An Intersection Attack on the CirclePIN Smartwatch Authentication Mechanism

Djalel Chefrour, Yasser Sedira, Samir Chabbi

***Abstract*—We present a thorough security analysis of a recent smartwatch authentication mechanism called CirclePIN, which was considered resilient to several attacks including shoulder surfing and video recording. This mechanism avoids the direct entry of the personal identification number (PIN) by using consecutive screens of random colors that fool the attacker. We disclose a vulnerability in CirclePIN inherent to the way in which the users match the random colors to their PINs' digits and we illustrate how to exploit it with an intersection attack. This attack uses the information extracted from multiple video recordings of legitimate authentication sessions. We prove that it has a high probability of revealing the user PIN with only three video recordings and always succeeds with five. Our proof is twofold. We formulate the theoretical probability of success for the attack as a function of the number of available video recordings. Then, we validate this formula with a simulation of a large number of attacks to compute their experimental probability of success. In our estimation, manual information extraction takes around one minute per exploitable video recording. So, a complete intersection attack is cost effective in terms of time, as it lasts five minutes or less.**

***Index Terms*—Smartwatch authentication, Intersection attack, CirclePIN vulnerability.**

I. Introduction

Personal Internet of Things (IoT) devices like smartwatches are increasingly present in daily life and integrated into critical infrastructure sectors. For instance, in e-payment and healthcare applications, they can act as a trusted identity proxy for users in addition to their role as sources for sensitive telemetry data. It is therefore necessary that they implement strong and user-friendly authentication mechanisms to protect privacy, despite their hardware constraints [1].

Guerar *et al.* [1] argue that traditional smartwatch authentication mechanisms, such as direct personal identification number (PIN) entry on a numeric touch pad, lack a comprehensive understanding of the vulnerabilities and potential threats they face. As a result, these mechanisms suffer from usability and security issues, making them susceptible to attacks such as shoulder surfing. Indeed, there is evidence backed by real-world data that shoulder surfing is likely to violate user's privacy [2]–[5]. This could happen in different environments such as an individual's home, workplace, public transport, shopping center, or park. So, there is an established need to protect the users of mobile devices like smartwatches against shoulder surfing.

We are interested in this study by one work named CirclePIN, which was proposed by Guerar *et al.* [1] as a non-obtrusive and robust authentication mechanism on smartwatches. Compared to direct entry of a four-digit PIN on a numeric pad, CirclePIN enhances both the usability and security aspects of user authentication by leveraging smartwatch features such as its color touch screen and its crown. To hide the PIN, this mechanism requires that the user identifies its digits indirectly by mapping them to random colors displayed in consecutive screens on the smartwatch. Figure 1 shows an example of these screens for the entry of PIN 7521. The first screen (a) contains a table similar to a numeric pad in which the digits 0 to 9 are associated to random colors. The user memorizes the color beneath the leftmost digit of the PIN (*i.e.*, red and 7 in our example) and moves to the second screen (b). This contains a circle sliced into 10 sectors that are also colored randomly and surrounded by the 10 digits (0 to 9). By employing the smartwatch crown, the user rotates this circle until the color they memorized from the previous screen (*i.e.*, red) matches the next digit of their PIN (*i.e.*, 5) to obtain screen (c). At this stage, the user confirms the entry of the two first PIN digits by clicking the crown (or taping the OK button in the middle of the circle, if the crown acts only as a power button). Next, the user repeats the same process to enter the remaining digits. Namely, the PIN last digits 2 and 1 are matched with the color *pale green* from screens (d) and (e). We note that CirclePIN works only for PIN codes that contain an even number of digits.

Although we find that CirclePIN is easy to use, we discovered that it has a vulnerability which can lead to the disclosure of the user PIN through video-based shoulder surfing. If an attacker disposes of multiple video recordings of valid authentication sessions, they can exploit them using information intersection to reveal the PIN with a high probability of success. In this paper, we introduce the intersection attack on CirclePIN and prove its effectiveness in terms of probability of success and execution time through the following contributions:

- The identification of a threat model in which we list the assumptions necessary for the attack to succeed (in section III-A).
- The exposure of CirclePIN vulnerability to multiple video recordings intersection attack via a step-by-step illustration (in section III-B).
- The theoretical analysis of the attack and the formulation

Manuscript received 25 October 2023; accepted 14 November 2023.

Corresponding author: Djalel Chefrour

The authors are with the Department of Mathematics and Informatics at The University of Souk Ahras, BP 1553, 41000, Algeria (Emails: djalel.chefrour@univ-soukahras.dz, yassersedira20@gmail.com, s.chabi@univ-soukahras.dz

This research was supported by the Algerian Ministry of Higher Education and Scientific Research under Grant Number PRFU C00L07UN410120230002.

Digital Object Identifier 10.1109/JIOT.2023.3333964

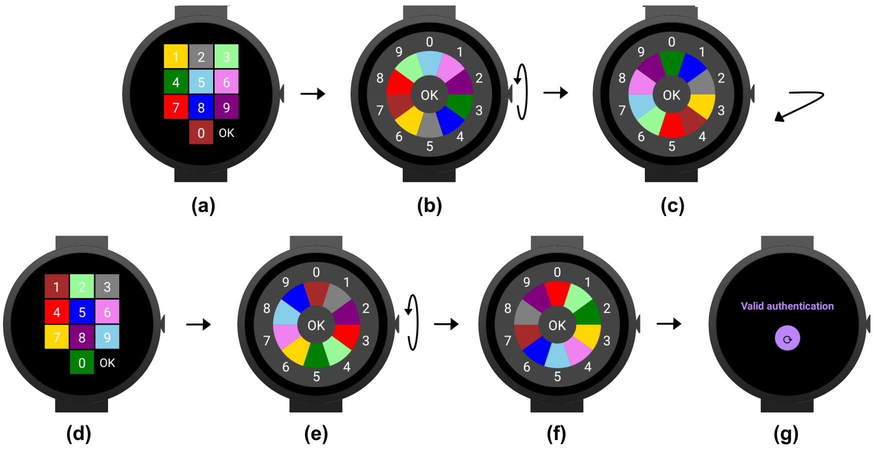

Fig. 1. Screens of the CirclePIN authentication session [1].

of its probability of success as a function of the number of available video recordings (in section IV). Namely, with three exploitable videos the attacker has an 80% chance of success, while with five videos they are assured to always reveal the user PIN.
- The calculation of the corroborating experimental probability of success (in section V). For this purpose, we developed a Python based attack simulator that we made available as open source.
- The estimation of the attack execution time with a breakdown of its manual steps (in section V). This estimation, like the screens used in the attack illustration, is based on our Android Wear implementation of CirclePIN, which is distributed as open source too, for reproducibility.

