UTP: A Novel PIN Number Based User Authentication Scheme

Srinivasan Rajarajan and Ponnada Priyadarsini
School of Computing, SASTRA Deemed University, India

Abstract: *This paper proposes a Personal Identification Number (PIN) number based authentication scheme named User Transformed PIN (UTP). It introduces a simple cognitive process with which users may transform their PIN numbers into a dynamic one-time number. PIN numbers are widely used for the purpose of user authentication. They are entered directly and reused several times. This makes them vulnerable to many types of attacks. To overcome their drawbacks, One Time Password (OTPs) are combined with PIN numbers to form a stronger two-factor authentication. Though it is relatively difficult to attack OTPs, nevertheless OTPs are not foolproof to attacks. In our proposed work, we have devised a new scheme that withstands many of the common attacks on PIN numbers and OTPs. In our scheme, users will generate the UTP with the help of a visual pattern, random alphabets sequence and a PIN number. Because the UTP varies for each transaction, it acts like an OTP. Our scheme conceals PIN number within the UTP so that no direct entry of PIN number is required. The PIN number could be retrieved from the UTP by the authenticator module at the server. To the best our knowledge, this is the first scheme that facilitates users to transform their PIN numbers into a one-time number without any special device or tool. Our scheme is an inherently multi-factor authentication by combining knowledge factor and possession factor within itself. The user studies we conducted on the prototype have provided encouraging results to support the scheme's security and usability.*

Keywords: *Personal identification number, shoulder surfing, keylogging, user authentication, otp, internet banking.*

Received July 19, 2016; accepted June 4, 2017

1. Introduction

There is phenomenal surge in the volume of online financial transactions across the world. They are preferred due to their convenience. The advent of mobile phones and mobile banking has resulted in mobile banking gaining popularity. The major obstacle in the growth of Internet banking is the concerns in its security. Two factor authentications by employing static Personal Identification Number (PIN) numbers and dynamic One Time Password (OTPs) offer reasonable security [24]. But OTP attacks are becoming common nowadays [18].

An attacker has the option of unleashing his attack at one the three locations-client's system, communication medium and the server. While attacks happen everywhere, the client systems are targeted frequently since they are easy targets. The user systems are generally poorly equipped to tackle security attacks. Stealing the user credentials when the user is participating in authentication by feeding his user Id, password and PIN numbers is relatively easier due to the feeble security environment at the client machine [22].

Some of the types of attacks that are carried out from user systems are mentioned below:

- *Shoulder Surfing Attacks*: in a shoulder surfing attack, the attacker manages to observe the victim's data entries either manually or by using a recording device or with the help of any malware and thereby masquerading him to access victim's account [17, 19].
- *Key logging*: key logging is done through hardware or software. In this attack, the attacker stealthily inserts a malicious code called key-logger into the user's system. The key-logger records and forwards the keyboard entries of the user to an attacker.
- *Man in-the-Browser Attack*: this is a new form of attack unleashed through the browsers. There is numerous security vulnerabilities present in the current browsers [6]. These vulnerabilities are exploited by attackers.
- *Cross-site Scripting*: Cross-Site Scripting (XSS) are vulnerabilities that could be present in the web applications viewed by users. Attackers inject malicious client side scripts into them before they are loaded onto the user's browsers [7] Browsers fail to filter out the cross-site scripts since it is hard to predict the runtime behaviour of dynamic codes [20].
- *Phishing*: phishing is an attack to gather the credentials of the users by deceiving them to believe that they are asked to disclose those details by a trust-worthy entity. This is typically done by sending spoofed e-mails with forged identities of the real authorities [3].
- *Man-in-the-middle Attacks*: in a Man-in-the-Middle Attack (MITM) attack, an attacker intercepts the communication between two legitimate parties and acquires vital details. Sometimes the attacker might

even alter the communication but still make them think that they are directly communicating with each other.

- *Public WiFi Attack*: it is very common nowadays for people to access public WiFi at restaurants, airports, coffee shops, libraries etc., Sometimes the information passed from the device is unencrypted within the wifi. This introduces the risk of user credentials such as passwords and PINs get sneaked by attackers [5].
- *Smudge Attack*: these attacks are specific to touch screen based smart phones. When the user of smart phone uses the software keyboard to type in the password or draws a pattern to unlock the phone, it might leave smudges on the screen. This could be utilized by an attacker to reconstruct the password or the pattern [8].
- *Guessing and Dictionary Attacks*: the attacker attempts to predict the PIN number by making random guesses. But normally systems will not permit beyond 3 wrong entries of PIN numbers to guard against guessing attacks. In that case, an attacker has only 0.03% probability to guess the PIN number comprising of four digits. But under a dictionary attack the attacker selectively attempts possible numbers and have a better chance of succeeding [1].

