Color wheel pin: Usable and resilient ATM authentication

Article *in* Journal of High Speed Networks · June 2016

DOI: 10.3233/JHS-160545

CITATIONS
4

READS
166

3 authors:

Meriem Guerar
Università degli Studi di Genova
13 PUBLICATIONS **44** CITATIONS
SEE PROFILE

Benmohammed Mohamed
Université Constantine 2
214 PUBLICATIONS **610** CITATIONS
SEE PROFILE

Vincent Alimi
National Graduate School of Engineering and Research Center (Caen)
18 PUBLICATIONS **75** CITATIONS
SEE PROFILE

Color Wheel Pin: Usable and Resilient ATM Authentication

Meriem Guerar*
Univ. Oran Mohamed Boudiaf
Oran, Algeria
meriem.guerar@univ-usto.dz

Mohamed Benmohammed
University of Constantine
Constantine, Algeria
benmoh123@yahoo.com

Vincent Alimi
525 Gaspe Street, Verdun
QC H3E 1E7,Canada
vincent.alimi@gmail.com

Abstract

We are witnessing a growing demand for ATM authentication solutions that overcome the limitations of the *de facto standard* mechanism based on magnetic card and numeric PIN, that has revealed to be weak against ATM-specific attacks (e.g. skimming and recording attacks). An emerging trend is relying on smartphones as a carrier for authentication. However, authentication mechanisms based on the use of a smartphone requires the same mechanisms to be resilient to new, smartphone-specific threats like device theft and common attacks like shoulder-surfing attacks and spyware. In this paper, we propose a new ATM authentication mechanism called Color Wheel Pin which combines a usable ATM authentication mechanism with robustness against smartphone and ATM specific and common security threats.

Keywords: ATM, Authentication, Mobile Security.

1 Introduction

In October 2015, the European ATM Security Team [1] reported that global card-skimming losses, which accounted for 131 million euros (U.S. $149 million) of the 156 million euros (U.S. $177.5 million) of ATM-related fraud losses reported for the first half of 2015, increased 18% during the first six months of 2015 when compared to the same period in 2014.

While some of these frauds involve physically compromised ATM machinery, as in the case of skimming attacks that steal customers' credentials, or software compromised ATM machinery, as in the case of jackpotting attacks that force the ATM to freely dispense all the money available, others simply involve the observation of users (shoulder-surfing attacks) to be able to replay their logging into the system. The banking system is currently pushing for a general upgrade of personal cards to a new standard that obsoletes the magnetic band in favor of electronic chips according to the EMV standard[2].

However, this upgrade will not solve the problem of skimming attacks as the new, very thin attacking devices are capable of intercepting the communication between the card and the ATM terminal. To solve this problem, we propose to enhance the authentication schemes integrating into it a user personal device such as a smartphone.

In our scheme, the whole process achieves resiliency to many types of attack such as skimming, recording, spyware injection and shoulder surfing. At the same time, the authentication process remains intuitive and short, thus highly usable. To our knowledge this is the first scheme that achieves this level of resilience and usability at the same time.

*Corresponding author

[1] https://www.european-atm-security.eu/files/European-ATM-Fraud-Incidents-up-15.pdf

[2] https://www.emvco.com/

The paper is structured as follows, in section 2 we provide a description of past work in this field; in section 3 we describe our scheme and the threat model it tackles; in section 4 we perform a security analysis of our scheme to prove its resilience to well-known attacks; in section 5 we provide the results of our usability experiments and, finally, in section 6 we provide some concluding remarks.

2 Related Work

In past work, many authentication mechanisms have been proposed to secure the interaction with public terminals, and most of them focus on ATM machines. These methods can be roughly divided in three categories according to the classification suggested by De Luca et al. [1]: software-based, hardware-based and user hardware-based approaches.

2.1 Software-based Authentication

Software-based authentications systems try to improve the security of user's input on a software level, without requiring any additional hardware. In 2004, Roth et al. [2] presented a PIN entry method which uses an obfuscation technique called *cognitive trapdoor game*. In their approach, the keypad is divided in two distinct colored sets, black and white. The user has to input the PIN digits indirectly by pressing a separate black or white key depending on which set it appears in. To identify one PIN digit, four rounds are required. This means that inserting a four digit PIN requires to press 16 buttons. Due to the created overhead, this system takes about ten times as long to enter standard PINs on a keypad. Additionally, recent research [3] showed that this system is vulnerable to a shoulder-surfing attack based on cognitive strategies and training, even in the absence of recording devices.