Before detailing these contributions, we start first by a review of related works in section II. Later, we discuss the results and the limitations of our study (in section VI), then conclude with a summary and a list of perspectives for future work (in section VII).

II. Related works

We start by visiting other works from the authors of CirclePIN that share the same design principle and hence have the same vulnerability to the multiple video recordings intersection attack. Then, we review briefly the most known smartwatch authentication mechanisms and compare them to CirclePIN. Next, we summarize the known attacks against these mechanisms by describing their way of working. Afterwards, we discuss the intersection attacks found in the literature that are like the one we devised.

A. Authentication mechanisms like CirclePIN

CirclePIN is inspired by another mechanism called Color Wheel PIN (CWPIN), which was developed by the same authors for authentication on Automated Teller Machines (ATM) using smartphones [6]. In CWPIN, random color tables are sent from the ATM to the smartphone screen using Near Field Communication (NFC). The color wheel on the other hand is displayed on the ATM screen, where it can be rotated using a touch pad. Users of CWPIN look for the color that corresponds to their first/third PIN digit on the smartphone screen, then map it to the second/fourth PIN digit by rotating the color wheel on the ATM screen. As CWPIN is similar in essence to CirclePIN, it is also vulnerable to the same multiple recordings intersection attack.

2GesturePIN is another smartwatch authentication mechanism from the same authors, that is closely comparable to CirclePIN in terms of design and vulnerability [7]. But, instead of using colored tables, 2GesturePIN employs an outer circle that houses the fixed digits, while a concentric rotating wheel within holds the randomized digits. Users rotate this wheel to align the PIN digits between the two circles. Nevertheless, as attackers can observe from these circles 10 combinations for each two digits, they can uncover the PIN using a few video recordings as we discuss in detail later for CirclePIN.

B. Smartwatch authentication mechanisms

Smartwatch authentication can be achieved via different mechanisms that might be categorized according to multiple criteria such as: complexity, cost, type of inputs, usability and resilience to security attacks. A good authentication mechanism should be secure and easy to use with reduced complexity, cost and number of inputs. However, as shown by the authentication mechanisms reviewed herein, striking the right balance between these criteria is not an easy task. For instance, increasing the complexity of a mechanism or its number of authentication inputs might improve its security but render it less easy to use. For the purpose of this research, we group the most known smartwatch authentication mechanisms in two high level categories according to their resilience (or not) to video-based shoulder surfing.

The first category comprises the traditional techniques based on passwords, PINs, or unlock patterns; and the more recent CirclePIN mechanism. They are easy to use and do not require additional hardware or training. The usability tests reported by CirclePIN authors [1] showed that it is less error prone than pattern-based techniques, which suffer from an average error rate close to 20% as estimated by [8]. Conversely, while these techniques are vulnerable to fortuitous shoulder surfing, CirclePIN is resilient to this particular attack. Nevertheless, a persisting malicious party can still break it by obtaining multiple video recordings of valid authentication sessions, as we illustrate here later with the intersection attack.

The second category includes the authentication mechanisms that are resilient to video-based shoulder surfing. They can be divided further in two non-orthogonal sub-categories according to the number and type of their authentication inputs. On the one hand, we put two-factor authentication (2FA) mechanisms. On the other hand, we group the schemes based on user traits such as biometric, continuous and behavioral characteristics.

2FA mechanisms on smartwatches are more robust than CirclePIN and the traditional mechanisms, as they require a second verification step from the user. This step is performed on a companion application or web browser running in a different device like a smartphone (*e.g.,* OAuth 2.0 support in Android Wearables [9]). We believe that, even if 2FA mechanisms take a little bit more time and involve a second device, this is justifiable given the improved security they provide. They remain resilient to shoulder surfing and to our multiple recordings intersection attack.

The smartwatch authentication techniques of the second sub-category rely on the user's unique behavioral or physiological traits, such as fingerprints, gestures [10] and heart rhythm [11]. These techniques are convenient, fast, provide strong security and do not require users to remember passwords or PIN codes. They are resilient to shoulder surfing, even if it is video-based and carried through multiple recordings. Unlike CirclePIN, they may require specialized hardware and additional training or setup. Furthermore, there is a risk for biometric data to be stolen or copied. Moreover, the collection of this data in the case of continuous authentication can be considered an intrusive monitoring by some users.

C. Attacks on smartwatch authentication mechanisms

The popularity of IoT devices like smartwatches is growing thanks to their rich set of features. As a result, they contain nowadays an increasing amount of sensitive user data like contact details, health records, payment information and position tracking. This makes them a valuable target for attacks that might compromise their authentication mechanisms and put the user privacy at risk. Such attacks include:

Guessing and brute force: An attacker might try to guess the user PIN/password either randomly or using some knowledge about the targeted person (*e.g.,* birth date), assuming it is used as an authentication secret. But the longer the PIN/password is, the less likely it is for the attack to succeed. For instance, with a four-digit PIN, there is only one chance of success out of 10000. To increase this probability, the attacker might resort to brute force by trying all the possible combinations for the user secret. Nonetheless, many authentication mechanisms mitigate such attacks by limiting the number of attempts to three or five.

Shoulder surfing: An attacker observes a user entering their authentication secret to uncover it [2]–[5]. This attack can be carried in person or via a video camera pointed at the interface of the device implementing the authentication (*e.g.,* smartwatch, phone, ATM). The targeted secret can be a: PIN, password, drawing unlock pattern, 2D touch-based signature or even a 3D magnetic-based gestural signature. Sahami Shirazi *et al.* [12] showed that authentication in the latter case is more secure to video-based shoulder surfing than the former ones.

The survey from Eiband *et al.* [3] found that most shoulder surfing in the wild was: opportunistic, with no malicious intent, common among strangers in public transport, involved smartphones in most cases and often went unnoticed. By contrast, the survey from Aviv *et al.* [2] concentrated on shoulder surfing using video recordings of smartphone authentications. Around 1200 participants were assigned the task of viewing recorded videos that showed victims entering PINs and unlock patterns in an attempt to uncover them. The recordings were filmed using GoPro cameras from multiple angles, covering various hand positions of the victims. The rate of successful attacks on six-digit PINs was 10.8% with a single view and 26.5% with multiple views of an authentication video. These rates increased respectively to 34.9% and 56.7% for four-digit PINs, which are easier to uncover because they are shorter. Furthermore, the attacks on unlock patterns were the most successful due to the capture of the victim finger movements and (when enabled) the screen feedback lines connecting the relevant points of the pattern. These empirical results highlight that multiple viewing of an authentication exacerbates its vulnerability to shoulder surfing. The evaluation of our intersection attack, which relies too on multiple video recordings of CirclePIN, confirms this finding.

