I. J. Computer Network and Information Security, **2024, 5, 113-123**
Published Online on October 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2024.05.09

Finding and Mitigating a Vulnerability of the Color Wheel PIN Protocol

Samir Chabbi
University of Souk Ahras / Department of Mathematics and Informatics, BP 1553 Souk Ahras 41000, Algeria
E-mail: s.chabi@univ-soukahras.dz
ORCID iD: https://orcid.org/0000-0001-9069-5544

Djalel Chefrour*
University of Souk Ahras / Department of Mathematics and Informatics, BP 1553 Souk Ahras 41000, Algeria
E-mail: djalel.chefrour@univ-soukahras.dz
ORCID iD: https://orcid.org/0000-0001-8068-4489
*Corresponding Author

Nour El Madhoun
LISITE Laboratory, ISEP, 10 Rue de Vanves, Issy-les-Moulineaux, 92130, France
E-mail: nour.el-madhoun@isep.fr
ORCID iD: https://orcid.org/0000-0001-7742-7748

Received: 22 May 2023; Revised: 10 August 2023; Accepted: 27 October 2023; Published: 08 October 2024

Abstract: There is an increasing usage in the banking sector of Smartphones enabled with Near Field Communication (NFC), to improve the services offered for the customers. This usage requires a security enhancement of the systems that employ this technology like the Automated Teller Machines (ATMs). One example is the Color Wheel Personal Identification Number (CWPIN) security protocol designed to authenticate users on ATMs using NFC enabled smartphones without typing the PIN code directly. CWPIN has been compared in the literature to several other protocols and was considered easier to use, more cost-effective and more resistant to various attacks on ATMs such as card reader skimming, keylogger injection, shoulder surfing, etc. Nevertheless, we demonstrate in this paper that CWPIN is vulnerable to the multiple video recordings intersection attack. We do so through concrete examples and a thorough analysis that reveals a high theoretical probability of attack success. A malicious party can use one or two hidden cameras to record the ATM and smartphone screens during several authentication sessions, then disclose the user's PIN code by intersecting the information extracted from the video recordings. In a more complex scenario, these video recordings could be obtained by malware injected into the ATM and the user's smartphone to record their screens during CWPIN authentication sessions. Our intersection attack requires a few recordings, usually three or four, to reveal the PIN code and can lead to unauthorized transactions if the user's smartphone is stolen. We also propose a mitigation of the identified attack through several modifications to the CWPIN protocol and discuss its strengths and limitations.

Index Terms: Authentication Protocol Vulnerability, ATM Security, NFC Smartphones, Attack Mitigation.

1. Introduction

Near Field Communication (NFC) is a technology used to enable contactless communication, within a short distance (maximum 10 centimeters), between two devices supporting it. NFC uses an electromagnetic field to transfer messages at a frequency of 13.56 MHz [1]. It is frequently used in the banking sector to perform contactless transactions. For example, instead of a traditional bank card, an NFC smartphone can be used at an Automated Teller Machine (ATM) to make an NFC cash withdrawal/deposit transaction [2].

NFC allows fast contactless transactions at any ATM [3] and is therefore beneficial to both clients and banks. In general, to perform an NFC enabled cash withdrawal/deposit transaction, the client needs to bring either his NFC bank card or his NFC smartphone close to an ATM. The latter reads the customer data from such a device and starts and authentication session. It also transmits the customer data to the bank's server that participates in the customer authentication before proceeding to the transaction authorization [4]. User authentication is an important security property that must be ensured during an NFC cash withdrawal/deposit transaction [5].

Unfortunately, an attacker can remotely steal the user banking data (e.g., username, Primary Account Number (PAN), expiration date, etc.) stored in an NFC bank card or an NFC smartphone and use it to harm the victim. In addition, the users' passwords and Personal Identification Number (PINs) can be stolen through many attacks such as: shoulder-surfing [6], smudge [7], brute force [8], side channel [9], replay [10], spyware [11], camera recording [12], video recording [13] and multiple registrations [14]. For this reason, there is a crucial need to improve the security of ATMs and protect them against malicious attacks, a need that is well established in the literature [15, 16]. In particular, securing the PIN code when an NFC smartphone is used for cash withdrawal/deposit transaction with an ATM is of the most importance [17]. For this purpose, the PIN code is stored in NFC smartphones in a hardware circuit called the Secure Element (SE). In addition to providing secure data storage, the SE protects the execution of sensitive applications, such as banking applications, from malware [18].

We are interested in studying the security of an ATM authentication protocol called Color Wheel Personal Identification Number (CWPIN). This protocol relies on NFC enabled smartphones to protect cash withdrawal/deposit in ATMs [19]. The CWPIN protocol is considered more cost-effective and more secure than many other protocols [19]. However, we carry in this paper a thorough analysis with concrete examples to show that CWPIN is vulnerable to the multiple video recordings intersection attack. We also prove the effectiveness of this attack by formulating its theoretical probability of success. In our attack, a malicious party can use one or two hidden cameras to record the ATM and smartphone screens during several authentication sessions. Then, the attacker can disclose the user's PIN through information (extracted from the video recordings) intersection, even if the PIN code is not typed directly. We show that in most cases three video recordings are enough to break CWPIN and that with five videos the attacker is assured to reveal the user PIN.

This paper is organized as follows. Section 2 discusses the related works, while section 3 presents the CWPIN authentication protocol in detail. Section 4 exposes the main goal of this work which is: the disclosure of CWPIN vulnerability and its exploitation in an attack that has a high probability of success. Section 5 describes some improvements to CPWING that we suggest in order to mitigate its vulnerability. Finally, we present our conclusion and perspective in section 6.