A similar approach is ColorPIN, proposed by De Luca et al. [4]. In this scheme, the user's pin is a sequence of four colored digits, where the color is selected among white, red, and black. To log in, the ATM machine displays three letters with different colors under each keypad digit. The user is asked to look at the bottom of each PIN digit and enter the letter corresponding to the PIN color. Since the letters are randomly generated and reassigned after each input, an observer fails to recognize the password through shoulder-surfing attack. However, an attacker can discover the original password by intersecting two sets of information acquired through recording attacks.

Other work that tries to protect password entry against shoulder surfing attack is the *spy-resistant keyboard* by Tan et al. [5]. Their approach is based on the indirection and on the fact that the capacity of short-term memory of a human is limited. To authenticate, the users have to memorize a particular part of the keyboard. Then, they are asked to do the selection in blank keyboard. Thus, an attacker has to memorize the entire keyboard to be able to reveal the typed character, which is difficult, if not impossible. In contrast, this approach increases the time and the complexity of the input. In addition, it is not resilient against observation attacks based on camera recordings.

Finally, Lee et al. [6] have presented a new PIN entry method that uses a random mapping between the PIN digits and alphabet characters as a challenge. The user has to memorize the characters associated with his/her PIN digits. Then, she has to enter the characters sequence after the corresponding challenge disappears from the screen. In this way, the attackers cannot observe the challenge and the response at the same time. In addition, they cannot completely memorize the instant mapping shown in the challenge. Hence, this approach protects the PIN entry method against shoulder surfing attack. However, similar to most software-based authentication mechanisms, this system is not resistant to camera recordings.

The main advantage of the proposals in this category is that they do not require any additional hardware and simple software updates are sufficient to improve their robustness. Unfortunately, the additional overhead and complexity used to secure the input lead to an increase in the authentication time and in

the error rate. In addition, they are susceptible to recording attacks.

2.2 Hardware-based Authentication

Another way to enhance the security of user's authentication in public terminals is to add an additional hardware that provides an invisible communication channel to transmit secret information. *Undercover* by Sasamoto et al. [7] uses a haptic device to secretly communicate to the users a keyboard layout that, in order to provide the correct answer, they have to combine with the graphical password displayed on the screen. Hiding a part of the challenge through haptic feedback increases the resistant against simple observation. However, it was shown in [8] that the system can be broken with high probability after the observation of about 10 authentication sessions, through timing or intersection attacks. In addition, this approach trades usability for security, taking an average login time from 35 to 45 seconds and error rates from 26 to 52 percent [9].

Bianchi et al. [10] designed a special keypad called *Secure Haptic Keypad* (SHK), which contains three hardware keys. In this system, the user's password is encoded as a sequence of tactons (i.e vibration patterns). To authenticate, the user has to explore with his fingertips which key between them produce the tacton corresponding to the password item and press its relevant button. After each input, the tactons are randomized over the three keys to make it hard the detection from shoulder-surfers. As pointed out in [9], this system has scalability and expressiveness problems. Bianchi et al. later developed the haptic wheel [11], which extend the recognition-based paradigm introduced in SHK to address the scalability problem. Both approaches are based on recognizing and selecting a sequence of tactons from a stimulus set. Since no visual feedback is provided, these systems are resistant to visual observation attacks. However, if the stimulus set exceed 3 or 4 items, it becomes hard for users to accurately select the password items [12].

Kumar et al. [13] involved an eye tracker to implement a gaze-based password entry. To log in, users have to select the password characters from an onscreen keyboard through the orientation of their pupils. Although this approach increases the resistance against observation attacks, it requires expensive eye-tracking equipment. Additionally, attackers could capture user passwords by using a rogue eye-tracker hidden close to the authentication terminal [7]. Other work in this field has been performed by De Luca et al. [14], which use eye gestures as passwords instead of on-screen character selection.

The main barriers to the widespread adoption of these systems are the hardware deployment costs and hardware manipulations that can be carried out by attackers once their characteristics are publicly available.

2.3 User Hardware-based Authentication

In contrast to hardware-based authentication, systems in this category require additional hardware owned by the users. A typical example of such hardware is the users' mobile phone and use the cellular network. As such systems are not a part of the public infrastructure, attackers are unable to manipulate them or perform skimming attacks. However, this requires both the mobile phone OS and the cellular network to be reistant against external attackers (see e.g. [15, 16, 17]).