Biometric spoofing: An attacker creates a copy of the user biometric data, such as a fingerprint, facial image or voice record, then replays it to authenticate [13].

Side-channel attacks: An attacker intercepts and analyzes signals emitted by the smartwatch during authentication to extract the password via a malware injected on the device [14]. For instance, Snoopy is a Trojan that masquerades as a fitness or gaming app and eavesdrops on motion data when users type or swipe their passwords on smartwatches [15]. It periodically uploads this information to the cloud, where it leverages deep neural networks trained with crowd sourced data to infer the user's password.

CirclePIN authors [1] reported that it is resilient to the previous types of attacks. Nevertheless, we show in this work that it is vulnerable to the multiple recordings intersection attack, which can be qualified as an elaborate and more effective form of video-based shoulder surfing.

D. Intersection attacks on authentication mechanisms

English [16] analyzed intersection attacks on recognition-based graphical passwords. These authentication mechanisms present several challenge screens to the user. In every screen there is only one pass image specific to the user that allows access, whilst the other images are random distractions. The attacker can identify the pass images by carrying several authentication attempts then selecting the most frequently shown images. This attack resembles ours because it relies on the fact that the distraction images have a lower probability of reappearing in every screen, as do the random combinations of two digits in CirclePIN screens. Furthermore, the effectiveness of the intersection attack on graphical passwords varies as a function of the number of authentication attempts needed before success. Both, *(i)* the ratio of pass images to distraction ones and *(ii)* the reuse of constant distraction images between sessions, increase the number of required attempts. By contrast, we provide a formal analysis and a simulation-based validation, to show that our intersection attack on CirclePIN has a high probability of success for a low number of exploitable video recordings.

III. Vulnerability of CirclePIN

In this section we first identify the vulnerability we have found in CirclePIN through a detailed threat model. The goal is to emphasize the circumstances where this vulnerability becomes apparent. Then, through an illustrative example, we delve into the detailed steps of how to exploit this vulnerability with an intersection attack, which reveals the user PIN by analyzing the video recordings of few valid authentication sessions.

A. The threat model

We assume that the attacker is able to record exploitable videos of several successful CirclePIN authentications. We mean by *exploitable* a video where the smartwatch screen is clearly visible during the whole authentication session. More precisely, in addition to the final CirclePIN screen that shows a successful authentication, the other screens of interest to the attacker are the ones that display the color table and the color circle at the end of its rotation. These are screens (a), (c), (d) and (f), in the case of the authentication session depicted by figure 1. They constitute the crucial moments in every video recording that the attacker analyzes to derive the user PIN. We note that there is no need to record the smartwatch crown itself when manipulated by the user to carry a successful attack. Moreover, we assume the attacker can get hold of the smartwatch (*e.g.,* steel it) once they derive the PIN from the recorded videos.

CirclePIN authors [1] evaluated its resilience to video-based shoulder surfing in an experimental setup that was considered a "best-case scenario from the attacker's point of view". They video-recorded 19 participants (enrolled as victims) while performing valid CirclePIN authentications in a seated position. The videos were captured with a smartphone camera pointed at the victim's smartwatch with no obstruction. Then, they were given to 15 other participants (cast as attackers) to analyze. None of them managed to correctly guess any PIN, even when they could play, pause, and rewind the recordings at will. Nevertheless, we note that 100% of the videos were exploitable in the sense defined above. The authors reported that they extracted "from each video" the two lists of 10 combinations of digits (that our intersection attack relies on) and gave them to the attackers [1]. This experiment shows that the availability of exploitable videos is highly likely when the recording is carried out with an attacker mindset. Consequently, we assume that the attackers will spare no effort to obtain exploitable videos, which is the worst-case scenario from a security point of view. Furthermore, we qualify these attackers as *agile* to mean that they have practiced the manual steps of the attack a dozen times so they can execute it quickly in a streamlined way.

To obtain an exploitable video, it is crucial to employ one or two wide-angle high-resolution cameras dissimulated in strategic places where the target user is susceptible to run CirclePIN. It is also important to orient the cameras such as they capture the user's wrist through a clear and unobstructed view, and therefore capture their smartwatch screen with the best video quality possible. Examples of such places include, but are not limited to, the ceiling and the walls of the rooms where the users usually sit and carry out an activity that involves a CirclePIN authentication. This could be at their workplace above their desk and/or at a cafeteria or a restaurant they frequent. The attacker could also place the spy cameras in the furniture of these locations, such as lamps, picture frames, etc. The advantage of such static hidden cameras is that they can be equipped with lasting batteries to record videos for long periods of time and to target multiple users. If equipped with motion and/or light detectors, such cameras can be programmed to operate only when users are present and hence cover longer surveillance periods.

A more sophisticated attack scenario might involve the use of drones that can carry video-based shoulder surfing from a certain distance (say a few typical building stories altitude). Such a drone can follow a target user waiting for an opportune moment to record their CirclePIN sessions. It can also zoom and move its camera in multiple degrees of freedom to capture the best possible view of the smartwatch screen. Although this scenario requires a live involvement of the attacker and additional hardware, it offers better chances to capture more targets, compared to the statically concealed cameras. Another more advanced attack scenario could involve infecting the smartwatch with malware that captures videos of its screen. This can be accomplished with a side-channel attack that exploits other vulnerabilities in the smartwatch.

Once multiple video recordings of successful authentication sessions are acquired, the attacker analyzes them to extract the user PIN. We illustrate how this is done with three recordings in the next section. However, it should be noted that this attack assumes that the user does not modify their PIN during the whole period of the multiple recordings.

B. Intersection attack on CirclePIN

We illustrate herein a successful intersection attack on CirclePIN that uses three video recordings of valid authentications. For this purpose, we describe the actions carried out by the attacker starting from the example of figure 1. This contains the screens of crucial moments extracted from the first video recording of a user with the PIN 7521. By picking the digits associated to each color in the screens (a) and (c), the attacker identifies 10 possible combinations for the PIN first two digits. Likewise, screens (d) and (f) give up 10 other combinations for the PIN last two digits. These two sets of 10 combinations are shown in Tables I and II.