1.1. Multi-Factor Authentication

User authentication schemes are broadly classified into three types. Knowledge factor is the first type that involves either a password or a PIN. Possession factor based authentication involves a physical proof to establish ones identity. It may be an Multimedia Messaging Service (ATM) card or a token. In the recent times, phone numbers and e-mail addresses are widely preferred since they are unique and are available with all users. An OTP is forwarded to the user during a transaction and users should to enter the same in the transaction page. While this alleviates the users from carrying any other additional security tokens like a card, the security will be compromised if the mobile phone is stolen or lost. Inherence factor is based on some physical attributes of a person which are unique and can not be impersonated. Bio-metric schemes that are based on finger print, iris, face etc. are in use. A multi-factor authentication is by combining more than one scheme of authentication to strengthen the authentication process and to increase its success rate [4].

1.2. OTP Issues and Attacks

Although OTPs are an effective way to mitigate attacks, they are vulnerable to certain types of attacks.Besides the possible attacks on OTPs, there is also an amount of inconvenience incurred by the users by since they have to wait for an OTP to be received on their mobile phone for every transaction. If the mobile phone is left at home by mistake or misplaced, if different mobile numbers are given for different bank accounts, when mobile phone usage is prohibited in an environment or it is a out of coverage area, if the mobile phone is unusable due to depleted battery etc. In all these occasions, the user can not successfully complete the transaction. Delay in delivery of OTPs is also very common reason for failed transactions.

2. Scope and Contributions

Considering the variety and the prevalence of attacks on Internet based transactions, we have designed a unique authentication scheme to withstand several of the listed attacks. The original contributions of our paper include:

Two variants of our proposed scheme for ensuring the verification of user's authentication credentials. First variant makes use of Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) and the second one uses encrypted Quick Response Code (QRCode) instead of CAPTCHA. Depending upon the required level of security, one of these schemes could be adopted for implementation.

- An implementation of our schemes to demonstrate their security and usability through analysis and user studies.
- Detailed analysis on the resistance of our scheme against various forms of security attacks.

Our proposed scheme is well suited for real world applications that demand utmost security in user authentications, especially those that involve financial transactions like e-banking, credit card, e-commerce etc. We have also ascertained the usability of our scheme through user evaluation studies.

3. Organization

The rest of the paper is organized as follows. In section 4 we have reviewed some of the related research works carried out already, section 5 discusses the system and attacker model, section 6 introduces the proposed schemes, user study is discussed in section 6 followed by the security analysis in section 7, a brief discussion comprising of strengths and limitations of the proposed scheme is carried out in section 8 section 9 presents the results and inferences of the user study, and section 10 concludes the paper. Table 1 consist of the expansions for the abbreviations used in this paper.

Table 1. Abbreviations used.

Symbol	Expansion
PIN	Personal Identification Number
OTP	One Time Password
XSS	Cross-Site Scripting
MITM	Man-in-the-Middle Attack
MMS	Multimedia Messaging Service
HSSA	Human Shoulder Surfing
RSSA	Recorded shoulder Surfing
UTP	User Transformed PIN
ATM	Multimedia Messaging Service
CAPTCHA	Completely Automated Public Turing Test to Tell Computers and Humans Apart
QR Code	Quick Response Code

4. Related Works

Several researches have been carried out to strengthen the security of user authentication based on PIN numbers. We classify the research works on securing user authentication into two categories-those that are aimed at developing secured PIN entry interfaces to withstand shoulder-surfing and keylogging attacks, and those that have attempted to improve the security by incorporating stronger OTP to resist Man-in-the-Middle, Man-in-the-browser and other OTP attacks. Since our proposed approach presents a unified solution integrating both the types of solutions, we felt that it is appropriate to discuss both categories of solutions from the literature.

A PIN entry method has been designed in [11] involving a row of digits and objects such as circle, rectangle etc. In order to feed the PIN digits, users have to position the same object that they noticed below their first PIN digit in the first round to be brought below the other PIN digits in each round. The objects could be moved clock wise and anti-clock wise with the help of the left and right arrows positioned beneath the two rows. While this scheme offers superior security against human-shoulder surfing attacks, it only provides minimal protection against recorded-shoulder surfing attacks. A colour based PIN entry keypad to avert shoulder-surfing attacks at ATM machines is proposed by [2]. Users should recognize their PIN digit and its associated colour. Then they should enter the alphabet displayed in that colour

A scheme based on colouring the PIN digits and allowing the users to choose the colour of their PIN digit is implemented under the immediate oracle choices [21]. In this scheme, the PIN entry is completed in m X n rounds where m is the number of digits in the PIN and n is the number of rounds for each digit. To extenuate the possibility of shoulder surfing in case the user is slow in completing the rounds and thereby giving ample opportunity for the observer to observe, the delayed oracle choice version is proposed. Although this improves defence against shoulder surfing, it could increase the number of wrong entries. Moreover this scheme offers no protection against the recorded-shoulder surfing attack.

An encrypted QRCode based PIN entry protocol is proposed in [15]. The QRCode that is encrypted by the server is displayed on the login page of the user. Users have to capture it using their mobile camera. The QRCode decoder which is present in the mobile phone decrypts and reveals the keyboard layout concealed inside the QRCode. Keeping that as reference, users have to enter their PIN numbers on the blank keyboard displayed on their system. For any observer who does not see the keyboard layout on the mobile phone, it is impossible to predict the actual PIN digits entered. But if the keyboard layout is seen by a potential attacker then it is very easy for him to trace the actual digits entered. This scheme is also feeble against the theft of mobile phone.

