CirclePIN: a novel authentication mechanism for smartwatches to prevent unauthorized access to IoT devices

MERIEM GUERAR, LUCA VERDERAME, and ALESSIO MERLO*, DIBRIS - University of Genoa
MAURO MIGLIARDI†, DEI - University of Padua
FRANCESCO PALMIERI‡, University of Salerno
LUCA VALLERINI*†, DEI - University of Padua

In the last months, the market of personal wearable devices is booming significantly, and, in particular, smartwatches are starting to assume a fundamental role in the Bring Your Own Device (BYOD) arena as well as in the more general Internet of Things (IoT) ecosystem, by acting both as sensitive data sources and as user identity proxies. These new roles, complementing the more traditional personal assistance and telemetry/tracking ones, open new perspectives in their integration in complex IoT-based critical infrastructures such as e-payment, healthcare monitoring and emergency systems, as well as in their usage as remote control facilities in smart services. Users can access their IoT devices at any time from any place through smartwatches. We argue that this new scenario calls for a strengthened and more resilient authentication of users on these devices, despite their limitations in terms of dimensions and hardware constraints that may considerably affect the usability of security mechanisms. In this paper we present an innovative authentication scheme targeted at smartwatches, namely CirclePIN, that provides both resilience to most common attacks and a high level of usability in tests with real users.

Additional Key Words and Phrases: Smartwatch, Mobile Authentication, Android Security, Circle-PIN

ACM Reference Format:
Meriem Guerar, Luca Verderame, Alessio Merlo, Mauro Migliardi, Francesco Palmieri, and Luca Vallerini. 2018. CirclePIN: a novel authentication mechanism for smartwatches to prevent unauthorized access to IoT devices. 1, 1 (May 2018), 19 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The proliferation of IoT devices tied into critical sectors such as finance, transportation and health-care is changing the perspective on what constitutes a critical infrastructure and on the importance of securing personal devices. As an example, consider self-driving cars endowed with a navigation system based on real-time traffic information: traffic information is critical to the correct functioning of routing algorithms and incorrect information can lead to catastrophic jams. In an IoT-based system where the information is obtained by crowd-sourcing data from a plethora of devices, strong authentication and attestation of the devices becomes paramount. Indeed, whether such a strong guarantee is

Authors' addresses: Meriem Guerar, meriem.guerar@unige.it; Luca Verderame, luca.verderame@unige.it; Alessio Merlo, DIBRIS - University of Genoa, Via Dodecaneso, 35, Genoa, Italy, I-16146; Mauro Migliardi, mauro.migliardi@unipd.it, luca.vallerini@studenti.unipd.it, DEI - University of Padua, Via Gradenigo 6, Padua, Italy, I-35131; Francesco Palmieri, fpalmieri@unisa.it, University of Salerno, Salerno, Italy; Luca Vallerini, alessio@dibris.unige.it, luca.vallerini@studenti.unipd.it, DEI - University of Padua, Via Gradenigo 6, Padua, Italy, I-35131.

Manuscript submitted to ACM

not present, it would be possible to mount a sybil-like attack with many devices unrelated to real users, to force onto the system the false belief that a critical artery is clogged and traffic needs to be rerouted to secondary, less capable routes; this, in turn, would actually lead into locking the mobility in a zone and creating traffic problems where none existed.

While the case of securing the communication infrastructure from Denial of Service (DoS) attacks [1] or sanitizing it in an energy sustainable way [2] as well as the reliable attestation of fully machine-controlled devices require their own solution and the latter is one of the cornerstones of Trusted Computing [3, 4], in this paper we will focus on the human factor, i.e., the case in which, as in the navigation system example above, personal devices either play a core role in the Observe, Orient, Decide and Act loop that controls a critical infrastructure, or they act as a sort of trusted identity proxy for the user when interacting with the IoT system [5]. We argue that, in both of the latter situations, it is fundamental to have a way to authenticate the user in a strong and as unobtrusive as possible way. Furthermore, simply filtering out non-human access will either introduce an additional step to the authentication process [6] or require the user to interact with the device [7] in a way that, given the smartwatch form factor, is far from natural and convenient. Indeed, users often become the weakest link in the cybersecurity chain mostly when security mechanisms involve procedures that are not perceived commensurate to the security risk.

Today, as the IoT is involved in almost all the fields of our daily life (e.g., smart home, digital healthcare, smart grid, smart city), the integration of our personal devices into it is rapidly growing. Among these personal devices rapidly diffusing into the IoT ecosystem, smartwatches are one of the most popular and widely used due to their convenience and capabilities. These devices are now assuming a fundamental role in the BYOD arena, since their functionalities are no more limited to gaming, health and fitness tracking, and they can be used for making payments (see the recent Apple and Google Pay examples) and even controlling access to any devices in home appliances like smart TVs, smart refrigerators, thermostats and smart door locks, to name a few. However, these benefits come with increasing risks in terms of security and privacy. In addition, smartwatches can now operate independently or coupled with smartphones, and are extremely promising for collecting sensitive personal data unobtrusively and continuously (from built-in sensors such as accelerometer, gyroscope and biosensors), as well as for acting as trusted proxies for the user's identity. Clearly, due the critical security implications of both the above activities, this can be done if and only if reliable and usable authentication mechanisms are available on these devices.

Unfortunately, introducing effective and usable security enforcement practices into smartwatches may reveal itself as quite complex a task, or even a real nightmare. First of all, like many other wearable objects, smartwatches have not been designed as standalone devices, with their own specific and well-defined functions and interactions with the outside environment, so that a specific security perimeter can be determined in order to properly implement security enforcement policies and mechanisms. Instead, they often live in a strict integration with other devices, such as smartphones or personal assistants, extending their usability and adding some new functionalities. For example, the Apple Watch is almost useless in absence of an association with an iPhone or iPad; the same holds for the Galaxy Gear device, that for working as expected needs the presence of an Android device such as a phone or tablet. This introduces potential threats associated to vulnerabilities in the communication link (e.g., an insecure Bluetooth connection) and to the eventual compromise of the connection point to the external networks (i.e., the associated smartphone/tablet). Second, there are very few state-of-the-art consolidated application-level security mechanisms available for wearable devices, opening the door to any kind of sensitive information disclosure or security breach associated to activities driven by such devices. Last, but not least, usability is a major concern in securing small-sized devices such as smartwatches; in fact, due to the very limited dimension of their touch screens, there are no chances for complex passwords or challenges, and, despite their known vulnerabilities to several kind of attacks, very simple and immediate mechanisms such as pattern locks and PIN codes seems to be the only viable alternatives for performing authentication activities on such devices.