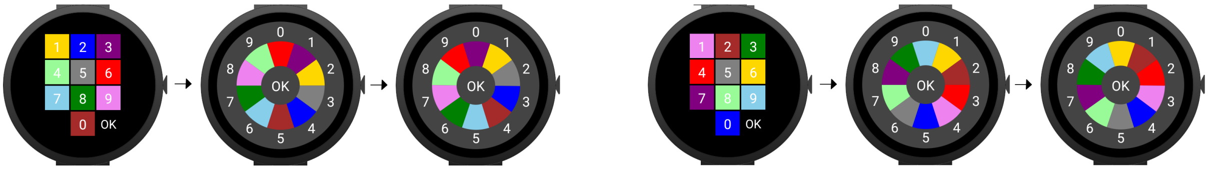

Fig. 2. Screens of the second CirclePIN authentication session.

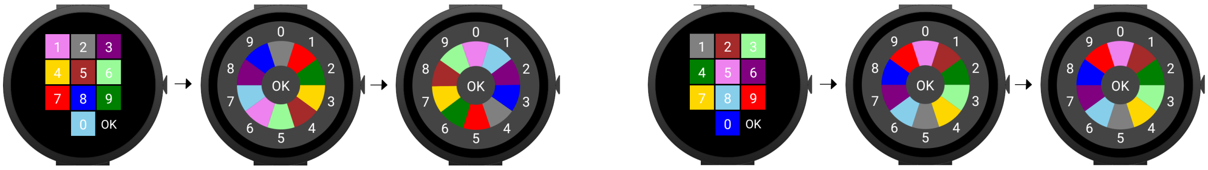

Fig. 3. Screens of the third CirclePIN authentication session.

TABLE I
POSSIBLE COMBINATIONS OF THE PIN FIRST TWO DIGITS FROM THE FIRST CIRCLEPIN SESSION.

1st digit	0	1	2	3	4	5	6	7	8	9
Color	🟫	🟨	▨	🟩	🟩	🟦	🟪	🟥	🟦	🟪
2nd digit	4	3	2	6	0	7	8	5	1	9

TABLE II
POSSIBLE COMBINATIONS OF THE PIN LAST TWO DIGITS FROM THE FIRST CIRCLEPIN SESSION.

3rd digit	0	1	2	3	4	5	6	7	8	9
Color	🟩	🟫	🟩	▨	🟥	🟦	🟪	🟨	🟪	🟦
4th digit	2	7	1	8	0	6	4	3	9	5

TABLE III
POSSIBLE COMBINATIONS OF THE PIN FIRST TWO DIGITS FROM THE SECOND CIRCLEPIN SESSION.

1st digit	0	1	2	3	4	5	6	7	8	9
Color	🟫	🟨	🟦	🟪	🟩	▨	🟥	🟦	🟩	🟪
2nd digit	4	1	3	0	8	2	9	5	6	7

TABLE IV
POSSIBLE COMBINATIONS OF THE PIN LAST TWO DIGITS FROM THE SECOND CIRCLEPIN SESSION.

3rd digit	0	1	2	3	4	5	6	7	8	9
Color	🟦	🟪	🟫	🟩	🟥	▨	🟨	🟪	🟩	🟦
4th digit	4	3	1	8	2	5	0	7	6	9

In each set of 10 combinations, only one element belongs to the PIN and is present in every valid authentication session. The remaining nine combinations do not belong to the PIN, because each digit from 0 to 9 is not repeated either in the color table or in the following color circle. Moreover, there is a high chance that these nine combinations will not reappear in consecutive authentication sessions, as they are generated randomly by CirclePIN. Therefore, by extracting, then intersecting a few sets of 10 combinations of two digits for each half of the PIN, the attacker can rule out the random ones and disclose the PIN digits as soon as the intersection yields one and only one element.

At the end of the first video recording analysis, the attacker obtains 10x10 possible combinations of four digits. Only one of them is the correct PIN. To eliminate the remaining 99 random combinations, the attacker must extract more sets from additional video recordings where it is highly unlikely that these 99 combinations will recur. Hence, the attack proceeds with the second video recording from which the crucial screens are extracted and shown in figure 2. Likewise, by comparing the relevant screens the attacker obtains two other sets of possible combinations for the two halves of the PIN. These are shown in tables III and IV.

After analyzing the second authentication session, the attacker can drastically reduce the possible combinations of digits that constitute the PIN by intersecting Table I with III and Table II with IV, while omitting the colored rows. The result is shown in equations 1 and 2, which give only two possible combinations for each half of the PIN. This means the attacker disposes of four possible combinations for the PIN (*i.e.*, $0421, 0438, 7521$ and 7538) and can already try them if they manage to get hold of the user smartwatch. As CirclePIN allows three authentication attempts before locking the device, then there are already three chances out of four for the attacker to authenticate successfully. Nevertheless, to get the correct PIN, the attack carries on with a third video recording.

$$\cap Tables(I, III) = \{04, 75\} \tag{1}$$

$$\cap Tables(II, IV) = \{21, 38\} \tag{2}$$

TABLE V
POSSIBLE COMBINATIONS OF THE PIN FIRST TWO DIGITS FROM THE THIRD CIRCLEPIN SESSION.

1st digit	0	1	2	3	4	5	6	7	8	9
color	🟦	🟪	▨	🟪	🟨	🟫	🟩	🟥	🟦	🟩
2nd digit	1	0	4	2	7	8	9	5	3	6

Figure 3 depicts the relevant screens extracted from the analysis of the third CirclePIN recording; whereas Tables V and VI illustrate the possible combinations of digits obtained from these screens. We note that the two rightmost circles in figure 3 are the same, because the color brown that matches digit 2 to 1 is already in place, so the user does not rotate the circle but just taps OK. At this stage the attacker can extend

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3333964

TABLE VI
POSSIBLE COMBINATIONS OF THE PIN LAST TWO DIGITS FROM THE THIRD CIRCLEPIN SESSION.

3rd digit	0	1	2	3	4	5	6	7	8	9
Color	🟦	⬛	🟫	🟩	🟩	🟪	🟪	🟨	🟦	🟥
4th digit	8	5	1	3	2	0	7	4	6	9

the previous intersection operation to these last two tables to reveal the correct halves of the PIN, 75 and 21, as show in equations 3 and 4.

$$\cap Tables(I, III, V) = \{75\} \tag{3}$$

$$\cap Tables(II, IV, VI) = \{21\} \tag{4}$$

IV. THEORETICAL ANALYSIS OF THE ATTACK

This section considers the general case of the intersection attack for any value of the user PIN and any number N of exploitable video recordings. In particular, we formulate $P(N)$ the theoretical probability of success for this attack and prove mathematically that it is very high, *i.e.*, 0.80, given only three exploitable video recordings; and is close to 1 with four videos. Notably, the same logic holds for CWPIN and 2GesturePIN which share the same design principle with CirclePIN. For brevity, we analyze only two PIN digits, as our attack independently reveals each half of a four-digit PIN. Next, we provide a comparative analysis of this attack with simple guessing, to show the added value brought by information intersection. Afterwards, we extend our analysis to PINs that are longer than four digits. To be precise, we use in this section the word *permutation* instead of *combination* of digits as their order matters. In the other sections, we stick to the term *combination* as it is more popular and is the one used in the CirclePIN work [1].