De Luca et al. [18] introduced VibraPass, a system that use the vibration of user's mobile phone as an invisible communication channel to tell the user whenever she should enter a fake character to his PIN. As a shoulder-surfer cannot detect the vibrations, he cannot distinguish between correct and fake characters. However, repeated observations would reveal the password.

Recently Khan et al. [19] proposed SEPIA, an obfuscated PIN authentication for ATM using cloud-connected personal mobile or wearable devices. In this approach, the user has to scan a QR code from the screen of an ATM machine and connects to the cloud based bank's SEPIA server to obtain one-time PIN templates. This PIN template will be displayed on the mobile screen or the wearable device. To

authenticate, the user is required to input his PIN code obfuscated within the PIN template on the ATM machine. Similar to VibraPass, repeated observations can lead to successful attacks by analyzing the differences between inputs. In addition, unlike wearable devices, using the mobile device makes this approach vulnerable to shoulder-surfing attack.

Nyang et al. [20] presented a visual authentication protocol resilient to keylogger attacks. In this scheme, a QR code along with a blank keyboard are displayed on the terminal screen. To authenticate, the user has to scan the QR code through his smartphone to obtain a random permutation of a keyboard arrangement. Then, she has to type in the password according to this arrangement on the terminal's blank keyboard. All the transmitted data between terminal and user's smartphone are encrypted, thereby providing a secure communication channel. However, this scheme is vulnerable to shoulder-surfing attack.

Unlike previously mentioned authentication methods that aim at enhancing the security in ATM terminals, Guerar el al. [21] proposed a Completely Automatic Public Physical test to tell Computers and Humans Apart (CAPPCHA) to enhance the security in mobile devices. In their scheme, the user is asked to tilt the device to a specific degree displayed on the screen and hold it still in this position for one second to have access to the PIN pad. Their security analysis shows that the proposed scheme is resilient against brute force attacks, side channel attacks and spyware-based recording attacks.

Seyed et al. introduced CipherCard[3], a physical token that should be placed over a touchscreen PIN-pad during the authentication process in order to translate the user's password into a different system password received through a touchscreen. This way, it hides the system password from both shoulder-surfing and recording attacks. However, this authentication method does not provide any protection against keylogger attack. In addition, it does not use devices already owned by the user, which may create an additional deployment cost.

3 System and Threat Model

Our system model depends on three entities, which are the user, the ATM terminal and the Bank server.

- **User**: The user is an ordinary human with limited calculation abilities and limited memory. We assume that the user owns a smartphone or other personal wearable devices with a camera or NFC technology such as smartwatch or Google Glass, along with four-PIN digits for authenticating with the ATM terminal. The user's smartphone can be also used as a credit/debit card. Otherwise, a credit/debit card is required.

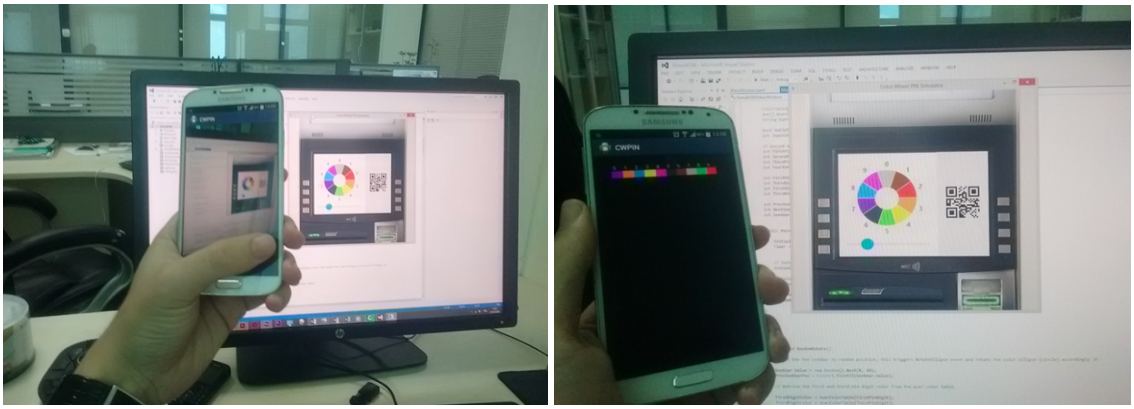
- **ATM terminal**: The ATM terminal employs touch screen technology and it may incorporate NFC technology. We assume that the communication channel between the server and the ATM terminal is secure (e.g with SSL protocol). We assume that ATM terminals are equipped with NFC reader, in case users own NFC-enabled smartphones.