There are several research works that have developed schemes to utilize the benefits of two factor authentication by incorporating OTPs into the authentication process. In the scheme called stegnoPIN [10] two keypads are provided to the user. One is the regular numeric keypad and the other is a challenge keypad in which numbers are randomly organized. The scheme makes use of proximity sensor to secretly view the keypad to generate an OTP. Though this scheme prevents shoulder surfing attacks, it takes longer for to complete PIN entry process. Securing the credit card payments through OTPs is the objective of [13]. In this scheme credit cards are embedded with chips to perform hashing based on the previous OTP and a secret binary string. This is forwarded to the card issuer for verification. But this scheme requires a smart card reader to be available for performing the computation of OTP. In their paper [19], the authors have proposed a key transfer process. According to this scheme the keys in the keyboard will be shuffled after the user have noted down the position of the key with the PIN digit. Once the keys are shuffled, there will be a blank keyboard without any characters displayed. Now the user is suppose to click the key on the keyboard which is the expected position of the PIN digit after shuffling. This scheme defends against both Human Shoulder Surfing (HSSA) and Recorded Shoulder Surfing (RSSA)

A comparison of the existing schemes with our proposed scheme in terms of their security against different types of attacks is provided in Table 2. From the study we realized that most of the schemes employ single factor only. But our scheme provides an integrated solution by encompassing the PIN number with a phone number or e-mail address. Access to the registered phone number or e-mail address is needed to receive the visual pattern.

Table 2. Comparative security analysis.

	HSSA	RSSA	Key-logging	Multi-Factor
ColorPIN [2]	✓	Δ	✓	×
IOC[21]	✓	✓	×	×
ColoredPatterns [22]	✓	Δ	✓	×
RotaryPIN [23]	✓	×	✓	×
LIN [11]	✓	Δ	✓	×
KeyTransfer [19]	✓	✓	✓	Δ
SwitchPIN [9]	✓	×	✓	×
PassObjects [3]	✓	×	✓	×
Visual Auth. Protocols [15]	✓	✓	✓	Δ
STL [16]	✓	✓	✓	×
SteganoPIN [10]	✓	✓	✓	×
User Transformed PIN (UTP)[proposed in this paper]	✓	✓	✓	✓

HSSA-Human Shoulder Surfing Attack, RSSA-Recorded Shoulder Surfing Attack, MITM- Man-in-the-Middle Attack ✓-denotes the scheme's strong resistance against the attack, Δ -denotes partial resistance, ×-denotes poor or no defence.

5. System and Attacker Model

5.1. System Model

The system model of our system consists of four entities-client, server, user, a smart phone with QRCode decoder and an email account. The need for QRCode decoder is only required for the second variant of our scheme which makes use of encrypted QRCode. The alternate variant based on CAPTCHA does not require any QRCode app. An image pattern would have to be sent either to the registered mobile number as a MMS or as an email to the email address. For the purpose of conceiving the User Transformed PIN (UTP) with the help of the image pattern, alphabets sequence, CAPTCHA or QR Code, users are expected to have basic cognitive capability. We have ensured that the expected level of this capability of the user is kept within nominal range. The user study also ascertained the same.

5.2. CAPTCHA and QRCodes Variants

Our proposed system is implemented in two variants. The first variant makes use of a CAPTCHA. A CAPTCHA is a challenge-response test that is expected to be cleared only by human beings. This is used to discriminate humans from bots or computers to prevent automated attacks by computers [12]. CAPTCHAs are generally based on text reading or visual perceptions. But any hard AI problem could be made use of for developing a CAPTCHA [25]. Though CAPTCHAs alone are not foolproof to attacks, they could be included as one of security measures along with other protections.

In the second variant, the CAPTCHA is replaced by a Quick Response Code (QRCode). A QRCcode is a type of bar code that contains information about an item and it is machine readable. It was originally invented for automobile tracking. The advent of smart phones has resulted in a new usage of QR codes. In the recent times QR codes have got an encrypted version called encrypted QR code. Data Encryption Standard (DES) algorithm is used for the encryption. The keylogging resistant visual authentication scheme of [15] *et al.* is based on encrypted QR Codes.

5.3. Attacker Model

An attacker according to our system is an adversary attempting to make gains by exploiting the actual users by stealing their PIN number and impersonating them. They could employ any attacking strategy to achieve their mission. We classify the attacker's strategies into two categories as follows:

- Attacks that are unleashed from the user's system: Some of those attacks are shoulder surfing, keylogging, Man-in-the-browser and mobile malwares.
- Attacks that are triggered from remote locations: Man-in-the-middle, replay attacks, phishing, password guessing, dictionary attacks, brute-force attacks and denial of service attacks are examples of such attacks.

An intelligent attacker might employ multiple strategies to increase his chances of a successful attack.