In this work we present a new smartwatch-oriented user authentication mechanism that is both very usable and resilient to the most common attacks a smartwatch might be subjected to. To prove our claims, we have performed a usability study with several real, independent users involved. The paper is structured as follows: in section 2 we describe the state of the art; in section 3 we present our assumptions and our threat model; in section 4 we introduce the CirclePIN concept and we describe how it is to be used; in section 5 we perform a security analysis of our mechanism to show it's resilient to the most common attacks; in section 6 we describe our usability study and we discuss our findings; finally in section 7 we provide some concluding remarks.

2 RELATED WORK

Recently, being a very personal interface to the Internet of Things, smartwatches have attracted the attention of researchers with regards to the methodologies and techniques that can be adopted to perform usable and reliable authentication. Until now, there are two predominant types of authentication mechanism on smartwatches: the regular PIN and Pattern Lock. Unfortunately, these authentication methods are known by their vulnerability to several types of attack such as brute force, shoulder surfing and side channel attacks. In this section we briefly review some of the most relevant proposals for smartwatch user authentication.

Yang et al. [8] proposed MotionAuth, a motion-based authentication for wrist worn devices. MotionAuth uses the movement data collected during gesture performance and two different verification methods, namely a histogram method and a dynamic time warping method, to verify the identity of the user wearing the smartwatch. Authors studied four different types of gestures, namely, raising hand, lowering hand, rotation, and circle. Their method, however, requires users to perform awkward movements, e.g., drawing a circle in the air, which is not convenient in public places. Similarly, Lewis et al. [9] introduced a motion-based real-time authentication mechanism for smartwatches. Their system applies behavioral biometrics to verify the user's authenticity by collecting the data measured by the accelerometer and gyroscope when the user perform a given gesture and uses the dynamic time warping algorithm for template generation and matching. Authors achieved an accuracy of 84.6%. However, the results were based upon a very limited dataset (i.e., 5 users). Johnston and Weiss [10] studied the feasibility of using smartwatches for gait-based biometrics and found that gait is sufficient to identify an individual with modest accuracy but it is not sufficient to be used alone. Authors mention that the main limitation of their work is that the data for each user was collected on the same day which is not a realistic model for a real-world application. In addition, their preliminary tests show that when the training and test data are collected from different days the results degrade significantly. Nguyen and Memon [11] developed TapMeIn, a tap-based authentication method for smartwatches. To login, the user is required to perform a sequence of rhythmic taps anywhere on the touchscreen. During the user's taps, some physiological and behavioral characteristics are collected such as tap pressure, size of touch and timestamp to verify whether the tapped melody belongs to this user. Authors achieved an accuracy of 98.7%. However, they mentioned that the usability results were very surprising and deserve further investigation because their study have been conducted in a lab with a limited dataset which may favor the classification process and may result in high accuracy. In addition, they didn't analyze the effect of other conditions such as the user's age on the performance of their system. The aforementioned authentication methods are related to a specific behavior of a human while performing some tasks, such as hand movement, gait, and typing rhythm. However, in [12], authors described the limitations related to each behavioral biometric approach. For example, Multiple users may have the same hand waving patterns. Wearing an outfit, such as a trench coat or a footwear, may change a person's walking style and person's typing behavior changes considerably throughout a day with different states of mind such as excited, tired, etc. These limitations related to human behavior nature among others might be the main barriers to solely rely on a behavioral system.

On the other hand, a significant amount of research has been conducted to enhance the security of the regular PIN or Pattern lock authentication methods on smartphone (e.g. [13–19]), ATM (e.g. [20–23]) and recently on smartwatches. However, beside the security issues, Pattern and PIN entry on a smartwatch suffer from usability issues related to small screen size. In order to address this issue, Oakley et al. [24] proposed a novel authentication input called PIC (the Personal Identification Chord). Unlike the PIN, PIC enables the user to use only four large buttons to enter ten different inputs via taps to one or two larger buttons. However, while their recall study shows that both PICs and PINs achieve high recall rates and input accuracy, their usability study shows that PIC is modestly slower and more error prone than PIN. In addition, PIC is not resilient to side channel and shoulder surfing attacks. In [25], authors introduced a two-factor authentication method, called Draw-a-PIN. To log in, the user is required to draw his PIN digits sequentially on the touchscreen instead of typing it. Beside the correctness of the PIN, Draw-a-PIN uses the drawing behavior of the user as an additional authentication factor. While Draw-a-PIN provides some advantages with respect to shoulder-surfing resilience, the usability study of its implementation on a smartwatch [26] showed that it is not usable to unlock the smartwatch due to its high error rate and long authentication time (i.e., Overall Average Error rate 20.65% , Overall Average authentication time 7356 ms).

The work we present in this paper is inspired from the Color Wheel PIN protocol (CWPIN) [20]. This protocol uses the smartphone to enhance the security of the PIN input on the ATM. In addition to the PIN, a table of ten colors is shared between the user (i.e., smartphone) and the server. Before the authentication, a random color index arrangement is exchanged between the user (i.e., smartphone) and the server through QR code or NFC. To login, the user is required to use the colors that correspond to the first and the third PIN digits from the color table on his smartphone as an indicator to input the second and fourth PIN digits respectively on the ATM. The user input is performed by sliding the seekbar to rotate the Color Wheel displayed on the ATM machine. The whole process achieves resiliency to many types of attack such as skimming, recording, spyware injection and shoulder surfing. At the same time, the authentication process remains intuitive and short, thus highly usable. However, using the smartphone as an additional authentication factor and the seekbar to rotate the Color Wheel make it impractical to unlock the smartwatch. In this paper, we suggest a new way to input the PIN on the smartwatch using the crown. To the best of our knowledge, this is the first scientific paper that proposes to use the crown to input the PIN on a smartwatch.

