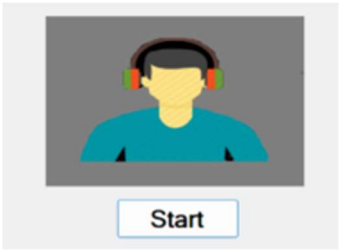
TrusPass based dynamic passwords expire after single use. Each time of login, a different password is generated as per the TrusPass keyboard generated by the server. Server places the 54 pictures ☺ on the 54 keys of the keyboard based on a pseudo random algorithm. Consequently, the probability of the same password repeating is $(1/54!)^n$, where n is the number of times the password is generated. This probability is so close to zero that it can effectively be considered negligible.

5.5 Sample Password Entry

In this section we present the password entry for the password value "pass" through the TrussPass keyboard. In the first stage the user hears the number for the Token assigned by the server. In this example 15 is conveyed as the Token. Next the keyboard for image token selection is displayed.

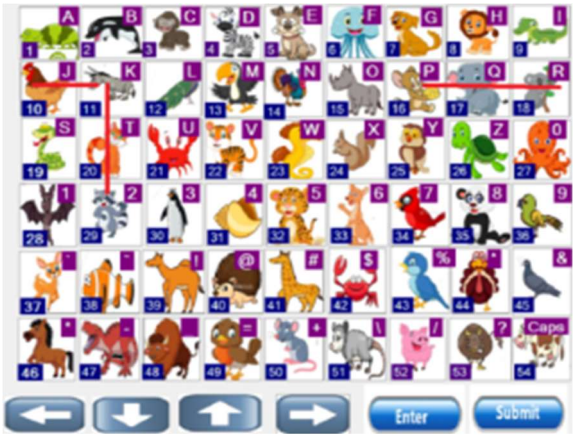
User observes the keyboard and notes down the animal "fox" at 15th key as the Image token. Now the keyboard for password entry is shown. User aligns the image token to his first password character which is "p" by pressing the Up arrow button twice and Left arrow once. Then the user presses the Enter button to submit "p" as the password character.

In this way, the user accomplishes the entry of all the remaining characters and finally presses the Submit button. The screen shots of the keyboard while entering "pass" are presented in Figure 3.



(a)

(b)



(c)

(d)

(d) (e)

Figure 3: Sample Password Entry (a) Token number 15 conveyed by voice (b) User identifies Δ as the image on key 15 (c) User aligns Δ to the first password character "p" (d) User aligns Δ to the second password character "a" (e). User aligns Δ to the third password character "s"

6 SECURITY ANALYSIS

In this section, we analyze the security of our proposed scheme against different attacks. We assume that attacks could be actuated from any of the four possible locations: client system, mobile phone and the server. Our scheme does not allow the attacker to learn the password by successfully attacking any one of those points.

6.1 Dynamic Password Leaked

After the user completes the password entry on the TrusPass keypad, a QRCode is generated encompassing the dynamic password *dwd*. When a user captures the QRCode through the mobile app, an attacker may also capture the QRCode without the user's knowledge. Then the attacker may find out the *dwd* value from the QRCode. With the help of *dwd*, the values of *D* can be easily found since *dwd* comprises the characters from the V vector. The indexes of the *dwd* characters in *V* are the values of D. But the attacker also needs to know the values of *E* and *O* to trace *R*. *E* is a collection of 54 random numbers shuffled using Fisher Yates algorithm [29] with *T* value as the initial feed for the character dwd_1. For subsequent characters, *O* values of the previous characters are used as the feed. In order to detect the *O* values, the attacker must correctly identify Δ of the user and the particular key among the 54 keys on which it was present. For a password of 8 characters $P_T = 54$. If E is found then $P_O = 8 / 54^8$. So it is not possible for the attacker to trace the user's password with the help of the *dwd*.

6.2 Shoulder Surfing Attacks

Shoulder surfing attacks are carried out by observing or recording the screen activities during the password entry and then trying to trace the password typed. But in our scheme there is no screen activity. In the Δ selection phase, attackers get to see the keypad with images randomly positioned on the keys. But they cannot identify Δ since they do not hear the *T* which is conveyed through the headphone. Because the entire password entry is facilitated by voice navigation support, the attacker gets no clue about which key contains Δ and to which key it is moved. Many of the proposed schemes are vulnerable to multiple recording SSA even though they are resistive against single recording SS. But our scheme can withstand attacks involving any number of recordings.