The probability of success for the intersection attack is determined by the likelihood that the intersection of sets, each representing possible permutations of two digits extracted from N sessions, yields only those permutations that belong to the PIN. This is equivalent to saying that none of the nine random permutations will appear in every one of the N sessions. So, we need to express first the probability R of a random permutation of two digits showing up in one session. Then, we analyze the case of N successive sessions where each random permutation must disappear at least once, to be excluded by the intersection operation.

If we were choosing a partial permutation with repetition of two digits out of 10, then the sample size would be equal to 100 and the probability of this choice would be $\frac{1}{100} = 0.01$. However, thanks to the design restrictions inherent to CirclePIN, the first digit in the permutation is always fixed in the color table. This is shown in the crucial smartwatch screens and in the first rows of the Tables I to VI extracted by the attacker. Furthermore, the second digit in the permutation is drawn randomly in a uniform way without repetition from a set of 9 possibilities only. This is apparent in the second rows of the mentioned tables. The tenth possible digit is also fixed as it belongs to the PIN. It will always show up at the same position in the tables extracted from all the N sessions. For instance, this is digit 5 in the second rows of Tables I, III and V. So $R = \frac{1}{9} = 0.11$.

We consider now the case of N successive CirclePIN authentication sessions. The first session contains nine random permutations of two digits where each one appeared with probability R. So, the probability that one of these permutations will be present in all the remaining $(N-1)$ sessions is actually $R^{(N-1)}$, as each session is independent from the others. Conversely, the probability that a random permutation is absent from at least one of these $(N-1)$ sessions is therefore equal to $1 - R^{(N-1)}$ (i.e., it is the complementary event of the previous statement). By generalizing to all the nine random permutations that showed up in the first session, we obtain the probability that each one of them must disappear at least once in the subsequent $(N-1)$ sessions. In other terms: $(1 - R^{(N-1)})^9 = (1 - 0.11^{(N-1)})^9$. Finally, as the same analysis applies to the two last digits of the PIN independently from the first two, we must square the latter probability to obtain $P(N)$ as expressed in the formula 5.

$$P(N) = (1 - 0.11^{(N-1)})^{18} \tag{5}$$

Concerning the case where only one video recording is available per PIN, we note that our attack does not apply as there are no multiple sets of permutations on which to perform the intersection. So, formula 5 is not relevant (NR) for $N = 1$. Instead, an attacker has merely one chance out of 10 to guess the first PIN digit and figure out the second one via color matching. The same thing holds for the third and fourth PIN digits. So, if we consider both halves of a four-digit PIN, the probability of success for this guessing attack is $(\frac{1}{10})^2 = 0.01$. Let us call it G and generalize it to N multiple videos. We can express this probability with formula 6, because the attacker can guess the PIN with the first video, or the second one, or the third, etc.

$$G(N) = \Sigma_{i=1}^{N} (\frac{1}{10})^2 = N \times 0.01 \tag{6}$$

TABLE VII
SUCCESS PROBABILITIES OF THE INTERSECTION AND GUESSING ATTACKS FOR AN INCREASING NUMBER OF VIDEO RECORDINGS.

N	1	2	3	4	5
$P(N)$	NR	0.12	0.80	0.98	1.0
$G(N)$	0.01	0.02	0.03	0.04	0.05

Table VII computes the success probabilities for the intersection and guessing attacks (P and G) rounded to two decimal places. Both are expressed as functions of the number N of exploitable video recordings. We can see that P converges to one quickly as N reaches five. In other words, an attacker is guaranteed to disclose the user PIN with the intersection of the information extracted from only five video recordings. In most practical cases, three video recordings are sufficient to yield one or two possible PINs that can be tried on a CirclePIN smartwatch, once the attacker gets hold of it. By contrast, G is drastically low for the same input and increases arithmetically, whereas P grows exponentially. Hence, the intersection attack is way more effective than simple guessing.

Finally, we extend the analysis above to the cases where an application implements CirclePIN with a longer user secret. Let us denote L as the length of the PIN in such instances, which is intentionally designed to be an even number of digits. In this scenario, the probability of a successful attack is the new function $P(N, L)$ expressed by formula 7. Table VIII shows how $P(N, L)$ evolves when the PIN length increases to an extreme case of 20 digits. Notably, while the probability of success declines gradually when only three recordings are available, it still remains quite high with four recordings and nearly perfect with five.

$$P(N, L) = (1 - 0.11^{(N-1)})^{(9*\frac{L}{2})} \tag{7}$$

TABLE VIII
SUCCESS PROBABILITY OF THE INTERSECTION ATTACK AS A FUNCTION OF THE NUMBER OF RECORDINGS N AND THE PIN LENGTH L.

P(N,L)	N=2	N=3	N=4	N=5
L=4	0.12	0.8	0.98	1.0
L=6	0.04	0.72	0.96	1.0
L=8	0.01	0.64	0.95	0.99
L=10	0.0	0.57	0.94	0.99
L=12	0.0	0.51	0.93	0.99
L=14	0.0	0.46	0.92	0.99
L=16	0.0	0.41	0.91	0.99
L=18	0.0	0.37	0.89	0.99
L=20	0.0	0.33	0.88	0.99

V. Experimental validation

To assess the effectiveness of the intersection attack experimentally, we have developed an Android Wear version of CirclePIN in Kotlin and an attack simulator in Python. We made both tools available as free and open source software[1]. They use four-digit PINs by default, as in the CirclePIN work [1]. Figures 1, 2 and 3 are screen captures from the execution of CirclePIN in an Android smartwatch emulator. The crown gestures are mimicked by the emulator Rotary Input feature.

The Python simulator has a double role. First, it serves as a graphical illustrator of the attack as shown by the example of figure 4. More precisely, it depicts the crucial CirclePIN screens, and the combinations of PIN digits derived from them, as if they were extracted manually from the video recording by the attacker. Second, the simulator runs many intersection attacks starting after the extraction of the crucial screens, to calculate the experimental probability of success.