3 ASSUMPTIONS AND THREAT MODEL

In many recent studies, researchers have considered smartwatches as a side channel to infer secrets entered on external devices such as smartphones [27, 28] and ATMs [29, 30] thus not taking into account the smartwatch as a venue for authentication; on the contrary, in [31], authors show the feasibility of inferring the user's password entered into the smartwatch itself through touch screen.

In this work, we are focusing on smartwatches as a future endpoint and personal interface to the Internet of Things, as such, we are interested to the second category of side-channel attacks. As an example of a smartwatch compromised with a side-channel enabling malware, we assume that the user installs on his smartwatch "Snoopy" [31], a fitness or gaming app which is instead a Trojan that eavesdrops motion data when users type or swipe their passwords on smartwatches. Snoopy periodically uploads the extracted motion data to the cloud, where it leverages deep neural networks trained with crowd-sourced data to infer the user's passwords.

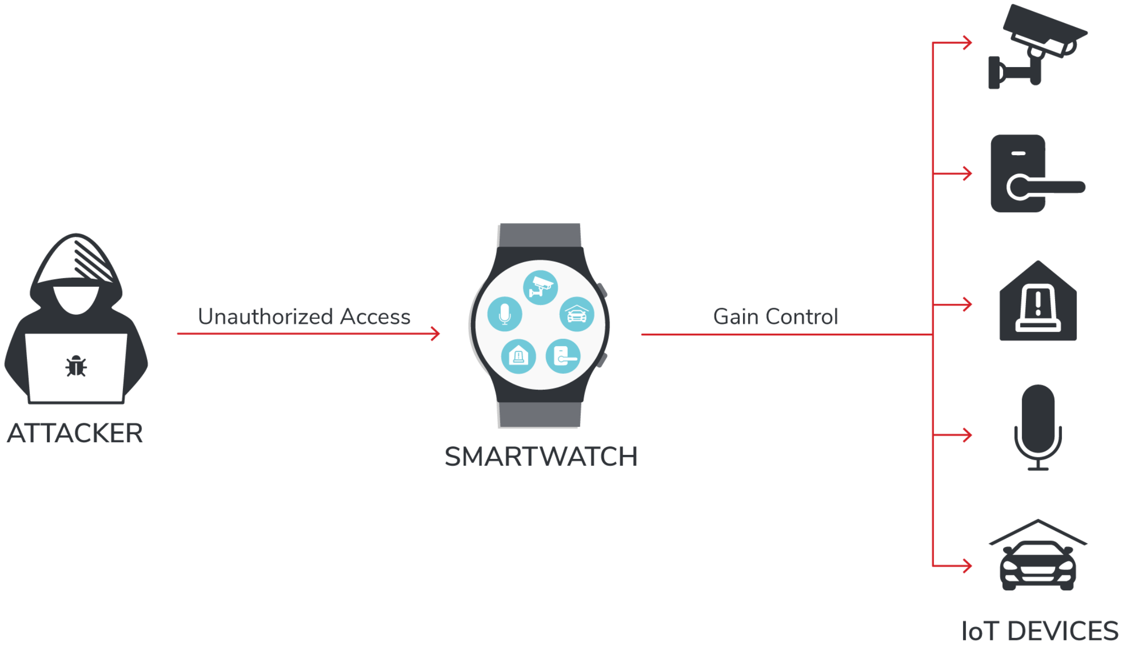

Fig. 1. Unauthorized access control to IoT devices through smartwatch

In relation to shoulder-surfing attacks, we assume that the user types his PIN code using CirclePIN in a public space where the risk of being observed by someone is high. The observation can be made once or multiple times. In addition, we assume that the shoulder surfer is reasonably close to the user and he is able to observe the entire authentication session. In the video-recording attack scenario, we assume that an attacker records the entire authentication session by using his smartphone camera to watch it later and to try identifying the user's PIN.

4 THE PROPOSED AUTHENTICATION METHOD

In order to highlight the concept and the benefits of CirclePIN, we assume that the user has installed on his smartwatch an application that controls some IoT devices. As shown in Figure 1, unauthorized access to the smartwatch enables the attacker to access all the IoT systems controlled and monitored by this watch. Unfortunately, the current dominant authentication methods on smartwatch are not sufficient [32] and suffer both from usability issues related to small screen size and security issues [31]. To address these issues, we introduce CirclePIN, a new secure and usable way to input the PIN on smartwatches. We achieve this goal by using smartwatch-specific features (i.e., the digital crown or the rotating bezel) and a random mapping between the PIN digits and a selection of colors.

The proposed method consists of K steps, where K=n and n denotes the length of the password. CirclePIN uses $(\frac{k}{2})$ steps to memorize the colors that correspond to $(\frac{n}{2})$ PIN digits and the $(\frac{K}{2})$ other steps to use these colors to input the $(\frac{n}{2})$ remaining PIN digits. Thus, n should be an even number. In this paper, we used four-PIN digits as password. Thus four steps are required, the first and third steps to memorize the colors that correspond to the first and third PIN digits and the two other steps to use these colors to input the second and fourth PIN digits.