- **Bank server**: This entity belongs to the financial institution and we assume that the user's credentials (i.e PIN and Table of Color) are stored securely on its database and it can communicate with the ATM terminal over secure connection.

3.1 Threat model

In this paper, we only consider security issues related to the user interaction with ATM. The security mechanisms used to make the server immune to attacks and to ensure secure communication channel

[3] `http://dspace.ucalgary.ca/bitstream/1880/50246/1/2014-1063-14.pdf`



(a) QR Code Scanning (b) Table of colors in a smartphone

Figure 1: Photographs of the prototype we have developed to demonstrate our authentication protocols. (a) Show the moments of a QR code scanning. (b) Show the table of color on smartphone obtained from the QR code.

between the server and the ATM terminal are out of scope. Our threat model assumption is based on real world threats. We assume that the attacker has a full control over the terminal and is able to carry out any of the following action:

1. The attacker can install a skimming device which looks very similar to the original card reader in color and texture [4]. When users insert their card in the ATM, credit and debit card information is secretly stolen and usually stored on some type of electronic device. Attacker can then encode the stolen data onto a blank card and use it to loot the user's bank account.

2. The attacker is able to install a concealed camera in the ATM which is able to record the user's input (i.e PIN) on the ATM touch screen. We assume that the camera can record multiple authentication sessions of the same user in order to extract the PIN. This technique is typically used in conjunction with the skimming device.

3. The attacker can stand behind the user and is able to observe both smartphone and ATM screen performing a shoulder-surfing attack.

4. The attacker can inject malicious software such as keylogger on the ATM to record the user's credentials. Then, she sends it over the network or the ATM printer.

5. The attacker is able to capture the transmitted information between the ATM and the user's smartphone.

6. We assume that the attacker is able to steal the user's smartphone.

3.2 The Color Wheel Pin Protocol

In this section, we present Color Wheel PIN protocol (CWPIN), which is designed to enhance the PIN input security on common touch screen devices (e.g ATM) using the user's smartphone. This protocol uses the PIN and a table of ten colors as a shared secret between the user and the server. The user is

[4] `http://krebsonsecurity.com/2010/01/would-you-have-spotted-the-fraud/`

not required to remember this table, that it is stored in the user's smartphone. When the user inserts her card to the ATM slot or taps her contactless card on the NFC reader, the ATM displays a colored wheel devised to 10 parts numbered from 0 to 9 (Color Wheel), then, the user taps his phone or scans the displayed QR code in order to receive the color table as shown in the Figure 1. To authenticate, the user has to use the colors that correspond to the first and the third PIN digits from the color table on his smartphone as an indicator to input the second and fourth PIN digits respectively on the ATM. More in detail, the CWPIN protocol comprises the following steps (see Figure 2):