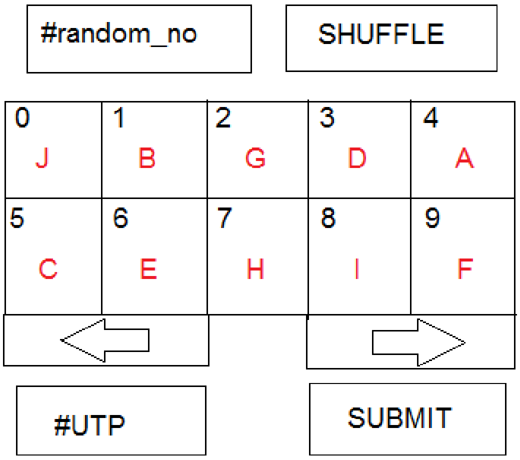
6. Proposed System

The proposed system makes use of three components that allow the users to construct and enter a UTP. They are as follows:

1. Static or fixed PIN number.
2. Visual pattern.
3. Random alphabets sequence.

6.1. Model of the PIN Entry Interface

The model of our proposed scheme's PIN entry interface is shown in Figure 1. It comprise of a input box to enter the random number sent by the server, a button to initiate the alphabets shuffle operation based on the random number entered, a set of ten keys with a combination of a digit and an alphabet in each key. There are two arrows- backward and forward. At the bottom, there is an input box to enter the UTP number that the user conceives and a submit button to forward the number for verification at the server.



0 J	1 B	2 G	3 D	4 A
5 C	6 E	7 H	8 I	9 F

Figure 1. Model of the proposed PIN entry interface.

6.2. Static PIN Number

A PIN is a four digit secret number assigned by the bank and is communicated to the customers at the time of opening the account. Customers may also choose a number of their choice as their PIN number. This number is either permanent or replaced periodically. Since four digits are commonly used for PIN numbers, we have assumed the same for our evaluations. But our system would easily scale for PIN numbers of any digits.

6.3. Visual Pattern

This is a crucial element of our scheme. It is made up of ten alphabets from A to J. Each alphabet is assigned a number from 1 to 8. The number for an alphabet has been assigned randomly. This visual pattern is generated at the server and is forwarded to the customer as an image through email or MMS or WhatsApp. Possession of this image is essential for users to complete their UTP entry. Users may keep this pattern in their mobile phone or they may keep a printed copy of this with them. Because this pattern alone is not adequate to break into the user's account without knowing the PIN number, accidental leak or loss of the pattern will not lead to security compromise.

But user should report to the bank and obtain a new copy of the visual pattern to avoid any chance of attack in future.

This visual pattern need not be newly generated each time when the user wants to do a transaction. The same visual pattern will result in different UTP numbers when it is combined with the random alphabets sequence and the PIN. This alleviates the hindrance of waiting for a new pattern every time. But for enhanced security, the pattern could be replaced after a specified number of transactions or time period. A sample visual pattern is presented in Figure 2. It shows 10 alphabets from A to J being assigned numbers from 0 to 9 randomly. These numbers will be used to substitute the PIN digits while forming the UTP number.

A - 1	B - 7	C - 6	D - 4	E - 0
F - 2	G - 5	H - 3	I - 9	J - 8

Figure 2. Visual pattern.

6.4. Random Alphabets Sequence

This is displayed on the page where the user is making the PIN entry. When the page is loaded, the same ten alphabets present in the visual pattern are displayed here. But the alphabets are arranged in a random order. The random order is determined by the server based on an algorithm and it changes every time. There are 4 four different groups of ten alphabets under G_1, G_2, G_3 and G_4. Though the alphabets are same, they are in different orders in each group. Figure 3 shows the sample alphabets sequence of a group. The ten alphabets shown initially are of group G_1 and are to be used for finding the first digit of the UTP. For conceiving the next three UTP digits, users should load the next groups of alphabets under G_2, G_3 and G_4 by simply press the right arrow button(->) displayed in the interface. This would cause the next alphabets group to get loaded on the screen replacing previous group.

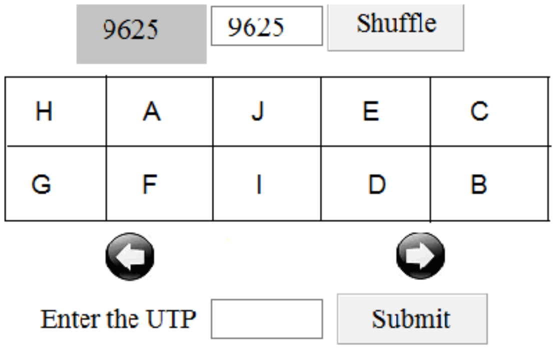
Besides the scrambling of alphabets done at the server, there is another level of randomization to be carried out by the user by shifting the alphabets. To accomplish this, user should enter the random number that is displayed in the form of a CAPTCHA or an encrypted QRCode. This causes the ten alphabets to get shifted based on the entered number. Because the number displayed is randomly chosen for each transaction, the arrangement of the alphabets will also vary each time.