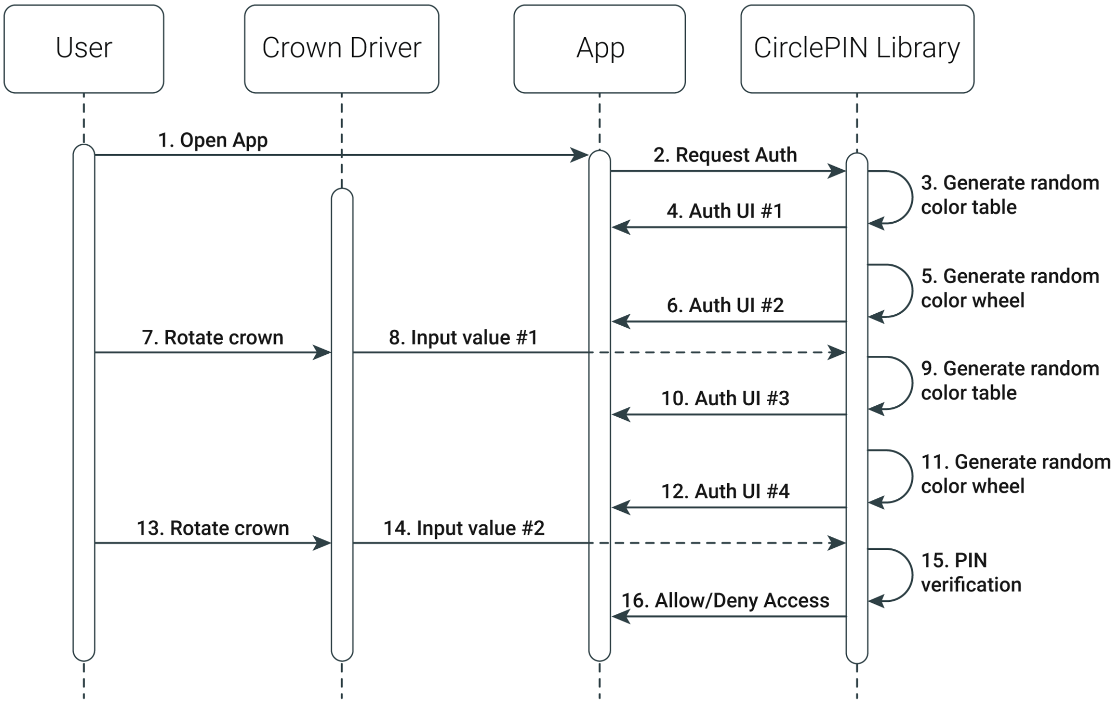

Fig. 2. CirclePIN authentication flow.

Figure 2 depicts CirclePIN authentication flow for four-PIN digits where CirclePIN is implemented as a library and embedded on a smartwatch application. Once an user taps on the application icon (step 1), an authentication request is sent to the CirclePIN library (step 2). At this point the CirclePIN library generates a random sequence of the table colors and renders the interface ♯1 that will be displayed to the user (steps 3 and 4). The user interface ♯1 consists of a table of ten colors arranged randomly. Then, the CirclePIN library generates a random sequence of the wheel's colors (step 5). After a short time[1], in which the user needs to identify which color is associated with his first PIN digit from the user interface ♯1, the second user interface will be displayed which consists of a colored wheel with ten equally sized sectors surrounded by numbers from 0 to 9 (step 6). In order to input the first pair of the four-PIN digits, the user is required to match the color of his first PIN digit with his second PIN digit by rotating the wheel through the smartwatch crown (step 7). If by chance the color and the digit are already in the right position, the user has only to click on the crown instead of turning it. The crown driver gets the rotation degree and sends this information to the CirclePIN library (step 8).

In the same way, the CirclePIN library generates a random sequence of the table colors and renders the user interface ♯3 (step 9). The user interface ♯3 which consists of a table of ten colors arranged randomly will be displayed (step 10). Once the colored table is displayed, the user has to recognize the color associated with his third PIN digit. The CirclePIN library generates a random sequence of the wheel's colors and renders the interface ♯4 (step 11). The user interface ♯4 consists of a colored wheel with ten equally sized sectors surrounded with numbers from 0 to 9 (step 12). In order to

[1] one second is usually enough, but the actual time may be configured by the user

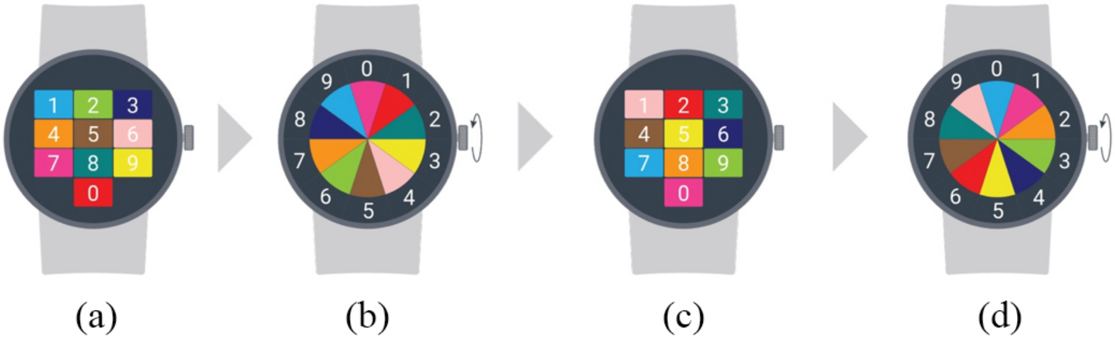

Fig. 3. CirclePIN authentication method.

input the second pair of the four-PIN digits, the user is required to match the color previously identified from table (i.e., user interface ♯3) with his fourth PIN digit through the smartwatch crown (step 13). The crown driver gets the rotation degree and sends this information to the CirclePIN library (step 14). After the second user input, the CirclePIN library is able to check if the input provided matches the user's PIN (step 15). At the end of the process, a notification - either success or failure - is displayed to the user through the smartwatch interface (step 16). Thus, instead of the traditional password-based authentication methods, we input n digits through ($\frac{n}{2}$) gestures.

An example of an authentication process is explained in Figure 3 in the hypothesis that the user's PIN code is '0182'. First, the user looks to the color of the first PIN digit '0' in the table, so, he recognizes the red color as an indicator. Then, he uses the crown to rotate the red color to '1' as shown in Figure 3 (b). When the user releases the crown, another randomized colored table will be displayed as shown in Figure 3 (c). The user memorizes the color of the third PIN digit '8' in the table (i.e., Orange). Then, the wheel rotates to a random degree. The same process is repeated with number '8' and '2', the user recognizes the Orange color as an indicator and uses it to input the number '2' as shown in Figure 3 (d).

CirclePIN authentication method has been developed to improve the user authentication on the smartwatch when it acts either as sensitive data sources or as user identity proxy to another IoT devices. Hence, how the smartwatch generates and sends the access token to the IoT devices is out the scope of this paper and can be the subject of a future work.