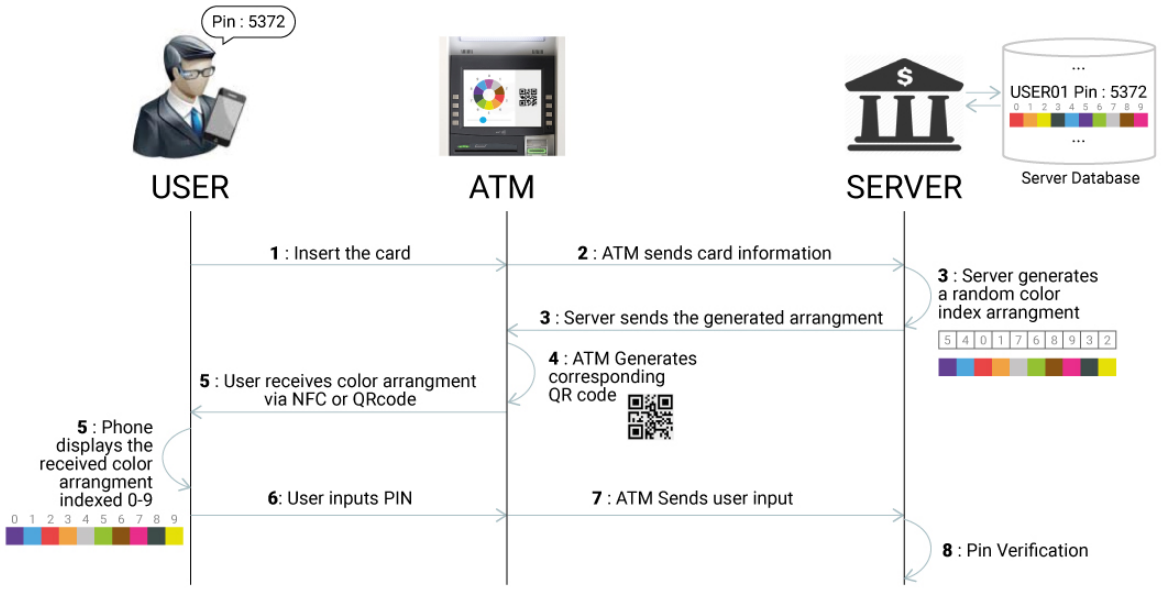


Figure 2: The CWPIN protocol.

1. The user inserts his card to the ATM slot or taps his contactless card on the NFC reader.

2. The ATM sends the card information to the server

3. The server retrieves the PIN code and the table of color corresponding to this user from the database. Each color is identified by its index in this table (e.g index of red is 0, index of blue is 4...).The server prepares a random permutation of a color arrangement (randomly shuffles the index table). The server then sends the shuffled colors index arrangement to the ATM. This way, only the user is able to have the same table of color generated by the server.

4. The ATM generates a text QR code of the received shuffled color index table (5401768932 in this example) in case that the current smartphone is not NFC enabled. Then, the ATM screen displays the Color Wheel and a seekbar to rotate the wheel along with the generated QR code.

5. The user taps his smartphone on the NFC reader or scans the QR code using the smartphone camera in order to receive the generated table, which is then displayed as horizontal array of ten colors numbered from 0 to 9 which are arranged according to the shuffled color index table received from the terminal.

6. In order to authenticate, the user only needs to perform two touch gestures, each consists in sliding the seekbar to rotate the Color Wheel. The first gesture consists in spinning the Color Wheel

so that the color that corresponds to the first PIN digit from the color table on the user's mobile device, matches the second PIN digit on the Color Wheel. Once the user lifts his finger off the seekbar, the Color Wheel spins to a random degree. The user then performs the second gesture, in which he should rotate the Color Wheel so that the color corresponding to the third PIN digit on the smartphone, matches the fourth PIN digit on the Color Wheel. Using the seekbar to rotate the Color Wheel neutralizes touch loggers and smudge attacks[22]. In the current example, the PIN is 5372, so the user has to match the green color (corresponding to the first PIN digit 5) with number 3 (second PIN digit) on the Color Wheel, then, to match the pink color (corresponding to third PIN digit 7) with number 2 (fourth PIN digit).

7. The user's input is sent to the server by the terminal.

8. The server checks whether the input was valid by verifying that the colors used to input the number 3 and 2 (i.e green and pink) are the same colors with index 5 and 7 respectively in the random table generated by the server.

Discussion. The basic idea is that the attacker can see the entire user's input but he cannot reveal the PIN since he could not know which color on the wheel has been used to input the PIN digits. The protocol is also resilient against recording and replay attack thanks to the randomization of both colors on the table and on the wheel. This way, the user will use a different color as an indicator to input the PIN digits, thereby turning the wheel to a different degree for each authentication session.

The random arrangement of colors in the table is generated by the server and shared with the user optically through QR code or by the means of NFC. There is no need to encrypt this data because the server will send the random arrangement of colors index, instead of colors arrangement. As the index of colors in the table stored on the smartphone is different for each user, an attacker will have a different table on his smartphone.

It is important to note that CWPIN protocol can be applied also using symbols or image instead of color for color blind people.

4 Security Analysis

In this section, we discuss the resilience of the proposed methodology to well-known authentication attacks.

Brute Force Attack. A Brute Force Attack is a password cracking method that uses an automated process to try all possible character combinations until the password is found [23]. In CWPIN, there are ten possible colors that can be matched with ten numbers in the two gesture performed by the user to input the PIN. Hence, we have 100 possible combinations of colors with numbers in the first gesture and 100 other combinations in the second gesture. Thus, the probability to successfully crack the user's PIN is 0,0001%. In addition, that ATM allows only three attempts before blocking the card. Therefore, CWPIN provides the same security measure as the traditional PIN against brute force attack.