2. Related Works

On the one hand, we summarize here after recent authentication schemes proposed for banking customer's authentication. We start with CWPIN then cover similar solutions. On the other hand, we discuss few security attacks that resemble our attack on CWPIN.

2.1. Authentication Methods for Banking Customers

Several methods have been proposed in the literature to authenticate bank customers during NFC cash withdrawal/deposit transactions [20]. Some methods rely on a PIN code or a password, while others use a biometric modality of the user (e.g., facial recognition, fingerprint recognition, voice recognition, etc.). These authentication methods have their advantages and limitations. Nevertheless, when compared to PIN or password-based authentication, the cost of biometric authentication methods outweighs their benefits in most cases [21]. The major inconvenience of a biometric method is when a biometric characteristic is stolen. In this case, the attacker can use the stolen characteristic at any time insofar as the user cannot change it, contrarily to a PIN code or a password. Hence, we concentrate in the remaining of this section on the most important PIN or password-based authentication methods beginning with CWPIN.

The CWPIN protocol is used during a cash withdrawal/deposit transaction where the user authenticates himself at an ATM by using an NFC smartphone application, instead of typing his PIN code directly [19]. After reading the customer identification via NFC, the ATM sends it to the bank server which generates a random index of by shuffling the numbers from 0 to 9. This shuffled index is sent to the ATM, which relays it to the user's smartphone via NFC. Then, on the one hand, the ATM displays on its screen a random wheel of 10 colors and a seek-bar to spin it, as shown in fig. 1. The colored portions of this wheel are surrounded by a fixed index from 0 to 9. On the other hand, the smartphone displays a colored array built by combining the shuffled index and a user specific color table stored in its Secure Element. This color array is also topped on the smartphone screen with an index that goes from 0 to 9. If NFC is not available in the smartphone, CWPIN can read the shuffled index from the ATM screen by scanning a QR code.

In order to authenticate, the CWPIN user selects the color corresponding to the first digit of his PIN code from the smartphone's array and spins the ATM wheel to match this color to the second digit of his PIN code. After this match, the wheel will rotate randomly. Then, the user repeats the previous step to match the color of the array corresponding to the PIN third digit with the PIN fourth digit on the ATM screen.

In ref. [22] the authors propose a novel PIN based ATM authentication solution called Dynamic Array PIN (DAP) and compare it to other schemes including CWPIN. DAP is considered more secure and more efficient as it works on the ATM only and does not require a smartphone using NFC or QR code scanning. One drawback of CWPIN that does not exist in DAP is that it can be difficult for the elder CWPIN users to memorize the correspondence between the digits and the colors.

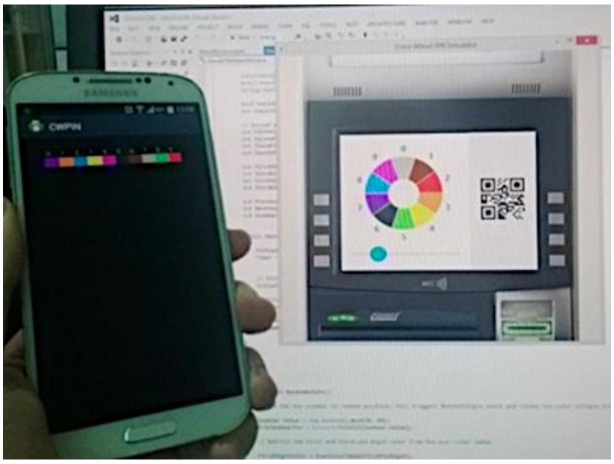


Fig.1. CWPIN displays on the smartphone a table of numbered colors obtained from the ATM via NFC or scanning a QR code [20]

The Revolving flywheel PIN method presented in [23] uses a wheel with three concentric layers: the outer layer, the middle layer and the inner layer. All layers contain ten sectors and are filled with four colors. The middle layer of the wheel contains the numbers displayed along with colors. The user cannot use this layer. This method uses only the inner and outer layers. To enter a digit, the user must observe the sector containing this digit, and then press the color present in the upper section plus the lower section of his sector. The user applies two clicks to enter a single digit, and then the wheel will rotate randomly clockwise and counterclockwise in order to fool the attacker. The analysis of this method shows that it is simple, has no cognitive load and is easy to follow by an ATM. On the other hand, this method is vulnerable to attacks that record the screen and the clicks of the user and can reveal his confidential data.

In the study [8], the authors propose a scheme based on honey encryption to protect Java smart card passwords against both brute force attacks and Denial of Service (DoS) attacks. Traditionally, smart cards systems store the user information in hashed form and will limit the number of retries when an attacker provides a wrong PIN. However, this leads to a DoS situation where the legitimate users are blocked from the system. The proposed method uses the principle of honey pot that exposes fake but credible data that the attacker is trying to access in the system. Its purpose is to deceive brute force attacks into believing that they have found the user PIN code and therefore stop their queries. Nonetheless, an attacker can still discover that the information he obtained is fake by comparing it with real information about the target victim, if he manages to collect the latter via other means. One such mean is a channel attack that gives access to the encrypted storage in the card system. By hashing the fake information, the attacker will know that it is fake if its hash is not present in this storage.