The shifting is done by left shifting the alphabets based on the entered number. For example, if the random number is 4321, then the alphabet sequence S_1 for selecting the first digit of the UTP will be left shifted by four positions, alphabet sequence for the second UTP digit gets shifted by three positions, third alphabet sequence gets shifted by two positions an the last shifted by one position. For any reason if user wants to re-enter or verify the UTP digits they had already entered, the left arrow key will help them to scroll back to view the previous set of alphabets.

Each alphabet also carries a numeric digit in its top left corner. These numbers are displayed in ascending order from 0 to 9. They remain fixed and do not get randomized along with the alphabets.

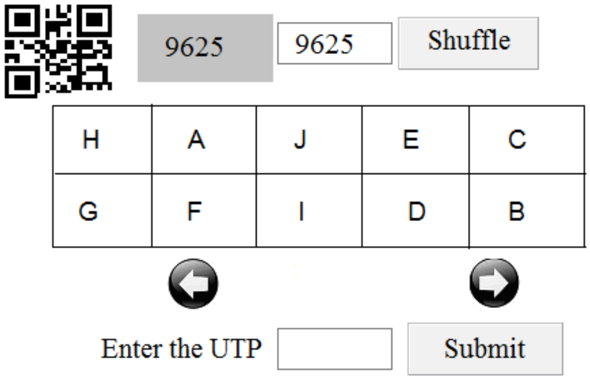
The following Figure 3 contains the screen shots of the implementation of our proposed scheme in Asp.net. The first interface is a CAPTCHA variant which simply shows the random number '9625' in a

CAPTCHA form. The second interface in Figure 4 is a QRCode variant. In this the number is played as a QRCode. It needs to be captured and decrypted using the mobile phone to reveal the random number. This is only possible if the user knows the correct password for decryption. The password could be made of the PIN number itself or it could be a separate password value.



H	A	J	E	C
G	F	I	D	B

Figure 3. Random Alphabets Pattern for first digit of UTP selection with random number displayed in CAPTCHA and up and down arrow buttons.



H	A	J	E	C
G	F	I	D	B

Figure 4. QRCode variant with random number concealed in a encrypted QRCode.

6.5. Modules Used in the Scheme

In this section, the two variants of the proposed scheme based on CAPTCHA and QR Code are described. Our scheme makes use of the following functions:

- *Visual Pattern Generation*: we have developed an algorithm for the generation of the visual pattern. The mechanism of this algorithm is to randomize the numbers from 0 to 9 and associate each number with the ten alphabets.
- *Random Number Generation*: this algorithm would generate a four digit number each time. The generation of the random number is independent of the numbers generated previously and subsequently.
- *Random Alphabets Sequence Generation*: based on the random number typed in by the user, a shift algorithm is applied to shuffle the alphabets of each PIN digit position.
- *UTP Verification*: this is the algorithm executed by the server after receiving the UTP number from the user. This is done in order to reproduce the actual PIN number of the user.

6.6. UTP Generation at the Client

This is the core logic of the proposed scheme. With the help of the visual pattern, alphabets sequence and the PIN number, user develops the UTP. This is the algorithm that enables the verification code at the server to decipher the PIN number from the UTP to authenticate the user. The algorithm for UTP generation is provided below (Algorithm 1).

Algorithm 1: Algorithm for UTP generation

First input that is to be entered by the user is the 4 digit random number that is displayed on the transaction page. Let the digits of this number from highest to lowest order be d_1, d_2, d_3 and d_4.
There are four groups of alphabets in the random alphabets sequence. Each group comprise of the same ten alphabets from 'a' to 'j', but are in a different random order in each group of G1, G_2, G_3, and G_4.
Left shifting of the alphabets under G_1, G_2, G_3, and G_4 is done by the system for d_1, d_2, d_3 and d_4 number of times respectively.
In the second step, user recalls his first PIN digit p_1. Then he identifies the alphabet in l_i of G_1 that contains the number p_1 at its top left corner. Let it be β.
Then find out the number R_1 equivalent to the alphabet β from the visual pattern and consider this as the first digit of the UTP.
Press the right arrow button to load the alphabets belonging to group G_2.
Repeat step 4 to 6 with the PIN digits of p_2, p_3 and p_4 by using the alphabets in the lists of G_2, G_3 and G_4 to obtain the, R_2, R_3 and R_4 digits of the UTP.
Finally submit the UTP number composed of $R_1R_2R_3R_4$.

6.7. UTP Verification at the Server

Once the UTP number is received by the server, it should invoke the verification logic to check whether the entered number contains the actual PIN number assigned to the user. For that the PIN number has to be uncovered from the UTP number. The following algorithm is used for the UTP verification. (Algorithm 2).

Algorithm 2 : Algorithm for UTP verification

Retrieve the user's PIN number comprising of $\{P_1,P_2,P_3,P_4\}$ that is stored in the database and the visual pattern array pertaining to that user.
Using the random number that was passed to the user in a CAPTCHA or QRCode form, invoke the Shifting module on the array of random alphabets of R sent to the user previously.
For the four digits of UTP $\{D_1,D_2,D_3,D_4\}$, the following step will be followed:
Take the first digit D_1. Retrieve the alphabet A_1 equivalent to D_1 from the visual pattern array VIS, by using D_1 as an index into VIS as $A_{1=}$ VIS[d_1].
Find the column number C_1 from the first row of random alphabets array that contains A_1. by doing a comparison;- if RAN[1,i] is equal to A_1 then C_1=i else increment i.
Repeat steps iv to v to obtain C_2, C_3 and C_4.
If C_1= $P_1$and $C_{2=}$ P_2 and C_3= P_3 and C_4= P_4 then authentication is successful, otherwise user failed in authentication.