6.3 Form Grabbing Attacks

Form grabbing attacks have become widespread in recent times for stealing confidential information entered by users. Here the malware that enters into the victim's system grabs the data that user types on the HTML forms and shares it with the attacker. This attack is very effective because the data entered by the user is captured before they are encrypted. Under our proposed scheme since the password characters are converted into random characters as the user types in the

characters, even if the user credentials are acquired by the attacker it will not be possible to know the actual password.

6.4 Evil Twin Attack

In our proposed scheme the user's password is converted into a high entropy dynamic password. The malware that is inserted into the free public Wi-Fi usage could put user's credentials under risk. Not every public Wi-Fi employs strong encryption protocols. Similarly through an evil twin attack, the attacker may trap the legitimate user to join a fake Wi-Fi network that has been setup [30]. When users enter their account details, they are traced by the attacker. But under the TrusPass scheme, the user's actual password is not entered during login. Attacker could only obtain the dynamic password through the Wi-Fi transmission. Table 4 compares the protection offered by different schemes against client side attacks. It shows that our scheme resists many attacks.

Table 4. Comparison of the security of schemes

Scheme	SSA	Other Attacks
SSSL [31]	HSSA:✓ RSSA: ✓	X
gTapper,gRotator *gTalker*[32]	HSSA: ✓ RSSA: ✓	X
EyePassword[33]	HSSA: ✓ RSSA: ✓	X
DE-PAKE[34]	HSSA: X RSSA: X	✓
TrusPass[Proposed]	HSSA: ✓ RSSA: ✓	✓
HSSA: Human Shoulder Surfing Attacks RSSA: Recorded Shoulder Surfing Attacks		

7 USABILITY ANALYSIS

It is important for user authentication schemes to be user friendly. A highly complex system will not be opted by users even if it offers superior security.

7.1 User Study

The motive of conducting the user study is to find out answers for the following questions:

- How much additional time is taken for entering password through the TrusPass keyboard?
- How many wrong password entries are made?

These two metrics are crucial indicators to determine the usability of our scheme.

We identified 115 persons for the user study. It comprised 58 male and 57 female members who were belonging to different age groups. The maximum age of the member was 56 and the minimum age was 21. We shared the introductory video on our proposed scheme. Participants were shared the URL of the TrusPass web application and their login ids. They were requested to register their chosen passwords by entering them through the TrusPass keyboard. After registration they were asked to enter their passwords 15 times at their convenient times within one month. We obtained the total number of entries, number of wrong entries, number of correct entries made in the first attempt and duration of entries of all the participants from the server.

Password Entry Time: Only the successful password entry times were considered. The password entry time commences from the loading

of the application and ends when the user presses the submit button. As we expected, the password entry times were longer than the conventional direct password entry time. But from the data we were able to sense that the password entry time decreases as the user gets accustomed to our keyboard by performing more entries. Age of the participants also had an impact on the entry time. Young participants were relatively quicker. But there was no noticeable difference in the password entry time based on the gender. Table 5 presents the minimum and maximum password entry times.

Wrong Password Entries: 36 participants entered passwords wrongly in their first attempts. But the number of wrong entries reduced in the subsequent attempts. Only 5 participants wrongly entered passwords during more than 10 entries. 4 participants never entered their passwords wrongly in all the 15 entries. We found that the number of wrong entries was initially high and it reduced after several entries were completed as users got used to the keyboard. We could not see any difference in the number of wrong entries based on age and gender.

Participants Feedback on TrusPass Keyboard: With the help of the questionnaire filled by the participants, we could infer the following points. TrusPass scheme requires more concentration and focus to correctly enter passwords unlike direct password entry schemes which can be completed almost effortlessly. Participants opined that they were not nervous about possibility of somebody behind them observing their screen and knowing their passwords while entering through TrusPass keyboard. They are willing to accept TrusPass keyboard as an option for password entry. The user feedback results are provided in Table 6.