2.2. Intersection Attacks

Authors in the study [24] show an intersection attack on recognition-based graphical passwords. In this authentication method, the user is presented with several challenge screens during a single session. Each screen contains one image from a set of pass images that allow access and are specific to each user. The other images on each screen are distractors. The attacker makes several authentication attempts, and then he identifies the pass images as those that have been seen most frequently. In a sense, this attack is like ours (see sections 3 and 4), as the distracter images have a lower probability of reappearing, as do random permutations of two color-matched digits in CWPIN. Nevertheless, the authors study the effectiveness of their attack and its countermeasures through a simulation. Their goal is to identify the best mitigation from the statistics of the number of attempts needed before success. On the other hand, in addition to illustrative examples, we provide a formal analysis of the probability of success of our intersection attack based on the number of video recordings.

Our attack on the CWPIN protocol is in some respects similar to Statistical Disclosure Attacks (SDA) on anonymizing systems [25]. These systems hide the identities of senders, receivers, and their messages in communication networks within a mixture of other messages. The sender's message is cryptographically transformed and stored in the system until the stock reaches a certain threshold, after which the mixture of stored messages is sent to their receivers in a single round. The SDA attack assumes that a sender transmits no more than one message in a round and that the rest of the mix is uniform background traffic. This is somewhat like the CWPIN property that the permutation of two PIN digits in a disclosed set of 10 is always present in subsequent sessions, while the other nine are random permutations. Furthermore, SDA statistically isolates the user's sending behavior by observing a large number of receiver sets. Therefore, while the probability formula that calculates the success of our attack on CWPIN is simple, the calculation of the SDA probability that a certain set is the set of a receiver for a particular sender is exponential. This has led the SDA authors to use approximate probabilities.

3. Authentication with the Color Wheel PIN Protocol (CWPIN)

The Color Wheel PIN Protocol handles authentication sessions by following the next steps:

- The user brings his NFC smartphone close to the ATM, which retrieves the data needed for the transaction from the smartphone via NFC and transmits it to the bank server.
- The bank server fetches the user's PIN code and a table of colors from its database. Each color is characterized by its index varying from 0 to 9. The server generates a random arrangement of colors indexes and sends it to the ATM.
- The ATM sends the shuffled index to the smartphone via NFC. The ATM displays a random color wheel and a seek-bar to rotate it. The colored portions of the wheel are surrounded by numbers from 0 to 9 on the ATM screen.
- The smartphone displays on its screen a horizontal table of ten colors built from the shuffled index and the color table stored in its SE. This color table is topped with an index from 0 to 9.
- The user selects the color corresponding to the first digit of his PIN from the smartphone's table, then on the ATM screen, spins the color wheel to match this color to the second digit of his PIN.
- At the end of the user movement, the ATM rotates the color wheel with a random degree.
- The user repeats step 5 by picking (from the phone screen) the color that corresponds to the PIN’s third digit, then spins the wheel (on the ATM screen) to match this color with the PIN’s fourth digit.
- The ATM sends the user input to the bank server, which checks the validity of the entered PIN by comparing it with the one stored in its database.

4. Vulnerability of the CWPIN Protocol

In this section, we demonstrate how the CWPIN protocol is vulnerable to a multiple video recordings intersection attack through theoretical analysis and experimental illustration.

4.1. Threat model and attack principle

We assume that the attacker can record both the user's smartphone screen and the ATM terminal during several legitimate CWPIN authentication sessions. This can be achieved by using one or two hidden cameras judiciously in the vicinity of an ATM frequently used by the targeted person. In the attacker's best-case scenario, a camera hidden in the middle of the ATM's top, covering its screen and keyboard, should capture both the ATM's screen (including the CWPIN seek bar) and the user's smartphone screen. As the user will touch the seek bar to rotate the color wheel, it is natural for him to keep his smartphone in his other hand, close and parallel to the ATM screen for easy color matching.

It could be argued that a single camera is not enough to cover the smartphone screen, if the smartphone is held by the user at an angle perpendicular to the ATM screen or close to it. Therefore, to accommodate such a case, the attacker would have to use two cameras placed in the upper left and right corners of the ATM. This would also allow targeting both right- and left-hand users. The two cameras can be hidden in a horizontal panel that also includes batteries to allow recording for long periods. Instead of one or two cameras on the ATM, the attacker can record the user's sessions while shoulder surfing, with a smartphone camera or a camera hidden in a glass.

Alternatively, in a more complex scenario, the attacker can target the user's smartphone and ATM terminal with malware that records and exfiltrates video from the screens of both devices. This is possible with screen capture software if the attacker manages to run it remotely on these devices at opportune times. Once acquired, the multiple video recordings of legitimate authentication sessions will be analyzed by the attacker to reveal the correct PIN code. We illustrate and demonstrate in the next section that three to four recordings are sufficient for the attack to work. Additionally, we assume that the PIN code is not changed by the user while being targeted by this attack.

To use the disclosed PIN in unauthorized transactions, we assume that the attacker can take over the user's smartphone. All these assumptions are realistic conditions that can be met by a determined malicious party pursuing specific targets. In the case of malware injection, the malicious party may target multiple customers from multiple banks that use the CWPIN protocol.

Finally, the principle on which our attack relies can be expressed as the following: with each CWPIN authentication session recorded, the attacker can easily discover 10 possible permutations for the first two digits of the PIN code and another 10 permutations for its last two digits. These two sets of permutations have an interesting property: Only one two-digit permutation in each set belongs to the PIN code and will appear in each authentication session, while the remaining nine permutations are random two-digit matches that are unlikely to reappear in successive CWPIN sessions. Consequently, the intersection of some of these sets of 10 possible permutations will yield a singleton that contains two digits belonging to the PIN code.