6.8. Procedure for UTP Entry

The steps to be followed by the user to generate the UTP and to complete the authentication are given below:

1. After logging into the account with the help of the user Id and password, a random alphabets pattern along with a CAPTCHA or QR Code will be displayed on the PIN entry page.
2. If it is a CAPTCHA variant, user should read the number from the CAPTCHA and enter it into the adjacent textbox.
3. In case if it is a QR Code variant, then a QR Code appears on the page. Users should capture the QR Code using their mobile phone's camera through the QR Code app. As soon as the QRCode is captured, users will be prompted to enter the password for decryption. Then the QR Code decoder will display the actual number to be entered.
4. Now they should press the Shuffle button. This will cause the alphabets in the random alphabets pattern to be rearranged according to the shifting algorithm described earlier.
5. The first set of ten alphabets to be used for computing the first digit of the UTP is displayed now. User should open the visual pattern image and keep it ready to be used for the UTP generation.
6. Recalling the first digit of their PIN number, users should find out the alphabet displayed with that number on the random alphabets pattern. Then user should glance at the visual pattern to identify the number assigned for that alphabet. That number becomes the first digit of the UTP. For example, assume that the user's PIN number is 5263 and the alphabet with the 5 displayed in its top is is 'e'. Then user should look for the number assigned to 'e' in his visual pattern. In case the number for 'e' is 7, then 7 becomes the first UTP digit equivalent to the first PIN digit 5.
7. In the next iterations, users would press the down arrow to load the next row of alphabets. Repeating the previous process, they will be able to identify the UTP digits equivalent to their PIN digits.
8. Now four digits of UTP representing the four digits of the fixed PIN are formed by the user.
9. User will have to submit the UTP for verification.
10. At the server, the UTP algorithm is run with the inputs of stored PIN number of the user and the user entered UTP. If there is a match, user is authenticated positively, otherwise declined.

For example, consider the sample visual pattern present in Figure 2 and the sample random alphabets sequence given in Figure 3. It is already left shifted 9 times based on the first digit of the random number 9625. If we assume 2567 as the secret PIN number of the user, then the first digit of the UTP to be entered is '8'. This is because the alphabet that contains '2' in the random alphabets sequence is 'J' and the number representing alphabet 'J' in the visual pattern is '8'. Similarly user should find out the replacement numbers for 5, 6, and 7 to form the four digit UTP number.

7. Security Analysis

In this section we are analyzing the resistance of the scheme under various attack scenarios.

7.1. Guessing and Dictionary Attacks

Even if the PIN number is chosen vulnerably, attacks are annulled because of the transformation of PIN number into an UTP. The system will not accept the actual PIN number if entered directly. It requires to be transformed into an UTP with the help of the appropriate visual pattern of the user.

7.2. Shoulder Surfing Attacks

A human SSA is prevented in our scheme because the user does not enter the PIN number directly but he enters only the UTP. Even if the attacker is able to peep into the secret visual pattern, it is not possible to reconstruct the PIN number with the help of it. Besides that the UTP entry takes place in a password entry field which only displays stars for each entry.

In the case of recorded SSA, the attacker manages to record the on-screen PIN entry process with a camera or a malware in the system that records screen activity. But even if the attacker manages to record the UTP entry process, as long as the visual pattern is not available to him, he can not attack the account.. If the scheme is implemented based on the encrypted QRCode version, then the attacker would also acquire password for decrypting the random number to compete his attack.

7.3. Keylogging Attack

Even though only the physical keyboards are susceptible to keylogging attacks and software keyboards are immune to them, the usage of third-party keyboard apps for android based smart phones have introduced the possibility of attacks into the software keyboards too. Our proposed scheme offers maximum amount of security against keylogging attacks. Because the user only enters the UTP number which is only valid for one transaction, any keylogging attack will be futile. So users may use any keyboard without being concerned about keylogging attacks.

7.4. Man-in-the-Middle Attacks

Any kind of attack that captures the PIN entry being transmitted will be nullified in per our proposed scheme. The UTP number that is being transmitted from the client to the server is of no use after that transaction or after the transaction period expires. It is also impossible for the attacker to reveal the inherent PIN number in the UTP number without obtaining the random number, random alphabets sequence and the visual pattern

7.5. Browser and Cross Side Scripting Attacks

Since browsers could only record the entries of the users, they could only record the UTP number, not the actual PIN number that is not typed anywhere. CSS attacks are also defied due to the same reason.