To execute one attack, the simulator takes as input parameters the available number of video recordings N and the targeted PIN, and proceeds as follows:

1) For each recording, the simulator mimics a legitimate authentication session, then extracts the relevant information from it, which is subsequently used for the intersection.
2) For each pair of consecutive digits in the PIN, it generates two random sets containing 10 colors each. These are associated to the 10 digits of CirclePIN's table and circle, respectively.
3) Next, the simulator finds the color associated to the first digit in the table, then rotates the circle until this color aligns with the second digit, mimicking the behavior of a legitimate user. This concludes the simulation of the authentication for the digits pair and the attack commences.
4) The simulator derives and stores the set of 10 possible combinations of two digits from the states of the table and the circle. Starting from the second recording, it intersects this derived set with those obtained from previous recordings.
5) The attack persists until the intersection produces one and only one combination for each pair, indicating a success. Otherwise, if the intersection yields no unique result across all the processed recordings, the attack is considered unsuccessful.

The simulator repeats the intersection attack 1000 times to measure the experimental probability of success, given a number of available video recordings. We note that each attack is independent from the others with its own simulated recordings. So, a new PIN is generated randomly for each attack and the output result (*i.e.*, success or failure) is saved. As the number of attacks increases, we calculate at each increment the experimental probability by dividing the number of successes by the number of attacks carried so far. The latter corresponds to the data points on the horizontal axes of the plots in figure 5. There are four plots because we ran the loop of 1000 attacks from scratch for every value of N between two and five. The vertical axes indicate the experimental probabilities we obtained. They match and therefore validate the theoretical probabilities formulated earlier.

We note that the variability at the beginning of the curves in figure 5 is due to the limit of the sample size (*i.e.*, the number of attacks) at the start of the simulation. As established by the statistical Law of Large Numbers, limited sample sizes tend to produce more fluctuation in the observed events because of their inherent randomness. Whereas, with a large number of attacks the experimental probability converges to the population mean (expressed here by the theoretical formula) and stabilizes.

The whole simulation includes an outer loop over N with the range $[2..5]$, a middle loop that iterates 1000 attacks, and an inner loop with the range $[1..N]$, which simulates the retrieval and the intersection of information from the recordings. In total, the simulation runs for about two seconds on a laptop equipped with a 2.80 GHz Intel Core i7-1165G7 CPU. However, this does not include the time needed to extract the crucial information from the recordings manually. This time can be broken down per recording as follows:

1) The duration of the user authentication, as the attacker needs to watch it till the screen that indicates success. Assuming the video playback happens at normal speed and the attack starts from the beginning of the authentication session, this step lasts around four seconds which is the average CirclePIN authentication time reported in

[1] https://github.com/cdjalel/CirclePIN/

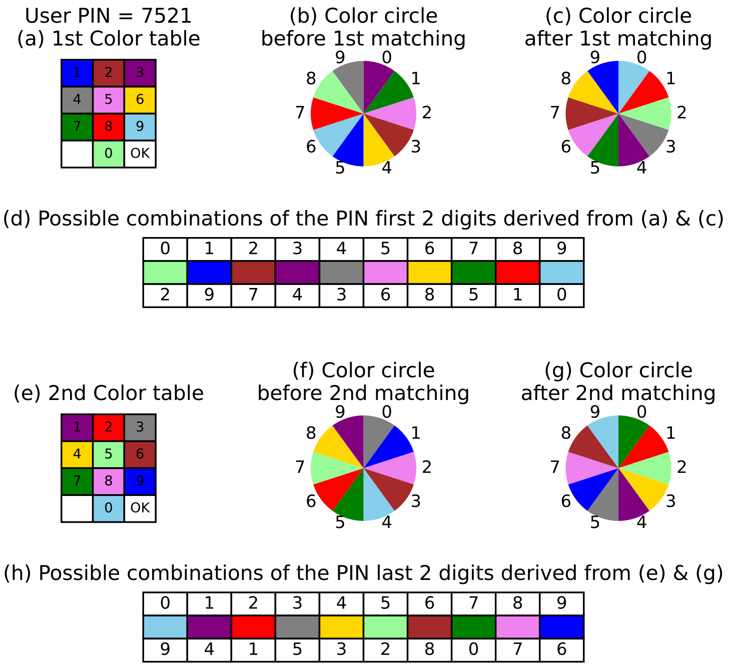
User PIN = 7521
(a) 1st Color table
1 2 3 4 5 6 7 8 9 0 OK
(b) Color circle before 1st matching
(c) Color circle after 1st matching
(d) Possible combinations of the PIN first 2 digits derived from (a) & (c)
0 1 2 3 4 5 6 7 8 9
2 9 7 4 3 6 8 5 1 0
(e) 2nd Color table
(f) Color circle before 2nd matching
(g) Color circle after 2nd matching
(h) Possible combinations of the PIN last 2 digits derived from (e) & (g)
0 1 2 3 4 5 6 7 8 9
9 4 1 5 3 2 8 0 7 6

Fig. 4. Graphical simulation of a CirclePIN authentication session.

[1].

2) The time needed to extract the four crucial screens of CirclePIN (*e.g.*, (a), (c), (d) and (f) in figure 1). On the one hand, the screens containing the color tables (e.g., (a) and (d)) are the easier to fetch as they do not change when they are displayed. Their extraction consumes two seconds per screen based on our experience with the intersection attack. That is to pause, take a window capture and save it. On the other hand, the screens containing the color circles after rotation (*e.g.*, (c) and (f)), require a bit more attention from the attacker to make sure the rotation really ended. This is accomplished by waiting for the display of the consecutive screens (*e.g.*, (d) and (g)), then pausing, rewinding the video for few frames, capturing and saving. The extra step of rewinding consumes as little as one second when mapped to a key press in the video player.
3) The period required to find the digits combinations using color matching between the relevant screens. The attacker streamlines this process by: filling the first digits (*i.e.*, the first rows of tables I and II) in advance, reading the cells of each color table in sequence, looking up each color in the circle and retrieving its digit. This takes around two seconds per color, which we multiply by 10 to count the number of combinations, then by two to cover both halves of a four-digit PIN.
4) The time necessary to intersect the sets extracted from three, four or five recordings. This time is negligible as the intersection operation is easily automated in a spreadsheet or a script as done in our attack simulator.