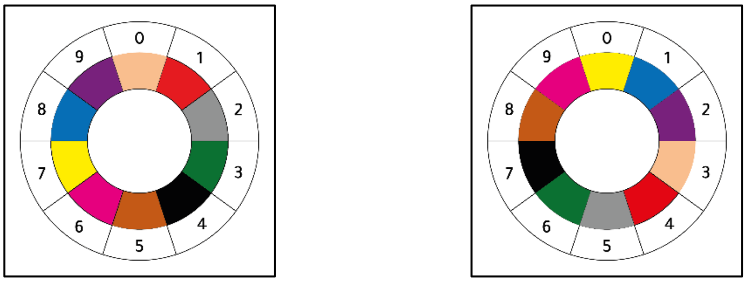
4.2. Experimental Illustration

We illustrate hereafter a successful attack on the CWPIN protocol through the steps performed by a legitimate user during multiple authentications. We interweave these steps with details of what a malicious party needs to extract from a video recording of each session and what actions it takes to disclose the user's PIN. We assume that the PIN in this example is “9026”.

At the beginning of the first ATM authentication session, the user of the CWPIN protocol will see the indexed colors of Table 1 displayed on his smartphone and the color wheel of fig. 2(a) displayed on the ATM screen. To enter the PIN code, the user first identifies yellow that corresponds to the PIN first digit, which is 9, in the color table of his smartphone. Then, the user rotates the ATM wheel so that its yellow portion corresponds to the PIN second digit, which is 0, and releases his finger. The exact moment when the user releases the ATM seek bar is shown in fig. 2(b).

Table 1. Color table displayed on the smartphone at the beginning of the first CWPIN session

0	1	2	3	4	5	6	7	8	9
🟥	🟩	🟧	🟦	⬜	🟪	🩷	🟫	⬛	🟨



(a) (b)

Fig.2. Color wheel displayed on the ATM screen: (a) at the start of the first CWPIN session, (b) just after matching the first two digits of the PIN

Fig. 2(b) constitutes a crucial moment for the attacker who looks for it in the video recording of the session. Specifically, if the seek bar is visible in the video recordings, it is obvious to capture the exact moment of the release of the user finger. Otherwise, to make a better estimate, the attacker relies on the difference between the speeds at which the wheel is rotated by the user (via the seek bar) and spun by the ATM (at a random angle, just after the finger is released). The first speed is generally slow and variable because it is caused by a manual gesture of the human, while the second is artificial because it is generated by a computer.

This artificial spinning movement is not described in detail in the original paper [19]. Therefore, we suppose that the wheel spins at a constant speed followed by a sudden stop. Alternatively, the rotation of the wheel can be implemented with a monotonic, steady increase in speed, followed by a monotonic decrease. In any case, there is no evidence in the original description of CWPIN that this rotation is purposefully made to look like a human gesture, in order to deceive any attack. Also, the description does not mention the direction of this automatic rotation. Therefore, if the rotation always goes in the same direction (clockwise or counterclockwise) and the user has rotated the wheel in the opposite direction, the attacker will then easily find the crucial moment described previously.

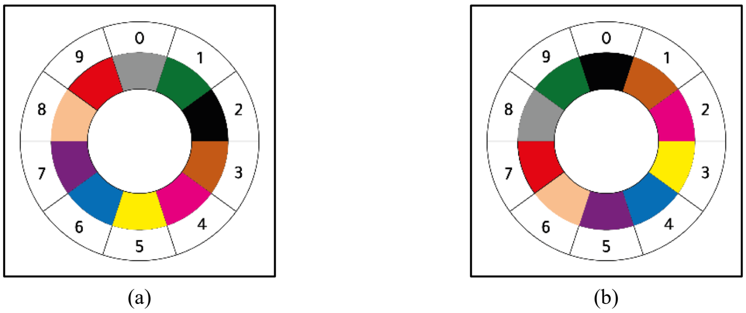
With a screenshot of the smartphone and a snapshot of the ATM screen at that crucial moment depicted in fig. 3, an attacker can easily guess the possible permutations of the first and second digits of the PIN. Indeed, as one of the different colors is used to match two digits between the two screens, the attacker will simply constitute a table of the 10 possible permutations of two digits by comparing each of the 10 colors of the two screens. This is table 2 in our example.

Table 2. Possible permutations of the first two digits of the PIN from the first CWPIN session

1st digit (smartphone)	0	1	2	3	4	5	6	7	8	9
Matching color	🟥	🟩	🟧	🟦	⬜	🟪	🩷	🟫	⬛	🟨
2nd digit (ATM)	4	6	3	1	5	2	9	8	7	0

When the user releases his finger after entering the first two digits, the ATM will rotate the color wheel by a random degree, say 180° in this example. So, the ATM displays the wheel shown in fig. 3(a), after which the user proceeds to enter the PIN last two digits by identifying the shrimp color in the smartphone's color table as the one corresponding to digit 2 as it is the third digit of the PIN. Then, he turns the wheel so that its shrimp portion matches the digit 6, which is the PIN fourth digit (see fig. 3(b)). Finally, the ATM validates the entered PIN code with the bank server and authorizes the user.