5 SECURITY ANALYSIS

5.1 Guessing attack

The chance that a guessing attack (GA) succeeds depends on the size of the password space. The larger the password space, the smaller the chance that a guessing attack is successful. The number of possible CirclePIN password combination is 10^n, where n denotes the length of the password. Since, CirclePIN is designed in such a way that ($\frac{n}{2}$) digits are used as an indicator to input the other ($\frac{n}{2}$) digits, n should be an even number. Thus, the probability of guessing successfully the user's password is:

$$P_{GA}(CirclePIN) = 10^{-n} \; where \; n = 2m \; for \; some \; m \in \mathbb{N}. \quad (1)$$

In this paper, CirclePIN uses the standards four-PIN digits and thus, the success probability of a guessing attack against CirclePIN is $P_{GA}(CirclePIN)$= 10^{-4}. Similar to the regular PIN, CirclePIN prevents any further inputs after three-failed passcode attempts.

5.2 Side channel attack

As described above, smartwatches could be infected by a trojan horse (e.g., Snoopy) [31] that, while masquerading as a fitness or gaming app, could infer the PIN typed on the smartwatch touch screen. Side-channel attacking malware, including Snoopy, leverage motion sensors to measure the smartwatch minor displacement caused when the user interact with the touch screen (i.e., tap or swipe his finger). However, this attack is not effective if the PIN pad dynamically change (i.e., random keypad) [31] or if the user don't use the touch screen as means of input. CirclePIN provides adequate protection against this kind of attack by leveraging the smartwatch digital crown to input the PIN instead of the touch screen. As we will describe in details in section 6.1, in some cases the user is required to click on the center of the wheel or on the Ok button, an action that could be captured by a side-channel attacking malware; yet, this interaction with the touch screen doesn't reveal any useful information related to the PIN digits.

Since CirclePIN is the first work that uses the crown for PIN input on smartwatch, to the best of our knowledge there is not any work in the literature that aim to detect the crown movement through sensors as side channel. However, even if an attacker succeed to reveal the crown rotation degree using sensors, he cannot reveal the user's PIN. This is because there is not a direct connection between the user's crown rotation degree and the PIN digits. When the user rotates the crown in step 2 and 4, ten colors in the wheel will be matched with ten numbers, while there is no way to know which combination (i.e., color-number) among them is part of the user's PIN and what is the correspondent number of this color through smartwatch sensors. In addition, the user's crown rotation degree is different for each authentication session due to the randomization of the colors position which prevent replay attack. Hence, CirclePIN is resilient to motion-based side-channel attacks and its success probability against CirclePIN is 0.

Although stealing the PIN through smartwatch when the users authenticate to their smartphones [27, 28] or ATM machines [29, 30] is out the scope of this paper, we wanted to mention that using CirclePIN on these devices provides protection against this category of side-channel attacks as well. As shown in Figure 4, to unlock the smartphone using CirclePIN, the user has to slide the seekbar to rotate the Color Wheel. However, since the user's input is different each time he authenticates, due the randomization of the table colors and the wheel colors positions, the smartwatch couldn't reveal any useful information about the PIN.

5.3 Shoulder surfing attack

Performing authentication using the PIN or Pattern Lock in public environments (i.e., coffee shop, library, bus, class) exposes the user to shoulder surfing attack (ShA). Attackers can observe the user's PIN or Pattern by looking over the user's shoulder.

Unlike the traditional PIN and Pattern lock, CirclePIN is an indirect input of the user's PIN that leverages the crown and a random mapping between the PIN digits and a selection of colors. In the first step, the attacker cannot know which color in the table belongs to the first PIN digit, hence, he is not able to identify which color the user is using to input the second PIN digit. In addition, when the user turns the crown in the second step, all the colors of the wheel will turn with the same degree and will be matched by a specific number. Hence, ten colors will be matched with ten numbers and only the user knows which one is part of the PIN. The same holds for steps three and four. One potential attack scenario is to try to guess the first and the third PIN digits and memorize their corresponding colors in step 1

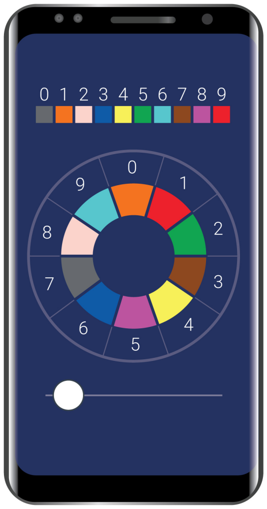

Fig. 4. CirclePIN authentication method on smartphone.

and 3. Then, memorize the numbers matched with these colors in step 2 and 4 after the user's input. This attack is challenging due to the short confirmation time of the user's input in step 2 and 4 which usually takes less than 750 ms. Nevertheless, even if an attacker has a perfect view and was able to identify the numbers correctly in step 2 and 4, the success probability of shoulder surfing attack against CirclePIN is:

$$P_{ShA}(CirclePIN) = 10^{-(\frac{n}{2})} \; where \; n = 2m \; for \; some \; m \in \mathbb{N}. \tag{2}$$

Since, in this paper we used the standard four-PIN digits to make it simple, the attacker has to guess only the first and the third PIN digits in step 1 and 3 with a probability of 10^{-1} and memorizes their corresponding colors. We assume that he was able to follow the user's input and identify the numbers matched with these colors in step 2 and 4. Hence, in the worst cases the probablity of shoulder surfing attack against CirclePIN is:

$$P_{ShA}(CirclePIN) = 10^{-1} \times 1 \times 10^{-1} \times 1 = 10^{-2}. \tag{3}$$

Even if we assume that an attacker has an exceptional memory and succeeds in memorizing four sets of ten combinations of numbers with their corresponding colors (i.e., forty combinations), he cannot reliably guess the user's PIN. In this scenario, while the user focus his attention on one particular combination at each authentication step, the attacker have to pay attention and memorize all the ten combinations at each step. This, according to studies, given the short time the whole authentication process takes, is beyond human memorization capabilities; furthermore, even if the attacked could memorize all the combinations perfectly, the success probability cannot be greater than $10^{-(\frac{n}{2})}$.