Skimming attack. This kind of attack involves the installation of skimming devices which are usually undetectable by the ATM users in order to secretly record the bank account data. Afterwards, the attacker can use a cloned copy of user's credit or debit card without the user's awareness until a statement arrives with purchases they did not make. In CWPIN system, having a copy of user's card is not enough to have a successful access, the user's PIN and smartphone are required. Therefore, this kind of attacks usually is used in combination with a camera or spyware attacks to record the user's credentials.

Recording attack. A well-known attack to record the user's input is through installing a hidden camera on the ATM terminals. Unlike the traditional PIN, CWPIN is an indirect input of the user's PIN. For each authentication session, the attacker is able to infer two sets of ten combinations of numbers with their corresponding colors. However, he is not able to know which combination among them belong to the right PIN digits. In addition, since the color used as an indicator to input the second and fourth PIN digits is randomly changed for each authentication session, multiple recording of ATM screen is useless. An attacker cannot reveal any useful information by performing the intersection of the recorded data.

Spyware attack. The spyware is a malicious code used to steal the user's credentials. A typical example of this kind of attack is the keylogger. In CWPIN, recording the user's touch coordinates does not reveal the user's PIN. This is due to the randomization of i) the position of colors in the wheel upon each user's input and ii) the color in the table for each authentication session. In addition, CWPIN uses the seekbar to rotate the wheel instead of direct PIN input. Hence, the keylogger would record a different user's input each time, thereby either being unable to replay the recorded data in the next transaction or reveal the user's PIN digits from multiple recorded data. Summing up, there is no need to either assume the device to be malware-free, nor the need to implement complex techniques (e.g. [24, 25, 26] to detect malware installed on the phone.

Shoulder-surfing attack. Performing a transaction in a public space lets the user exposed to shoulder-surfing attack. An attacker is able to stand behind the user in the waiting line and try to memorize the user's input. Since the CWPIN uses an indirect input, even if the attacker observes both the smartphone and the ATM screen, he could not reveal the user's PIN. In the smartphone screen, an attacker will observe a table of color but he is not able to know which one among them has been used by the user to input the PIN digit. While on the ATM screen, when the user swipe his finger on the seekbar to input the PIN digit, all the colors of the wheel will turn with the same degree and will be matched by a specific number. Thus, an attacker cannot get any useful information about the user's PIN by observing the smartphone as well as the ATM screen.

Smartphone theft. In the CWPIN protocol, it is not possible to authenticate successfully even if the attacker steal the user's smartphone. In the worst cases, the user uses his smartphone as credit or debit card also. Thus, the attacker will be able to interact with ATM terminal until step 5. However, he is unable to complete the authentication process as the PIN code is required to make valid transaction requests successfully.

5 Experimental Results

Designing new authentication mechanisms has the following requirements to be taken into account: *memorability*, *usability* and *security*. Thus, in addition to the security factor, the passwords should be easily memorized, take short time of input at a low error rate. To this end, in this section, we evaluate the usability of the proposed scheme.

In order to put CWPIN to the test, we developed an Android application (to be installed on user's smartphone), and a touch screen desktop application to simulate the ATM and store statistics. The development tools used for the implementation of the Android application were Eclipse KEPLER SR2, android SDK 4.04 and JAVA 1.7.0. We used an open source library, called ZBar android SDK 0.2 [5], for scanning the QR codes through the smartphone camera. For the desktop application, we used the .net

[5] `http://zbar.sourceforge.net`

WPF framework under Visual Studio. The test equipment consisted of a Galaxy-I9300 smartphone (1.4 GHz Dual-Core CPU, 1 GB RAM) running Android 4.0.4 and a 15 LCD Touch screen PC.

The study was conducted with 27 volunteers with an average age of 26 years (in the rage 15 – 58), nine of them were females. At the beginning, the prototype has been explained in detail to each participant. They were encouraged to train until they felt familiar with it. When the participants felt ready, each one has been asked to perform five authentication sessions. Thus, the results are based on 135 authentication sessions performed by 27 participants. Authentication time was measured from touching the seekbar in the first gesture to lifting finger at the end of the second gesture. The logged data stored on the test equipment (i.e PC) has been used to calculate the average authentication time and the error rate of CWPIN. The time of scanning the QR code has been also logged and stored in the smartphone in order to calculate the average communication time between the smartphone and the ATM terminal.