Once again, at the point depicted by fig. 3(b), the attacker will be able to guess the 10 possible permutations of the last two digits of the PIN code. These are obtained as before by comparing the smartphone and ATM screens and matching the corresponding indexes to each color. The result of this step is shown in Table 3.



(a) (b)

Fig.3. Color wheel: (a) after spinning with a random degree in the middle of the first CWPIN session, (b) just after entering the tow last PIN digits

Table 3. Possible permutations of the last two digits of the PIN from the first CWPIN session

3rd digit (smartphone)	0	1	2	3	4	5	6	7	8	9
Matching color	🟥	🟩	🟧	🟦	⬜	🟪	🩷	🟫	⬛	🟨
4th digit (ATM)	7	9	6	4	8	5	2	1	0	3

At the end of this first authentication session, the attacker will have 10x10 possible permutations of the four digits of the PIN code. One of these permutations is certainly the correct PIN. To exclude the wrong permutations, the attacker needs other video recordings of valid authentication sessions where these permutations do not appear.

Table 4 and fig. 4 show what the user gets at the beginning of the second CWPIN authentication session. The colors that will be used to match the PIN digits two by two are respectively green and blue.

Table 4. Color table displayed on the smartphone at the beginning of the second CWPIN session

0	1	2	3	4	5	6	7	8	9
⬛	🟨	🟦	🩷	⬜	🟪	🟧	🟫	🟥	🟩

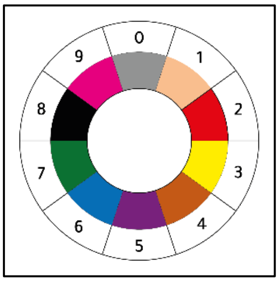
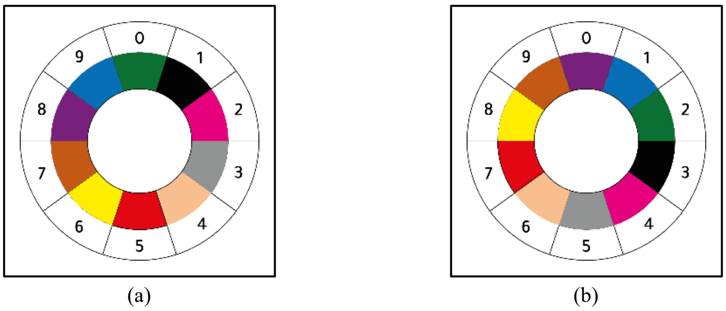


Fig.4. Color wheel displayed on the ATM at the start of the second CWPIN session

As with the first session, the attacker will look for the crucial moments when the user's finger is released after submitting the first two digits of the PIN code, and then the last two. These moments are shown in fig. 5(a) and 5(b). Each of these moments will help the attacker to extract the possible permutations of two digits, which are respectively shown in Tables 5 and 6.



(a) (b)

Fig.5. State of the color wheel just after entering (a) the first two and (b) the last two digits of the PIN in the second CWPIN session

Table 5. Possible permutations of the first two digits of the PIN from the second CWPIN session

1st digit (smartphone)	0	1	2	3	4	5	6	7	8	9
Matching color	⬛	🟨	🟦	🩷	⬜	🟪	🟩	🟫	🟥	🟩
2nd digit (ATM)	1	6	9	2	3	8	4	7	5	0

Table 6. Possible permutations of the last two digits of the PIN from the second CWPIN session

3rd digit (smartphone)	0	1	2	3	4	5	6	7	8	9
Matching color	⬛	🟨	🟦	🩷	⬜	🟪	🟩	🟫	🟥	🟩
4th digit (ATM)	8	3	6	9	0	5	1	4	2	7

At this point, the attacker can omit the corresponding-colored rows from Tables 2, 3, 5, and 6; consider each column with a two-digit permutation to be an element of a set; intersect the set obtained from Table 2 with the one from Table 5 to obtain a set of possible combinations for the first two digits of the PIN {16, 90}; and do the same with Tables 3 and 6 to get a set of possible combinations for the last two digits of the PIN {26,55}. In other words, at the end of the second authentication session, the attacker reduces the possible permutations of each half of the PIN code to only two possibilities for each half. Therefore, the possible permutations for the whole PIN code in this example are only four: {1626, 1655, 9026, 9055}. Knowing that the ATM offers three possible attempts for the user to enter the PIN code, the attacker already has three chances out of four to find the correct PIN. However, to increase his chances in finding the correct PIN, the attacker proceeds to analyze the video of a third CWPIN authentication session.

For reasons of brevity, we do not show the screens of the smartphone and the ATM screens from the third recorded session. But we give only the possible permutations of the PIN digits, two by two, obtained by the attacker at the end of this session as shown in Tables 7 and 8.

Table 7. Possible permutations of the first two digits of the PIN from the third CWPIN session

1st digit (smartphone)	0	1	2	3	4	5	6	7	8	9
Matching color	⬜	🟨	⬛	🩷	🟦	🟩	🟩	🟪	🟩	🟥
2nd digit (ATM)	8	7	9	2	5	1	6	3	4	0

Table 8. Possible permutations of the last two digits of the PIN from the third CWPIN session

3rd digit (smartphone)	0	1	2	3	4	5	6	7	8	9
Matching color	⬜	🟨	⬛	🩷	🟦	🟩	🟩	🟪	🟩	🟥
4th digit (ATM)	5	4	6	9	2	8	3	0	1	7

Similarly, by extending the intersection step of Tables 2 and 5 to Table 7 we obtain the set {90}, which yields the first two digits of the user PIN. Likewise, the intersection of Tables 3, 6 and 8 gives the set {26}, which reveals the last two digits of the PIN. Finally, the correct PIN was disclosed by analyzing the video recordings of only three CWPIN authentication sessions.