Table 5. Password entry time

Mean	Min	Max	Sd
18.9	16	23	1.2165

Table 6: User Feedback about TrusPass scheme

Question	Yes	No
Are you apprehensive that your accounts could be breached by stealing passwords?	84 %	16 %
Do you trust that TrusPass could protect your passwords from attackers?	76%	24 %
Do you opine that TrusPass tool is convenient to use for password entry ?	64 %	36 %
Whether you like the Graphical keyboard designed in the scheme?.	80 %	20 %
Will you adopt TrusPass keyboard scheme for your regular usage?	72 %	28%

8. CONCLUSION S

Despite the difficulties and vulnerabilities, passwords offer the flexibility of changing their values periodically and to maintain different values for different accounts. People are acquainted with password based user authentication. So strengthening password security is an essential need. In this paper we have proposed a novel graphical keyboard scheme that elevates password security by dynamically changing the password characters as they are being typed by users. The same password transformed into different dynamic passwords by the proposed keyboard. So even if

the user reuses the same password for multiple accounts, our scheme mitigates its vulnerability by changing it into different values each time. Since the password entry in our scheme requires human participation for moving the image token to the password characters, it is not possible to carry out automated brute force attacks against password servers. Attackers can not retrieve the actual password from the user's device using any client side attacks. Our security analysis proves that proposed scheme is able to withstand many common attacks on passwords.

A drawback of our scheme is the additional time taken to enter passwords since it requires positioning image token over the password characters. But this can be justified by the substantial security that our scheme provides against multiple attacks. In future, we plan to design schemes with minimal password entry times without compromising the usability and security

REFERENCES:

[1] Rajarajan, S., And Plk Priyadarsini. "Jumbledkeys: Two Factor User Authentication Scheme Using Partitioned Virtual Keyboard." *Journal of Theoretical and Applied Information Technology* 102.21 (2024)..

[2] Nehme, A., Li, M. L., & Warkentin, M. (2024). Adaptive and Maladaptive Factors behind Password Manager Use: A Hope-Extended Protection Motivation Perspective. *Computers & Security*, 103941.

[3] Binbeshr, F., Siong, K. C., Yee, L., Imam, M., Al-Saggaf, A. A., & Abudaqa, A. A. (2025). A systematic review of graphical password methods resistant to shoulder-surfing attacks. *International Journal of Information Security*, *24*(1), 1-22.

[4] Thanh Vu, S. N., Stege, M., El-Habr, P. I., Bang, J., & Dragoni, N. (2021). A survey on botnets: Incentives, evolution, detection and current trends. *Future Internet*, *13*(8), 198.

[5] Bošnjak, L., & Brumen, B. (2019). Shoulder surfing: From an experimental study to a comparative framework. *International Journal of Human-Computer Studies*, *130*, 1-20.

[6] Bianchi, A., Oakley, I., & Kwon, D. S. (2011). Spinlock: A single-cue haptic and audio PIN input technique for authentication. In *Haptic and Audio Interaction Design: 6th International Workshop, HAID 2011, Kusatsu, Japan, August 25-26, 2011. Proceedings 6* (pp. 81-90). Springer Berlin Heidelberg..

[7] Lee, M. K. (2014). Security notions and advanced method for human shoulder-surfing resistant PIN-entry. *IEEE transactions on information forensics and security*, *9*(4), 695-708.

[8] McConkey, K., Ayranci, T. E., Khamis, M., & Grizou, J. (2024). IFTT-PIN: A Self-Calibrating PIN-Entry Method. *arXiv preprint arXiv:2407.02269*.

[9] Bianchi, A., Oakley, I., Kostakos, V., & Kwon, D. S. (2010, January). The phone lock: audio and haptic shoulder-surfing resistant PIN entry methods for mobile devices. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction* (pp. 197-200)..

[10]Lee, M. K., Nam, H., & Kim, D. K. (2016). Secure bimodal PIN-entry method using audio signals. *Computers & Security*, *56*, 140-150.

[11] Alsuhibany, S. A., & Almutairi, S. G. (2016). Making PIN and password entry secure against shoulder surfing using camouflage characters. International Journal of Computer Science and Information Security, 14(7), 328.

[12] Rajarajan, S., Kalita, R., Gayatri, T., & Priyadarsini, P. L. K. (2018). Spinpad: a secured pin number based user authentication scheme. In *2018 International Conference on Recent Trends in Advance Computing (ICRTAC)* (pp. 53-59). IEEE.

[13] Lee, M. K., Nam, H., & Kim, D. K. (2016). Secure bimodal PIN-entry method using audio signals. *Computers & Security*, *56*, 140-150.

[14] Rajarajan, S., & Priyadarsini, P. (2019). UTP: a novel PIN number based user authentication scheme. *Int. Arab J. Inf. Technol.*, *16*(5), 904-913.