Table 1: Experimental Results

Gender	Age	Avg. Time (msec)	Success	QR Code Avg. Scan Time (msec)
male	18	5984	5/5	285
male	15	3702	5/5	296
male	20	4621	4/5	301
female	18	5124	5/5	287
female	15	6380	5/5	306
male	18	5474	5/5	325
female	15	5627	5/5	315
male	15	6226	5/5	284
male	25	4837	5/5	307
male	25	4524	5/5	328
female	23	4586	5/5	298
male	27	3924	5/5	292
male	25	5276	5/5	300
male	27	5334	5/5	289
female	32	6454	5/5	314
male	35	4810	5/5	325
female	26	3570	5/5	290
male	58	5487	4/5	277
female	48	5680	4/5	308
male	35	4356	5/5	297
male	26	3128	5/5	295
male	30	5256	5/5	302
female	25	2992	5/5	307
male	26	2837	5/5	281
male	28	3392	4/5	276
female	24	3318	5/5	343
male	27	5128	5/5	321

The experimental results are shown in details in the Table I. This study shows that CWPIN allows the users to authenticate in a short time (i.e 4546 ms) and a low error rate (i.e. 2.96%). In addition, the scanning time for the QR code has revealed to be very fast (i.e about 300 ms).

6 Conclusion and Future Work

As the reports of the European ATM Security Team show, ATM frauds are growing both in number and in the monetary loss they cause. To tackle this problem, many banks are sponsoring a campaign targeted at updating the users' personal cards and, as a consequence, the ATM terminal too. In the "race to arm" though, criminal countermeasures to this campaign have already started to appear. In this paper, we propose a scheme capable of thwarting, among the others, also the attacks that involve a compromised ATM terminal. More in details, our scheme, called Color Wheel PIN (CWPIN) is resilient to i) brute force attacks, ii) skimming attacks, iii) recording attacks, spyware attack, iV) shoulder-surfing attack.

While it provides a high level of resilience to several types of attack, at the same time our scheme does not compromise usability as:

- it does not introduce an additional cognitive burden to the user;
- it is highly intuitive;
- it integrates an object that is nowadays very common, namely a smartphone.

To our knowledge, our scheme is the first one to provide at the same time high resilience and usability and we thus consider it a viable solution for next generation ATM terminals. Nonetheless, there are several new directions and use cases (such as access to social networks [27]) that we would like to investigate in the future. First of all, our plan is to implement CWPIN protocol on other personal wearable devices such as Google Glasses or Smart Watches, and conduct the user study to test its usability with the adoption of such devices. Second, we plan to test CWPIN in other application fields, such as strong authentication in mobile devices executing sensitive applications such as mobile payment. To pursue these goals, it will also be necessary both to take into account the energy costs of our security scheme [28]. Finally, we will further explore the problem of secure communication between the client and the server during the phase in which the randomized color table is provided.

References

[1] Alexander De Luca. Designing usable and secure authentication mechanisms for public spaces. Mai 2011.

[2] Volker Roth, Kai Richter, and Rene Freidinger. A pin-entry method resilient against shoulder surfing. In *Proceedings of the 11th ACM Conference on Computer and Communications Security*, CCS '04, pages 236–245, New York, NY, USA, 2004. ACM.

[3] Taekyoung Kwon, Sooyeon Shin, and Sarang Na. Covert attentional shoulder surfing: Human adversaries are more powerful than expected. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 44(6):716–727, June 2014.

[4] Alexander De Luca, Katja Hertzschuch, and Heinrich Hussmann. Colorpin: Securing pin entry through indirect input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1103–1106, New York, NY, USA, 2010. ACM.

[5] Desney S. Tan, Pedram Keyani, and Mary Czerwinski. Spy-resistant keyboard: More secure password entry on public touch screen displays. In *PROCEEDINGS OF THE 17TH AUSTRALIA CONFERENCE ON COMPUTER-HUMAN INTERACTION: CITIZENS ONLINE: CONSIDERATIONS FOR TODAY AND THE FUTURE*, pages 1–10. ACM Press, 2005.

[6] Mun-Kyu Lee and Hyeonjin Nam. *HCI International 2013 - Posters' Extended Abstracts: International Conference, HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part II*, chapter Secure and Usable PIN-Entry Method with Shoulder-Surfing Resistance, pages 745–748. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[7] Hirokazu Sasamoto, Nicolas Christin, and Eiji Hayashi. Undercover: Authentication usable in front of prying eyes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 183–192, New York, NY, USA, 2008. ACM.