4.3. *Theoretical Analysis*

For simplicity, while analyzing why the multiple record intersection attack works on the CWPIN protocol, we will only consider the first two digits of the PIN code in this section. The same logic applies to its last two digits because our attack reveals each half of the PIN code independently of the other half.

We start from the following intuitive observation: the permutation of the two digits belonging to the PIN (and corresponding to the color chosen by the user) will always appear in successful CWPIN authentication sessions (provided that the PIN is unchanged between these sessions); whereas, the other permutations of two digits derived by the attacker (from the indexes associated with the remaining nine colors) will have a very low chance of reappearing in successive authentication sessions. This observation can be confirmed by calculating the probability P that the intersection of the sets of possible permutations of two digits obtained from n CWPIN sessions will yield only the digits of the PIN. This condition is equivalent to saying that none of the nine random permutations will reappear in all these n sessions. If our observation is correct, then P should be very high (close to 1) for a small n. This in turn means that only a few video recordings are needed for the attacker to disclose the correct PIN.

More formally, when considering the permutations of two decimal digits in general (regardless of CWPIN), we are dealing with “partial permutations with repetition” of two digits from 10. That is a sample space of 102 = 100 possibilities {00, 01, 02 ... 98, 99}. However, in the case of CWPIN, this space becomes smaller thanks to constraints that are inherent to this protocol design. Namely, the number of possible permutations that can appear in the columns of one table extracted by an attacker from a valid session (e.g., table 8) is smaller for the following reasons. Firstly, one permutation (i.e., one column in each table) is fixed as it belongs to the PIN and will always show up. Secondly, the other nine permutations/columns are drawn uniformly at random from the space of all possible “permutations without repetition”, because any digit (belonging to the PIN or not) is not repeated within each row of the table. More precisely, because the first row of each table is fixed as it comes from the smartphone screen, the random permutations show up in nine of the cells of the second row of the table which comes from the color wheel on the ATM screen. Thus, their number in one session, which is the size of the sample space from which the attacker is drawing, is 9!.

In a similar way, the number of possible samples where a random permutation is always present is equal to the number of permutations without repetition of the remaining eight columns of the table: 8!. Hence, the probability that a random permutation of two digits not in the PIN appears in the set extracted by the attacker from one CWPIN session is equal to: 8!/9!=1/9=0.11.

We consider now the case of n CWPIN authentication sessions where any nine random permutations of two digits not in the PIN appeared in the first session. The probability that one of these random permutations will be present in all the remaining (n-1) sessions is actually $(0.11)^{(n-1)}$, because these sessions are independent of each other. By contrast, the probability of the complementary event, i.e., that a random permutation is absent from at least one of the (n-1) sessions, is therefore equal to: $1-(0.11)^{(n-1)}$.

Finally, for the probability condition P to be true, each of the nine random permutations from the first session must disappear at least once in the subsequent (n-1) sessions. Likewise, the same analysis holds for the nine random permutations of the two last digits in the PIN. So, for a whole PIN of four digits, we need to consider 18 random permutations of two digits. Consequently, P can be expressed by the formula (1):

$$P(n) = (1 - (0.11)^{n-1})^{18} \quad (1)$$

Table 9 shows that, for a precision of two decimal places, P quickly converges to 1 when n reaches 5. In other words, as mentioned earlier, an attacker is guaranteed to disclose the user's PIN with the intersection of the information extracted from five video recordings. In most practical cases, three recordings are sufficient to yield one or two possible PINs that can be tried on the ATM, which usually allows three tests.

Table 9. Probability of attack success by finding the PIN code with n video recordings

n	P(n)
2	0.12
3	0.80
4	0.98
5	1.00

We note that the attack we successfully demonstrated on the CWPIN protocol is not affected by the fact that the wheel is spun by a random degree at the end of the user's interactions. Indeed, the attack only requires a capture of the smartphone and ATM terminal screens at the crucial moment when the user releases his finger, after entering the first or the last two digits of the PIN code. The attack does not depend on the initial state of the color wheel. For this reason, it would work even if the CWPIN protocol has been modified to make the color wheel index completely random after the first two digits are entered.

5. Attack Mitigation

Considering the previous details of the CWPIN protocol's vulnerability to the multiple video recording intersection attack, it becomes clear that to mitigate this vulnerability, we need to prevent any malicious party from detecting the crucial moment when the PIN digits are matched together. This is when users release their fingers after using the seek bar on the ATM screen. Obviously, it is not enough to ask the user to cover his finger with the other hand. As stated in the threat model, the attacker can always guess the crucial moment from the difference between the rotation speed of the wheel when it is turned manually and when it is turned by the ATM. In addition, using the other hand as a cover is extra work that is easily forgotten. Therefore, to prevent the intersection attack of multiple video recordings, we propose in the following subsections detailed modifications to the original CWPIN protocol, then a discussion of their feasibility, purpose and limit.

5.1. Suggested CWPIN Modifications

We suggest in the next list six modifications to the original CWPIN protocol. The aim is to make the crucial moment of the user's finger release indistinguishable for the attacker, even by reviewing the video recordings several times.

- Remove the seek bar completely from the ATM screen.
- Use instead a trackball placed next to the numeric keypad of the ATM terminal, to spin the color wheel.
- Cover the trackball with a shell that has a front opening similar to the one in fig. 6.