5.4 Video-recording attack

It is known that using an external recording device such as a smartphone to record the user's credentials is a more serious threat than shoulder surfing attacks. However, in real world scenario, it is not always easy to perform such an attack without the user's awareness.

CirclePIN provides an improved resilience against video-recording attack (VRA) comparing to the traditional PIN. Using CirclePIN, an attacker cannot get enough useful information about the user's PIN by simply watching a single video recording. In order to make the attack more efficient, the attacker has to write down the four sets of ten combinations of numbers with their corresponding colors. Then, match the first set with the second and the third with the fourth in order to obtain two lists of ten combinations of only numbers. The first two PIN digits are part of the first list, while the two other PIN digits are part of the second list. Hence, there are 100 possible combinations to find the correct PIN digits. Thus, the probability that an attacker successfully reveals the user's PIN using CirclePIN is 0.01.

In order to simulate video-recording attack, we recorded a successful authentication session to CirclePIN for each participant (i.e., 19). The video was recorded using the smartphone camera when the user was seated. We recorded the video in such a way the entire CirclePIN authentication session can be clearly observed with no obstruction of user's hands or shoulders. This, from the attacker point of view is considered the best case scenario. The videos were numbered and shared with 15 participants who did not participate in the test in order to simulate attackers. The attackers were allowed to watch, stop and replay the videos as long as they wanted. In addition, a document that contain the two lists of ten combinations of numbers of each video has been provided to them because we wanted to evaluate the security of CirclePIN in worst case scenario. The test has been done remotely because, contrarily to the usability test, there is no need for a controlled environment. Attackers were only required to guess three possible combinations of the PIN code from the list provided to them after watching each video one or multiple time and send it to us to compare them with the user's PIN. We compared their guessed PINs to the user's PIN corresponding to each video. The results show that none of the attackers could do any better than the statistical chance of 1 in 100 guarantees. Hence, the success probability of recording attack against CirclePIN is:

$$P_{VRA}(CirclePIN) = 10^{-(\frac{n}{2})} \; where \; n = 4. \tag{4}$$

It is important to notice that, even if 1 in 100 sounds a significant reduction of the secret space provided by a 4 digit PIN, in the case of a recording attack the probability of identifying a PIN entered in the traditional way is 1 (certainty).

5.5 Comparison of the security strength

In this section, we consider that CirclePIN use the four-PIN digits. Theoretically, Pattern lock is more secure than CirclePIN, PIN and TapMeIN methods against guessing attack because the number of the possible valid pattern is 389,112 $\approx 2^{19}$ which is about 39 times more than the 4-digit PIN (i.e., 10000). However, usually users chooses a pattern with a length limited to 4 or 5 dots; this decreases significantly the password space, in fact it becomes just 1624 for 4 dots patterns and 7152 for 5 dots patterns. Thus, the Pattern Lock provides better security than CirclePIN and the PIN only if the user choose a pattern with at least 6 dots as this length guarantees 26016 valid patterns. In contrast, either CirclePIN, PIN and a Pattern with solely 4 dots provide better security than TaPMeIN against guessing attack.

As shown in Table 1, CirclePIN provides a much higher resilience than both 4-digit PIN and Pattern lock against side-channel, shoulder surfing and video-recording attacks and a better security than TapMeIn and Draw-a-PIN against both shoulder surfing and video-recording attacks. It is important to mention that the success probability of side channel attack against PIN and Pattern lock methods in the table are extracted from [31], except the success probability of CirclePIN. While the security evaluation of TapMeIn [11] and Draw-a-PIN [25] against the other attacks are extracted from the original papers. Since, CirclePIN uses a random graphic interface and indirect input through the smartwatch digital crown instead of the touch screen, revealing the user's PIN through motion-based side channel attack is not possible as described in section 5.2.

In addition, CirclePIN is more resilient to smudge attack [33] than both Pattern lock and PIN. It is important to mention that the PIN is partially vulnerable to smudge attack (i.e., the attack can reduce the password space) while Pattern lock is fully vulnerable to this attack.

Table 1. Comparison of the security strength

	CirclePIN	PIN	Pattern Lock	TapMeIn	Draw-a-PIN
Guessing attack probability	0.0001	0.0001	2^{-19}	0.013	NA
Success probability for a side-channel attack [31]	0	0.86	0.95	NA	NA
Shoulder-surfing attack probability	0.01	1	1	0.035	0.067
Video-recording attack probability	0.01	1	1	0.041	0.0989

6 EXPERIMENTAL RESULTS

In this section, we evaluate the usability of CirclePIN and we compare it to the most commonly used authentication schemes, namely lock pattern and 4-digit PIN.

6.1 CirclePIN design

As mentioned above, CirclePIN can be implemented as library. However, in order to evaluate the usability of CirclePIN, we implemented it as standalone application on a LG Style Smartwatch running Android Wear 2.0 OS. For this reason, the design of CirclePIN was customized according to LG smartwatch features.

For example, LG Style watch is equipped with a digital crown rather than rotating bezel, therefore the wheel rotation is performed using the movement of the crown. In contrast to smartwatches that have more than one hardware button and allow the developer to assign a custom action to one or more of these buttons, LG Style Watch has only one hardware button which is used as power button and cannot be customized as it is reserved to the OS. Therefore, we added a circle button in the center of the wheel to replace the click on the crown in the case the wheel is correctly positioned (see Figure 5 (c)). Furthermore, we noticed during the initial testing of CirclePIN that the user can miss the table of color in step one or three in the case he is not focused on authentication or gets distracted by another activity.

Thus, we developed two versions of CirclePIN: the first one removes the table and start displaying the wheel without any user's interaction after a short time (i.e., 1 second), while the second requires that the user swipe his finger anywhere on the screen or click on an OK button on the screen (see Figure 5 (b)).

As shown in Figure 6, the same smartwatch was used for the usability test of the PIN and Pattern lock and all the developed test applications show the following menu when the user opens it (see Figure 5 (a)):

Practice: This option allows the user to perform the authentication as many times as he wants to get familiar with the concept and procedural details of the selected authentication method. While the app is in this mode, the time and user's authentication errors are not recorded in the log file.