[8] Toni Perković, Shujun Li, Asma Mumtaz, Syed Ali Khayam, Yousra Javed, and Mario Čagalj. Breaking undercover: Exploiting design flaws and nonuniform human behavior. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, SOUPS '11, pages 5:1–5:15, New York, NY, USA, 2011. ACM.

[9] A. Bianchi, I. Oakley, and Dong-Soo Kwon. Open sesame: Design guidelines for invisible passwords. *Computer*, 45(4):58–65, April 2012.

[10] Andrea Bianchi, Ian Oakley, and Dong Soo Kwon. The secure haptic keypad: A tactile password system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1089–1092, New York, NY, USA, 2010. ACM.

[11] Andrea Bianchi, Ian Oakley, Jong Keun Lee, and Dong Soo Kwon. The haptic wheel: Design & evaluation of a tactile password system. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3625–3630, New York, NY, USA, 2010. ACM.

[12] Andrea Bianchi, Ian Oakley, and Dong Soo Kwon. *Haptic and Audio Interaction Design: 6th International Workshop, HAID 2011, Kusatsu, Japan, August 25-26, 2011. Proceedings*, chapter Spinlock: A Single-Cue Haptic and Audio PIN Input Technique for Authentication, pages 81–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[13] Manu Kumar, Tal Garfinkel, Dan Boneh, and Terry Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd Symposium on Usable Privacy and Security*, SOUPS '07, pages 13–19, New York, NY, USA, 2007. ACM.

[14] Alexander De Luca, Martin Denzel, and Heinrich Hussmann. Look into my eyes!: Can you guess my password? In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pages 7:1–7:12, New York, NY, USA, 2009. ACM.

[15] N. Gobbo, A. Merlo, and M. Migliardi. A denial of service attack to gsm networks via attach procedure. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8128 LNCS:361–376, 2013.

[16] A. Armando, A. Merlo, M. Migliardi, and L. Verderame. Breaking and fixing the android launching flow. *Computers and Security*, 39(PARTA):104–115, 2013.

[17] A. Merlo, M. Migliardi, N. Gobbo, F. Palmieri, and A. Castiglione. A denial of service attack to umts networks using sim-less devices. *IEEE Transactions on Dependable and Secure Computing*, 11(3):280–291, 2014.

[18] Alexander De Luca, Emanuel von Zezschwitz, and Heinrich Hussmann. Vibrapass: Secure authentication based on shared lies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 913–916, New York, NY, USA, 2009. ACM.

[19] R. Khan, R. Hasan, and Jinfang Xu. Sepia: Secure-pin-authentication-as-a-service for atm using mobile and wearable devices. In *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2015 3rd IEEE International Conference on*, pages 41–50, March 2015.

[20] DaeHun Nyang, A. Mohaisen, and J. Kang. Keylogging-resistant visual authentication protocols. *Mobile Computing, IEEE Transactions on*, 13(11):2566–2579, Nov 2014.

[21] M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, and B. Messabih. A completely automatic public physical test to tell computers and humans apart: A way to enhance authentication schemes in mobile devices. pages 203–210, 2015.

[22] Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies*, WOOT'10, pages 1–7, Berkeley, CA, USA, 2010. USENIX Association.

[23] Mudassar Raza, Muhammad Iqbal, Muhammad Sharif, and Waqas Haider. Doi: 10.5829/idosi.wasj.2012.19.04.1837 a survey of password attacks and comparative analysis on methods for secure authentication.

[24] M. Curti, A. Merlo, M. Migliardi, and S. Schiappacasse. Towards energy-aware intrusion detection systems on mobile devices. pages 289–296, 2013.

[25] Migliardi M. Fontanelli P. Merlo, A. Measuring and estimating power consumption in android to support energy-based intrusion detection. *Journal of Computer Security*, 23(5):611–637, 2015.

[26] A. Merlo, M. Migliardi, and P. Fontanelli. On energy-based profiling of malware in android. pages 535–542, 2014.

[27] L. Caviglione, M. Coccoli, and A. Merlo. A taxonomy-based model of security and privacy in online social networks. *International Journal of Computational Science and Engineering*, 9(4):325–338, 2014.

[28] Migliardi M. Caviglione L. Merlo, A. A survey on energy-aware security mechanisms. *Pervasive and Mobile Computing*, 24:77–90, 2015.