- Change the rotation of the wheel on the ATM screen into a "mechanical" jerking movement similar to that of the second hand on a wall clock.
- When the user stops rolling the trackball, continue the spinning of the color wheel in the same direction for a random number of indexes.
- Include one or two changes of direction at random indexes to further deceive the attacker.

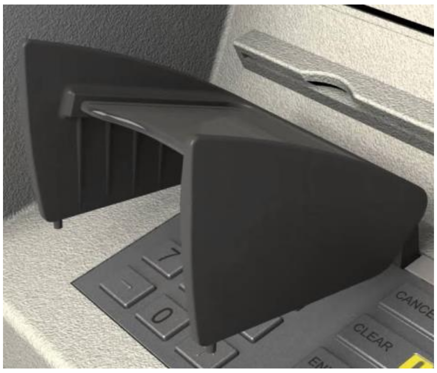


Fig.6. Example of a shell that hides the user’s finger when spinning the color when on the ATM screen

5.2. Discussion

Most of our proposed modifications to the CWPIN protocol are software-based and easy to implement. The only hardware modification required for our attack mitigation is the addition of a trackball covered by a shell. The purpose of this shell is to prevent any eavesdropping of the user's finger movements. A trackball, which is sometimes called a mouse ball or spherical mouse, allows the user to rotate the color wheel on the screen very easily. In fact, it is more intuitive to use for this particular task than a seek bar. Another alternative that achieves the same goal and is of equivalent benefit is a scroll wheel similar to the middle button of a mouse. However, for the same negligible cost, a trackball can also serve as a pointing device on the ATM screen.

The main software change we require for CWPIN concerns the color wheel spinning on the ATM screen, which should always look the same, regardless of whether the initiator is a human user or a computer. That is, the color wheel always moves from one index to another, at a constant pace, regardless of who causes its rotation.

Finally, we note that with our countermeasure, the CWPIN protocol is still vulnerable to a trackball logger injected into the ATM, just as its original version is vulnerable to a touch recorder. These recorders can always detect the end of the user's interaction with the color wheel, regardless of the last additional random spinning. Therefore, they can find the corresponding ATM screenshot in the video capture and continue the attack successfully.

6. Conclusions

We set out to disclose and prove that the CWPIN ATM authentication protocol is vulnerable to a multiple video recording intersection attack. We have shown that this attack exploits the fact that the PIN digits are entered two by two using one color each time. By comparing the indexes associated with each color on the smartphone and ATM screens at crucial times, an attacker can respectively reduce the possible permutations of the first two and last two digits of the PIN code to 10. Furthermore, this attack exploits the fact that nine of these 10 permutations are random matches of two digits that do not belong to the PIN code. In fact, their randomness depends on the quality of the entropy of the random number generator used by the implementation of the CWPIN protocol (when mixing the color indexes). The better the random generator, the less likely it is that the same permutation of two digits (not in the PIN code) will appear in subsequent authentication sessions, and the easier the attack. In other words, what was conceived as the strength of the CWPIN protocol is actually a weakness, as demonstrated by the high probability of attack success.

Finally, we not only found that the CWPIN protocol is vulnerable to the multiple video recording intersection attack, but we also proposed a mitigation of this attack. This mitigation masks the crucial moment when the PIN digits are matched by removing any difference between the user-generated and computer-generated color wheel rotations. Once implemented as detailed in the previous section, this mitigation requires an evaluation study that validates both its effectiveness and usability. We think that such a study requires the involvement of a representative number of test users and deserves its own publication. Therefore, we leave it as a perspective for future work. Additionally, the problem of a malicious party using a trackball logger to conduct the proposed attack remains possible and hence is an open question.

References

[1] Giese D., Liu K., Michael Sun, Syed T. and Zhang L., “Security Analysis of Near-Field Communication (NFC) Payments”, ArXiv, 2019. DOI:10.48550/arXiv.1904.10623

[2] Merkus J., “Security evaluation of the NFC contactless payment protocol using Model Based testing”, Master's thesis, Open University of Nederland, 2018.

[3] Wadii, E. L., Boutahar, J., Ghazi, S. E., “NFC Technology for Contactless Payment Ecosystems”, International Journal of Advanced Computer Science and Applications, Vol.8, No.5, pp.391-397, 2017.

[4] Alqassab A., Hikmat Ismael Y., “EMV Electronic Payment System and its Attacks: A Review”, AL-Rafidain Journal of Computer Sciences and Mathematics, Vol.16, No.1, pp.23-29, 2022. DOI:10.33899/CSMJ.2022.174392

[5] Chanal P. M., Kakkasageri M. S., “Security and privacy in IOT: a survey”, Wireless Personal Communications, Vol.115, No.2, pp.1667-1693, 2020. DOI:10.1007/s11277-020-07649-9

[6] Alsuhibany S. A., “A Camouflage Text-Based Password Approach for Mobile Devices against Shoulder-Surfing Attack”, Security and Communication Networks, 2021. DOI:10.1155/2021/6653076

[7] Shin H., Sim S., Kwon H., Hwang S., Lee Y., “A new smart smudge attack using CNN”, International Journal of Information Security, Vol.21, pp.25-36, 2022. DOI:10.1007/s10207-021-00540-z

[8] Mohammed S., Kurnaz S., Mohammed A. H., “Secure Pin Authentication in Java Smart Card Using Honey Encryption”, In 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), pp.1-4, 2020, IEEE. DOI:10.1109/HORA49412.2020.9152936