Start Test: This option depending on the authentication method he is currently testing, makes the user authenticate ten times. In this phase, the time and the user's errors are recorded in the log file.

Settings: This option allows the user to enter his PIN or Pattern. Special settings related to CirclePIN allow the user to select the version of CirclePIN (timed or with explicit acknowledge). In addition, if timed version of CirclePIN is selected, this option allows the user to personalize how much time the table of color will remain visible before presenting the wheel to the user (by default this time is 1 second).

Log files: This option allows us to access the log files of all the participants.

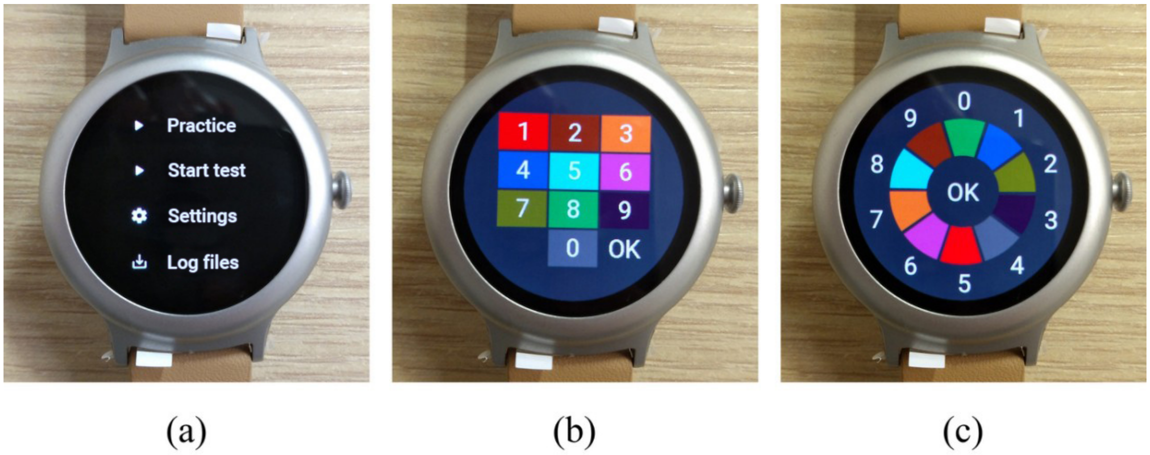

Fig. 5. CirclePIN Version 2 authentication method.

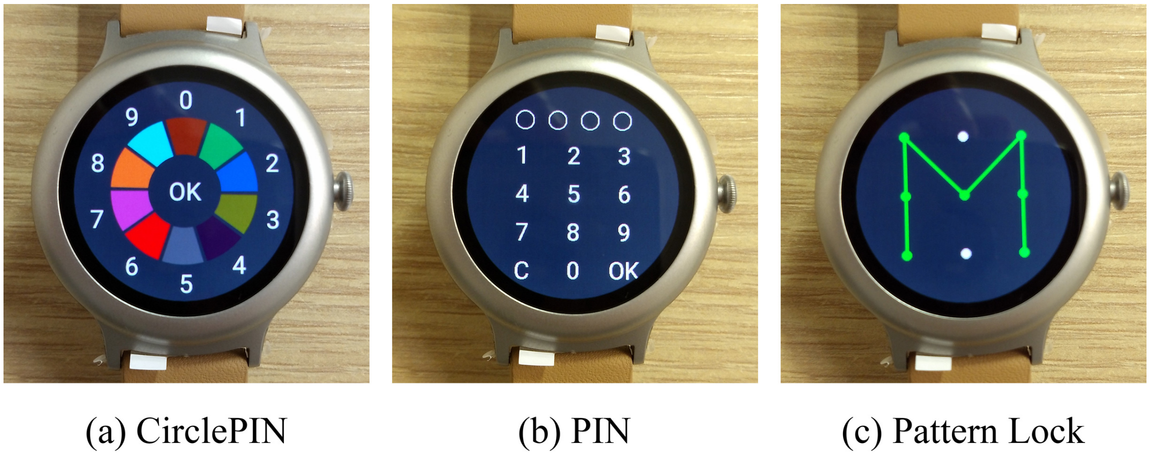

Fig. 6. Smartwatches authentication methods.

6.2 Participants

We recruited 34 volunteers from both the University of Padova and Genoa. The age of the participants to the study ranges from 21 to 59, among the participants 8 are females, while 26 are males. All participants were smartphone users but only three had any experience with smartwatches before this study. In our experiments, 19 participants participated as users and 15 as attackers.

6.3 Procedure

All tests started with the classic PIN and Pattern lock methods. Each participant was asked to enter a 4-digit PIN of his choice and a pattern with length larger than or equal to 4 in the settings of the application. Then, they were encouraged to do some training to get used to the keypad and the pattern on smartwatch. Due to the popularity of these methods, after just five training sessions, most of the participants declared to be ready for the actual tests. In the test phase, each participant was asked to unlock the smartwatch using each method ten times. Thus, the measured authentication time and error rates are based on 190 authentication sessions performed by 19 participants. The PIN authentication time was measured from the first key press to releasing the OK button, while the Pattern lock authentication time was measured from the time the participants first touch the screen until they lift their fingers.

The same procedure was followed to evaluate the usability of CirclePIN. More in details, each participant was asked to choose a 4-digit PIN. Then, the concept was directly explained using their PIN. Afterwards, they were encouraged to practice unlocking with the two versions of CirclePIN until they felt familiar with the concept. Usually, after 10 attempts on their own, they were already inserting the PIN correctly. In the test phase, they were required to repeat the authentication process ten times for each version of CirclePIN. Thus, the results are based on 380 authentication sessions performed by 19 participants.

In CirclePIN with explicit acknowledge (the OK button version) the authentication time was measured from the first key press to releasing the crown or the central button in the last wheel. While, in the other version, the authentication time was measured from touching the crown for the first time to releasing the crown or the central button in the last wheel.

At the end, we recorded each participant during one successful authentication session with CirclePIN using the smartphone camera, after asking permission and explaining to them that only the smartwatch would be recorded and what is the purpose of this video. Once participants have completed the test, a satisfaction survey was provided to each participant in order to collect the participant's evaluation for the CirclePIN method. Then, they were invited to note their comments as well as any problem they encountered.