7.6. OTP Attacks

In a typical OTP scheme, the OTP generation takes place at the server and it is then forwarded to the client's system. In this scenario, an OTP attack could happen at three possible locations- at the server, on the transmission medium and at the destination device. Among the three possibilities, the attack on the transmission medium that is called as replay attack is more prevalent [14]. In the proposed scheme, the UTP generation carried out at the client system which involves the participation of the user along with help of the patterns and PIN provided by the server. An advantage of our scheme is that the there is no connection between the subsequent UTP numbers. So predicting the UTP number based on the previously entered entries is nullified.

8. Discussion

Our scheme is an inherently multi-factor authentication scheme utilizing the user's knowledge of the PIN number, the visual pattern received in e-mail address or mobile number and the QRCode scanner with correct password to reveal the random number. It is impossible to breach the security of our scheme either by stealing the PIN number alone or by obtaining the visual pattern alone, or just by knowing the password of the QRCOde.

Since the PIN number is transformed into an UTP, it is not necessary to periodically change the PIN number. Frequently changing the PIN number and remembering the new PIN number is an inconvenience to users. But it is still possible to change the PIN number if required. Since the PIN number and the visual pattern generation are independent, changing any one does not invalidate or affect the other. Our scheme puts some additional overhead on the user to enter their PIN numbers and it could also result in slightly longer PIN entry time. But besides the elevated securities offered by our scheme, users also gain the following conveniences through our scheme:

1. No need to change the PIN numbers frequently to maintain security.
2. No need to wait for receiving the OTP each time while doing a transaction.
3. PIN numbers stolen will not lead to impersonation attacks on the user account.
4. No need to fear for using the public computer systems or for using the public wifi facilities or to use systems that can not be fully trusted for doing any Internet purchases or payments using the PIN number.

9. Usability Study

An important of goal of any user authentication scheme is the usability level of the proposed scheme. It is apparent that our scheme has introduced some amount complexity into the PIN entry process in comparison to the plain unsecured PIN entry mechanism. But we are attempting establish that the complexity level is meagre comparing to many of the proposed schemes and it is acceptable considering the significant security improvement it offers.

9.1. User Study

To conduct a hands on analysis of our proposed scheme with the help of real users, we implemented a model of the CAPTCHA variant scheme with the all the functionalities. We developed the web version our scheme using .Net. The next requirement was to rope in people to act as analysts of our scheme. For better accuracy of test results, we decided to include people representing different segments of people under different age groups and both genders. So we identified a total 35 participants having a mix of all.

We first conducted an introductory session explaining them the concept and the methodology of our scheme. We collected their e-mail addresses to forward the visual patterns and PIN numbers. The participants were asked to attempt the authentication with the help of the visual pattern and the PIN number they received at their convenient time and place. Their entries were recorded in the server for us to analyze latter.

The recording included the success/failure of the authentication and the time taken for completing the process. We carried out this test for a period of 15 days with the members trying the authentication 3 times at random gaps. The results of the study are presented in Figure 5. Through the study we ascertained that the proposed scheme is usable and the results indicate that PIN entry time is within tolerable range. We also found that the PIN entry time decreases in repeated usage. For more comprehensive assessment of usability we will have to carry out rigorous user studies with a larger user population.

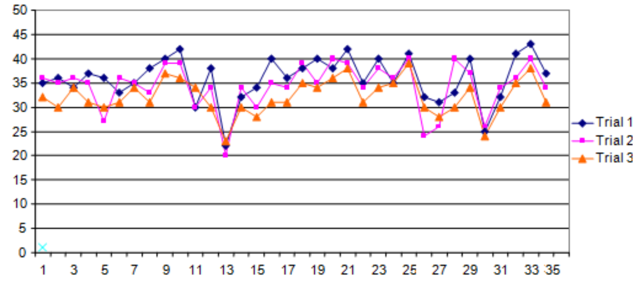


Figure 5. User study results of 35 participants based on three trials.

10. Conclusions

In this paper we proposed an authentication scheme that makes users to transform their PIN numbers into a one-time valid UTP numbers. The proposed scheme overcomes the drawbacks of the previous schemes and offers superior security and usability. The scheme makes use of an image token called visual pattern. By keeping this as a reference users could generate the UTPs. Although our scheme employs a user cognitive process, it is within the capacity of average human beings. Our user studies have also indicated that the usability of the scheme improves by practice and repeated usage. Our proposed scheme is likely to open a new direction for several researches in this domain. We plan to explore the possibility of developing a mobile version of our proposed scheme and to customize the scheme to suit the characteristics of a mobile phone.

References

[1] Bonneau J., Preibusch S., and Anderson R., "A Birthday Present Every Eleven Wallets? The Security of Customer-Chosen Banking PINs," *in Proceedings of International Conference on Financial Cryptography and Data Security*, Kralendijk, pp. 25-40, 2012.

[2] De Luca A., Hertzschuch K., and Hussmann H., "ColorPIN: Securing PIN Entry Through Indirect Input," *in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Atlanta, pp. 1103-1106, 2010.

[3] Dhamija R., Tygar J., and Hearst M., "Why phishing works," *in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Montréal, pp. 581-590, 2006.

[4] Dmitrienko A., Liebchen C., Rossow C., and Sadeghi A., "On the (in) Security of Mobile Two-Factor Authentication," *in Proceedings of International Conference on Financial Cryptography and Data Security*, Christ Church, pp. 365-383, 2014.