[15] Nyang, D., Mohaisen, A., & Kang, J. (2014). Keylogging-resistant visual authentication protocols. *IEEE Transactions on Mobile Computing*, *13*(11), 2566-2579.

[16] Alsuhibany, S. A., & Almutairi, S. G. (2016). Making PIN and password entry secure against shoulder surfing using camouflage characters. *International Journal of Computer Science and Information Security*, *14*(7), 328.

[17] Guo, Y., Zhang, Z., & Guo, Y. (2019). Optiwords: A new password policy for creating memorable and strong passwords. *Computers & Security*, *85*, 423-435.

[18] Rajarajan, S., Maheswari, K., Hemapriya, R., & Sriharilakshmi, S. (2014). Shoulder surfing resistant virtual keyboard for internet banking. *World Applied Sciences Journal*, *31*(7), 1297-1304.

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

[20] Shukla, D., & Phoha, V. V. (2019). Stealing passwords by observing hands movement. *IEEE Transactions on Information Forensics and Security*, *14*(12), 3086-3101.

[21] Kumar, B. P., & Reddy, E. S. (2020). An efficient security model for password generation and time complexity analysis for cracking the password. *Int. J. Saf. Secur. Eng*, *10*, 713-720.

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

[23] Singh, A., & Raj, S. (2022). Securing password using dynamic password policy generator algorithm. *Journal of King Saud University-Computer and Information Sciences*, *34*(4), 1357-1361.

[24] Andriotis, P., Kirby, M., & Takasu, A. (2023). Bu-Dash: a universal and dynamic graphical password scheme (extended version). *International Journal of Information Security*, *22*(2), 381-401.

[25] Xiao, Y., Li, C. C., Lei, M., & Vrbsky, S. V. (2012). Differentiated virtual passwords, secret little functions, and codebooks for protecting users from password theft. *IEEE Systems Journal*, *8*(2), 406-416..

[26] Shirvanian, M., Saxena, N., Jarecki, S., & Krawczyk, H. (2019). Building and studying a password store that perfectly hides passwords from itself. *IEEE Transactions on Dependable and Secure Computing*, *16*(5), 770-782..

[27] https://www.my1login.com/resources/ password- strength-test/

[28] Lee, M. K., Yoo, J., & Nam, H. (2017). Analysis and Improvement on a Unimodal Haptic PIN-Entry Method. *Mobile Information Systems*, *2017*(1), 6047312.

[29] Eberl, M. (2016). Fisher-yates shuffle. *Arch. Formal Proofs*, *2016*, 19. (2016).

[30] Guo, R. (2019). Survey on WiFi infrastructure attacks. *International Journal of Wireless and Mobile Computing*, *16*(2), 97-101.

[31] Perković, T., Čagalj, M., & Rakić, N. (2010). SSSL: shoulder surfing safe login. *Journal of communications software and systems*, *6*(2), 65-73.

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

[33] Dunphy, P., Heiner, A. P., & Asokan, N. (2010, July). A closer look at recognition-based graphical passwords on mobile devices. In *Proceedings of the Sixth Symposium on Usable Privacy and Security* (pp. 1-12).

[34] Dunphy, P., Heiner, A. P., & Asokan, N. (2010, July). A closer look at recognition-based graphical passwords on mobile devices. In *Proceedings of the Sixth Symposium on Usable Privacy and Security* (pp. 1-12)..

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

[36] Srinivasan, R. (2018). DragPIN: A secured PIN entry scheme to avert attacks. Int. Arab J. Inf. Technol., 15(2), 213-223.

[37] Kwon, T., & Na, S. (2014). SwitchPIN: Securing smartphone PIN entry with switchable keypads. In 2014 IEEE International Conference on Consumer Electronics (ICCE) (pp. 23-24). IEEE

[38] Shi, P., Zhu, B., & Youssef, A. (2009). A rotary pin entry scheme resilient to shoulder-surfing. In 2009 International Conference for Internet Technology and Secured Transactions,(ICITST) (pp. 1-7). IEEE..

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

NOMENCLATURE

C	client device
dwd	dynamic password genereted
E	array of 54 random numbers
M	image vector constiting of 54 images ids
P	Position indicator numbers, random numbers placed on each key
pwd	password of user
S	authentication Server
T	token number, a number between 1 to 54 randomly generated by server
U	user
V	array of keys
Δ	image token, a picture identified using T and used for the password entry
☺	cartoon pictures placed on the keys