To sum up, an agile attacker can manually extract the information (*i.e.*, the two sets of 10 possible combinations of two digits) from one video recording in roughly $4+2*(2+3+2*10) = 54$ seconds. We round this figure up to one minute to cover for some slack and the time spent switching to the next video recording. So, if the attack succeeds after exploiting three videos, which is very likely, then it takes around three minutes. Otherwise, if it requires five videos in the worst case, then it lasts five minutes only.

VI. Discussion

We reported in section II-C the survey from Eiband *et al.* [3] which showed that shoulder surfing in the wild, when carried out by direct observation, is mostly opportunistic. However, we have not found so far a similar study about video-based shoulder surfing in the real-world. Such a study would reveal the ratio of real-world exploitable videos among all recorded ones. To the contrary, several experiments about the resilience of authentications mechanisms to video-based shoulder surfing were run in controlled environments [1], [2], [12], [17]. The recording settings were prepared in a way favorable to the attacker, to assess the worst-case scenario from a security point of view. In turn, this gives a high ratio of exploitable videos. For instance, we outlined in section III-A that the videos recorded for CirclePIN evaluation were 100% exploitable. For this reason, we are convinced that carrying out a similar controlled experiment for the intersection attack will yield a high ratio too. Such ratios do not necessarily reflect real-world scenarios, which might be characterized by poor recording conditions. In fact, the clarity of a video recording can vary widely depending on many factors such as: camera distance, filming angle, lighting, smartwatch screen

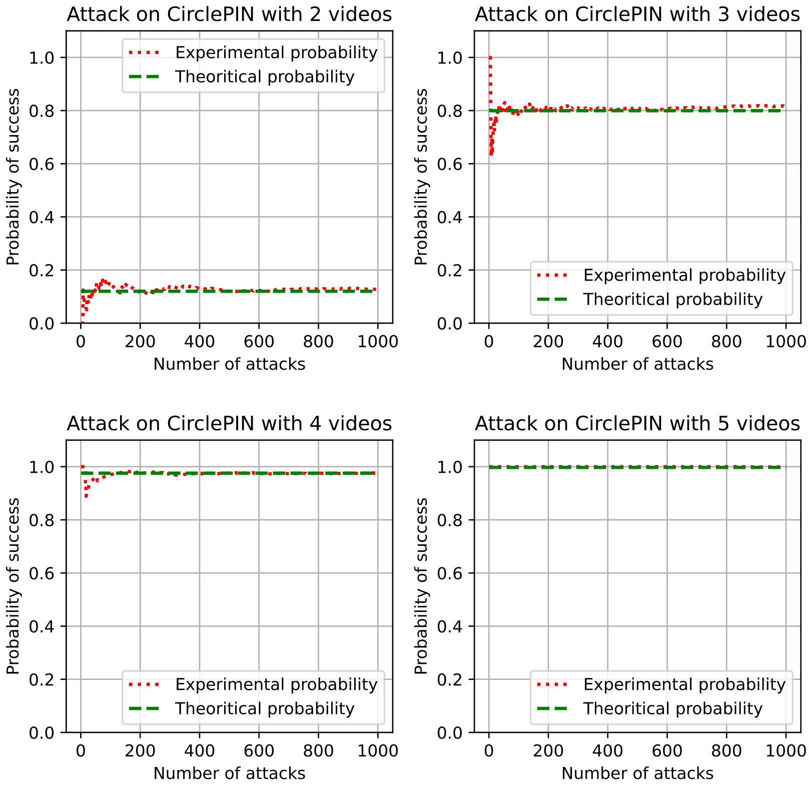

Fig. 5. The intersection attack probability of success as a function of available video recordings.

size, target's posture and movements, etc. Therefore, we expect the ratio of exploitable videos to be highly variable too.

Aviv *et al.* [2] established that the recording conditions undoubtedly have an impact on shoulder surfing. Namely, partial or poor video capture will make shoulder surfing unreliable. So, we can deduce that it will influence the intersection attack effectiveness too. Nonetheless, we believe this will not dissuade a persisting attacker who assesses that a specific target holds high value from optimizing their recording setup. In other terms, the intersection attack should be targeted and well prepared to ensure its success. Moreover, the quality of recorded videos can be improved using dedicated editing tools. Also, the automation of information extraction from these videos (for instance, through pattern recognition) will minimize the attack time. Consequently, this will allow the processing of a higher number of videos and compensate for a possible low ratio of the exploitable ones. It will also enable the targeting of a larger number of victims at low cost.

Assessing the ratio of real-world exploitable videos requires a long empirical study. This study should cover various usage scenarios with a representative number of participants. In our opinion, such a study deserves its own publication and is beyond the scope of this work. As CirclePIN is not deployed in any commercial products, to the best of our knowledge, we think this study is not justifiable for the time being. Furthermore, we contend that the demonstrated high probability of success for the intersection attack will likely deter the adoption of CirclePIN, unless effective mitigation measures are proposed. Despite our efforts to find such measures, a solution confined to CirclePIN itself has so far eluded us. We believe this is because the vulnerability we discovered is inherent in the design of this mechanism. Removing this vulnerability while preserving the main concept of digit-matching with random colors proves to be challenging.

Lastly, we think that 2FA mechanisms, such as OAuth 2.0 [18], offer better security for IoT devices. This is due to their hardware restrictions. In particular, smartwatches are not designed as standalone devices; instead, they work in tandem with smartphones. In this context, CirclePIN and similar mechanisms could be coupled with 2FA to make them resilient to many forms of video-based shoulder surfing, including our intersection attack. A victim using 2FA can deny access to their smartwatch from a second device, when notified about the attacker's attempt to authenticate with a disclosed PIN.

VII. Conclusion

In this work we disclosed a vulnerability in the recent smartwatch authentication mechanism CirclePIN. This vulnerability arises from the fact that the combinations of two digits belonging to the PIN are consistently present in valid authentication sessions, while the remaining combinations are unlikely to repeat in successive sessions due to their randomness. We devised and illustrated an intersection attack that exploits this fact by using multiple video recordings.

We also proved formally that the probability of the attack success converges quickly to one for a small number of video recordings. In practice, only three recordings are required for the attacker to uncover the correct user PIN. Furthermore, we confirmed the validity of the theoretical success rate of our attack by developing a Python simulator that computes the corresponding experimental probability. It does so by executing a large number of attacks for each possible value of N (the number of available video recordings) between two and five. The attack is also cost effective in terms of time, as it lasts five minutes or less when executed mostly manually.