[5] Fraser E., "The Failure of Public Wifi," *Journal of Technology Law and Policy*, vol. 14, no. 2, pp. 161, 2009.

[6] Grier C., Tang S., and King S., "Secure Web Browsing with the OP Web Browser," *IEEE Symposium on Security and Privacy*, Oakland, pp. 402-416, 2008.

[7] Jim T., Swamy N., and Hicks M., "Defeating Script Injection Attacks with Browser-Enforced Embedded Policies," *in Proceedings of the 16th International Conference on World Wide Web*, Banff, pp. 601-610, 2007.

[8] Kwon T. and Na S., "TinyLock: Affordable Defense against Smudge Attacks on Smartphone Pattern Lock Systems," *Computers and Security*, vol. 42, pp. 137-150, 2014.

[9] Kwon T. and Na S., "SwitchPIN: Securing Smartphone PIN Entry with Switchable Keypads," *in Proceedings of IEEE International Conference on Consumer Electronics*, Las Vegas, pp. 23-24, 2014.

[10] Kwon T. and Na S., "SteganoPIN: Two-Faced Human-Machine Interface for Practical Enforcement of PIN Entry Security," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 143-150, 2016.

[11] Lee M., "Security Notions and Advanced Method for Human Shoulder-Surfing Resistant PIN-Entry," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 695-708, 2014.

[12] Leung C., "Visual Security is Feeble for Anti-Phishing," *in Proceedings of 3rd International Conference on Anti-Counterfeiting, Security, and Identification in Communication*, Hong Kong, pp. 118-123, 2009.

[13] Li Y. and Zhang S., "Securing Credit Card Transactions with One-Time Payment Scheme," *Electronic Commerce Research and Applications* vol. 4, no. 4, pp. 413-426, 2006.

[14] Mulliner C., Borgaonkar R., Stewin P., and Seifert J., "SMS-based one-Time Passwords: Attacks and Defense," *in Proceedings of International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Berlin, pp. 150-159, 2013.

[15] Nyang D., Mohaisen A., and Kang J., "Keylogging-Resistant Visual Authentication Protocols," *IEEE Transactions on Mobile Computing*, vol. 13, no. 11, pp. 2566-2579, 2014.

[16] Perković T., Čagalj M., and Saxena N., "Shoulder-Surfing Safe Login in A Partially Observable Attacker Model," *in Proceedings of International Conference on Financial Cryptography and Data Security*, Tenerife, pp. 351-358, 2010.

[17] Por L., "Frequency of Occurrence Analysis Attack and its Countermeasure," *The International Arab Journal of Information Technology*, vol. 10, no. 2, pp. 189-197, 2013.

[18] Raddum H., Nestås L., and Hole K., "Security Analysis of Mobile Phones Used As OTP Generators," *in Proceedings of Information*

Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices, Passau, pp. 324-331, 2010.

[19] Rajarajan S., Maheswari K., Hemapriya R., and Sriharilakshmi S., "Shoulder Surfing Resistant Virtual Keyboard for Internet Banking," *World Applied Sciences Journal*, vol. 31, no. 7, pp. 1297-304, 2014.

[20] Reis C., Dunagan J., Wang H., Dubrovsky O., and Esmeir S., "BrowserShield: Vulnerability-Driven Filtering of Dynamic HTML," *ACM Transactions on the Web*, vol. 1, no. 3, pp. 11, 2007.

[21] Roth V., Richter K., and Freidinger R., "A PIN-Entry Method Resilient Against Shoulder Surfing," *in Proceedings of the 11th ACM Conference on Computer and Communications Security*, Washington, pp. 236-245, 2004.

[22] Shi P., Zhu B., and Youssef A., "A PIN Entry Scheme Resistant to Recording-Based Shoulder-Surfing, " *in Proceedings of the 3rd International Conference on Emerging Security Information, Systems and Technologies*, Athens, pp. 237-241, 2009.

[23] Shi P., Zhu B., and Youssef A., "A Rotary PIN Entry Scheme Resilient to Shoulder-Surfing, " *in Proceedings of the International Conference for Internet Technology and Secured Transactions*, London, pp. 1-7, 2009.

[24] Vaidya B., Park J., Yeo S., and Rodrigues J., "Robust One-Time Password Authentication Scheme Using Smart Card for Home Network Environment," *Computer Communications*, vol. 34, no. 3, pp. 326-336, 2011.

[25] Von Ahn L., Blum M., Hopper N., and Langford J., "CAPTCHA: Using Hard AI Problems for Security," *in Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, Warsaw, pp. 294-311, 2003.

Srinivasan Rajarajan received his M.Tech(CSE) from SASTRA University, Thanjavur in 2006. He is presently pursuing PhD at SASTRA University, India. He is an Assistant Professor at the same university. His areas of research interest include computer security, user authentication, e-banking and graphical passwords.

Ponnada Priyadarsini received her Ph.D from NIT, Trichy in the year 2009. Her research areas include algorithms, computational complexity, information security and graph theory.