[9] Chen D., Zhao Z., Qin X., Luo Y., Cao M., Xu H., Liu A., “MAGLeak: A learning-based side-channel attack for password recognition with multiple sensors in IIoT environment”, IEEE Transactions on Industrial Informatics, Vol.18, No.1, pp.467-476, 2022. DOI: 10.1109/TII.2020.3045161

[10] Shang J., Wu J., “LightDefender: Protecting PIN Input using Ambient Light Sensor”, In 2020 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp.1-10, IEEE, 2020. DOI: 10.1109/PerCom45495.2020.9127361

[11] Shammee T. I., Akter T., Mou M., Chowdhury F., Ferdous M. S., “A Systematic Literature Review of Graphical Password Schemes”, Journal of Computing Science and Engineering, Vol.14, No.4, pp.163-185, 2020. DOI: 10.5626/JCSE.2020.14.4.163

[12] Shubhra J., “ATM frauds: Detection & Prevention”, International Journal of Advances in Electronics and Computer Science, Vol.4, No.10, 2017.

[13] Guerar M., Migliardi M., Palmieri F., Verderame L., Merlo A., “Securing PIN-based authentication in smartwatches with just two gestures”, Concurrency and Computation: Practice and Experience, Vol.32, No. 18, pp.e5549, 2020. DOI:10.1002/cpe.5549

[14] Kobayashi K., Oguni T., Nakagawa M., “A Series of PIN/Password Input Methods Resilient to Shoulder Hacking Based on Cognitive Difficulty of Tracing Multiple Key Movements”, IEICE TRANSACTIONS on Information and Systems, Vol.103, No.7, pp.1623-1632, 2020.

[15] Andrew A., Wamema J., “Towards an Improved Framework for E-Risk Management for Digital Financial Services (DFS) in Ugandan Banks: A Case of Bank of Africa (Uganda) Limited”, Journal of Information and Organizational Sciences, Vol.46, No.1, pp.103-127, 2022. DOI:10.31341/jios.46.1.6

[16] Chandrasekran Y., Ramachandiran C. R., Kuruvikulam C. A., “Adoption of Future Banking Using Biometric Technology in Automated Teller Machine (ATM)”, In 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), 2022. DOI:10.1109/ICDCECE53908.2022.9793028

[17] Yadav K., Mattas S., Saini L., Jindal P., “Secure Card-less ATM Transactions”, In 2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA), pp.1-4, 2020. DOI: 10.1109/ICMICA48462.2020.9242713

[18] Groupe Speciale Mobile Association (GSMA). (2018) NFC Functions and Security Certification overview V1.0, 2018.

[19] Guerar Meriem, Benmohammed Mohamed, and Alimi Vincent. (2016 June). Color Wheel Pin: Usable and Resilient ATM Authentication. Journal of High Speed Networks, 22(3), pp. 231-240, 2016. DOI:10.3233/JHS-160545

[20] Smart payment association, “Biometrics in Payment: Breaking down barriers with high value payments”, 2018.

[21] Promontory, “Biometric authentication in payments: Considerations for Policymakers”, 2017.

[22] Chabbi S., Boudour R., Semchedine F., Chefrour D., “Dynamic Array PIN: A novel approach to secure NFC electronic payment between ATM and smartphone”, Information Security Journal: A Global Perspective, Vol.29, No.6, pp.327-340, 2020. DOI:10.1080/19393555.2020.1773583

[23] Kasat O. K., Bhadade U. S., “Revolving flywheel pin entry method to prevent shoulder surfing attacks”, in 2018 3rd IEEE International Conference for Convergence in Technology (I2CT), pp.1-5, 2018. DOI: 10.1109/I2CT.2018.8529758

[24] English R., “Simulating and modelling the effectiveness of graphical password intersection attacks”, Concurrency and Computation: Practice and Experience, Vol.27, No.12, pp.3089-3107, 2015. DOI:10.1002/cpe.3196

[25] Oya S., Troncoso C., Pérez-González F., “Meet the family of statistical disclosure attacks”, in IEEE Global Conference on Signal and Information Processing, pp. 233-236, 2013. DOI:10.1109/GlobalSIP.2013.6736858

Authors’ Profiles

Associate professor Samir Chabbi, University of Souk-Ahras / Department of Mathematics and Informatics, BP 1553 Souk-Ahras 41000, Algeria

Major interests: embedded systems and cyber security in the field of Radio Frequency Identification (RFID) and Near Field Communication (NFC).

Associate Professor Djalel Chefrour, University of Souk-Ahras / Department of Mathematics and Informatics, BP 1553 Souk-Ahras 41000, Algeria

Major interests: Computer networks, embedded and distributed systems: network protocols design, protocol analysis and implementation; software defined networks; network security; network performance measurements; network time synchronization.

Associate Professor Nour El Madhoun, LISITE Laboratory, ISEP, 10 Rue de Vanves, Issy-les-Moulineaux, 92130, France

Major interests: Network security; cryptographic protocols; EMV payment; NFC technology; blockchain and smart-contracts technologies.
.

How to cite this paper: Samir Chabbi, Djalel Chefrour, Nour El Madhoun, "Finding and Mitigating a Vulnerability of the Color Wheel PIN Protocol", International Journal of Computer Network and Information Security(IJCNIS), Vol.16, No.5, pp.113-123, 2024. DOI:10.5815/ijcnis.2024.05.09