In terms of limitations, our work did not assess empirically the ratio of exploitable videos from real-world recordings. We think this is not a high priority task; especially that CirclePIN is not deployed commercially as far as we know. Nevertheless, to surpass the difficulty of obtaining real-world video captures of CirclePIN sessions, one can resort to using virtual-reality environments to test the attack. For instance, this was performed recently by [19] to test shoulder surfing. Another perspective for our work is to devise a mitigation of the CirclePIN vulnerability that prevents the multiple recordings intersection attack (other than 2FA).

Finally, with regard to the automatic extraction of the relevant attack information from exploitable video recordings, artificial intelligence techniques such as pattern recognition with deep learning constitute a promising idea. Independently of CirclePIN, such techniques can also serve to test various video-based attacks on different authentications schemes, such as traditional password/PIN mechanisms. So far, we have found no works in the literature that investigate this possibility.

References

[1] M. Guerar, L. Verderame, A. Merlo, F. Palmieri, M. Migliardi, and L. Vallerini, "CirclePIN: A Novel Authentication Mechanism for Smartwatches to Prevent Unauthorized Access to IoT Devices," *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 3, mar 2020.

[2] A. J. Aviv, J. T. Davin, F. Wolf, and R. Kuber, "Towards Baselines for Shoulder Surfing on Mobile Authentication," in *Proceedings of the 33rd Annual Computer Security Applications Conference*. Association for Computing Machinery, 2017, p. 486–498.

[3] M. Eiband, M. Khamis, E. Von Zezschwitz, H. Hussmann, and F. Alt, "Understanding shoulder surfing in the wild: Stories from users and observers," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 4254–4265.

[4] H. Farzand, K. Marky, and M. Khamis, "Shoulder Surfing through the Social Lens: A Longitudinal Investigation & Insights from an Exploratory Diary Study," in *Proceedings of the 2022 European Symposium on Usable Security*. Association for Computing Machinery, 2022, p. 85–97.

[5] S. Schneegaß, A. Saad, R. Heger, S. Delgado Rodriguez, R. Poguntke, and F. Alt, "An Investigation of Shoulder Surfing Attacks on Touch-Based Unlock Events," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, 09 2022.

[6] M. Guerar, M. Benmohammed, and V. Alimi, "Color Wheel PIN: Usable and resilient ATM authentication," *Journal of High Speed Networks*, vol. 22, no. 3, pp. 231–240, 2016.

[7] M. Guerar, M. Migliardi, F. Palmieri, L. Verderame, and A. Merlo, "Securing PIN-Based Authentication in Smartwatches With Just Two Gestures," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 18, 2020.

[8] T. Nguyen and N. Memon, "Smartwatches Locking Methods: A Comparative Study," in *13th Symposium on Usable Privacy and Security (SOUPS 2017)*. Santa Clara, CA: USENIX Association, Jul. 2017.

[9] Google, "Authentication on wearables — Android Developers — developer.android.com," https://developer.android.com/training/wearables/apps/auth-wear, [Accessed 21-Jun-2023].

[10] A. Lewis, Y. Li, and M. Xie, "Real time motion-based authentication for smartwatch," in *2016 IEEE Conference on Communications and Network Security (CNS)*, 2016, pp. 380–381.

[11] Nymi, "Wearable identity purpose-built for the workplace — nymi.com," https://www.nymi.com/nymi-band, [Accessed 21-Jun-2023].

[12] A. Sahami Shirazi, P. Moghadam, H. Ketabdar, and A. Schmidt, "Assessing the Vulnerability of Magnetic Gestural Authentication to Video-Based Shoulder Surfing Attacks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. New York, NY, USA: Association for Computing Machinery, 2012, pp. 2045–2048.

[13] A. Hamed and A. A. Khalek, "Acoustic Attacks in the Era of IoT - A Survey," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 2019, pp. 855–858.

[14] A. Maiti, M. Jadliwala, J. He, and I. Bilogrevic, "Side-Channel Inference Attacks on Mobile Keypads Using Smartwatches," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2180–2194, 2018.

[15] C. X. Lu, B. Du, H. Wen, S. Wang, A. Markham, I. Martinovic, Y. Shen, and N. Trigoni, "Snoopy: Sniffing Your Smartwatch Passwords via Deep Sequence Learning," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, jan 2018.

[16] R. English, "Simulating and modelling the effectiveness of graphical password intersection attacks," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 12, pp. 3089–3107, 2015.

[17] L. Bošnjak and B. Brumen, "Shoulder surfing experiments: A systematic literature review," *Computers & Security*, vol. 99, p. 102023, 2020.

[18] D. Hardt, "The OAuth 2.0 Authorization Framework," RFC 6749, Oct. 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6749

[19] Y. Abdrabou, S. R. Rivu, T. Ammar, J. Liebers, A. Saad, C. Liebers, U. Gruenefeld, P. Knierim, M. Khamis, V. Makela *et al.*, "Understanding Shoulder Surfer Behavior and Attack Patterns Using Virtual Reality," in *Proceedings of the 2022 International Conference on Advanced Visual Interfaces*, 2022, pp. 1–9.

Biography

Djalel Chefrour received the Engineering degree in computer science from the university of Annaba, Algeria, in 1999; the M.S. degree in computer systems and networks from the university of Grenoble, France, in 2001; the Ph.D. degree in computer science from the university of Rennes, France, in 2005; and the Habilitation to direct research in computer science from the University of Souk Ahras, Algeria, in 2021. From 2004 to 2013, he was a Research and Development Engineer in the area of computer networks and embedded software with global leaders in the telecommunication and consumer electronics industries (Philips, Alcatel-Lucent, Thomson Telecom, Technicolor). Since 2013, he has been an Assistant, then an Associate Professor with the Mathematics and Informatics Department in the University of Souk Ahras, Algeria. He was also the Head of this department between 2021 and 2023. His research interests include the Internet of Things, Networks measurement, Software Defined Networks, Network time synchronization, Embedded and real-time systems, and Cybersecurity.

Yasser Sedira graduated from the University of Souk-Ahras with a Master's degree in Software Engineering in 2023. He received the bachelor degree in computer science from the same university in 2021. His interests include the Internet of Things, Software engineering and Mobile computing.

Samir Chabbi received the Engineering degree in computer science in 1996; the Magister degree in computer hardware and software in 2015 and the Ph.D. degree in informatics in 2021. From 1999 to 2015, he exercised the function of head of the staff service within a public economic company called cooperative of cereals and pulses. Since 2015, he has been an Assistant, with the Mathematics and Informatics Department in the University of Ghardaia, then in the University of Souk Ahras, Algeria. He is actually an Associate Professor. His research interests include The Internet of Things, Cybersecurity, and Near Field Communications (NFC).