6.4 Results and comparison with existing unlocking methods

The logged data stored on the smartwatch during the usability study were used to calculate the average authentication time and error rate of each user. Table 2 allows comparing the average authentication time and error rate of the proposed authentication method with the two most common methods currently available on smartwatches, namely PIN and Pattern Lock. Relevant statistical parameters, including confidence intervals, of the measured timings are summarized in Table 3, while, Figure 7 shows the box plots of the average timings.

Table 2. Comparison of CirclePIN with PIN and Pattern Lock authentication methods.

Authentication method	**Authentication time (s)**	**Error Rate (%)**
PIN	2.37	4.73
Pattern Lock	1.14	3.68
CirclePIN V1	4	1.05
CirclePIN V2	4.38	1.05

Test results show that Pattern Lock has the fastest authentication time. The reason is that the most participants choose a simple pattern such as the shape L. While, the authentication time of CirclePIN is slower than both the PIN and Pattern lock methods. We argue that the main reason is that CirclePIN adds an additional cognitive task. Unlike PIN and Pattern, using CirclePIN, the participants had to recognize the colors corresponding to the first and the third PIN digit and use it for the input. Thus, they need to match PIN digits with colors instead of directly input the secret. Yet, that's what makes CirclePIN more secure.

Table 3. Values and Statistical Parameters

	C.PIN-V1	C.PIN-V2	PIN	Pattern Lock
Average	4,00	4,38	2,37	1,14
Min	3,12	3,18	1,64	0,55
Max	6,35	6,76	4,63	2,17
Var	0,70	0,68	0,61	0,22
StDev	0,84	0,82	0,78	0,47
Conf Int alpha 0.05	0,39	0,38	0,36	0,22
Conf Int alpha 0.01	0,51	0,50	0,48	0,29

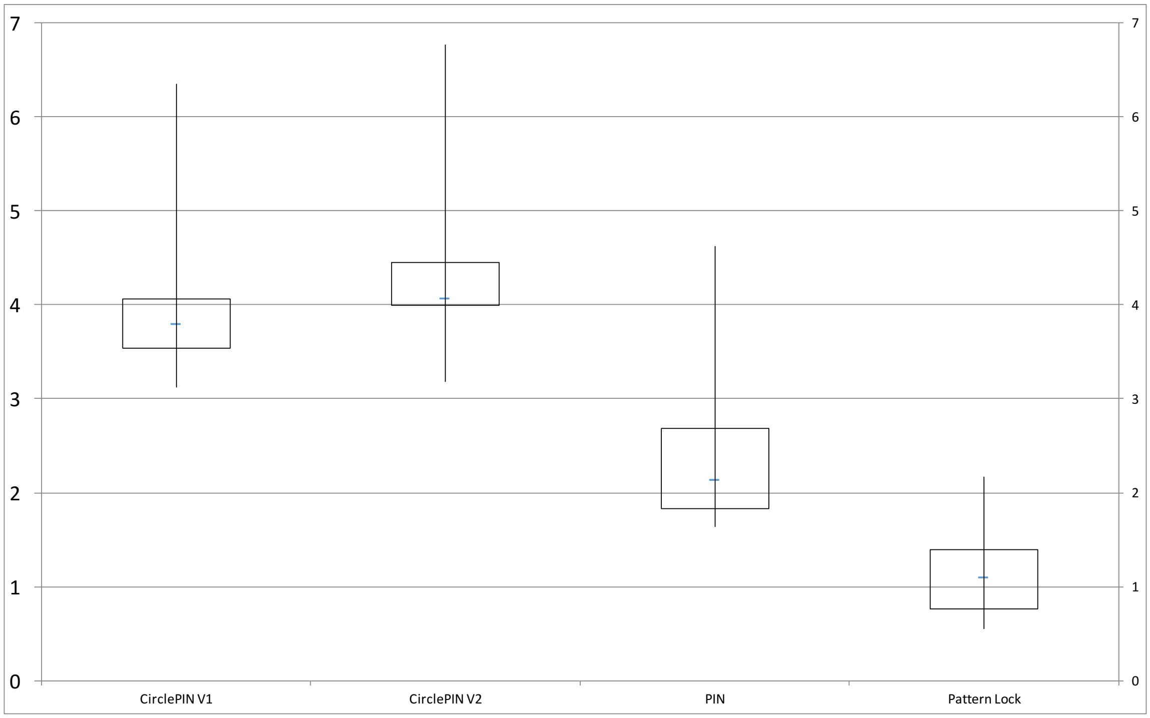

Fig. 7. Box Plot for the average authentication time of CirclePIN, PIN and Pattern lock authentication methods.

The two versions of CirclePIN had approximately the same authentication time, the small difference of time in version 2 is due to the additional user interaction with the screen in step one and three (i.e., user clicks on OK button or swipe his finger on the screen). The results in our study are almost similar in terms of input time with the previous studies that evaluate the PIN and pattern lock on smartwatches [11, 26]. In [26], the average authentication time of PIN and pattern is 2.17 s, 1.17 s, respectively. While in [11], the average authentication time of PIN and pattern is 2.19 s, 1.14 s, respectively.

In terms of error rate, CirclePIN has the lowest error rate among the evaluated methods and this is due to the use of the crown as mean of input rather than the touch screen. We believe that the very few errors that occurred were due to an ambiguity caused by choosing two shades of the same color (i.e., dark green and light green) in the CirclePIN test application. This was confirmed by a participant comment. The two versions of CirclePIN had the same error rate which means that swiping or tapping freely on the smartwatch screen doesn't lead to errors, in contrast to the PIN and Pattern lock which require the user to touch a small dots or small button size. Unlike [11, 26], our results show that pattern lock is less error prone than the PIN. In [26] [11], the error rate of the PIN method is 5.7%, 5.2%, respectively. While, the error rate of pattern lock is 7.3%, 6.8%, respectively. Hence, further investigation is needed with a larger data set.

Observing the input of CirclePIN, the lowest authentication times were achieved in the cases where at least on one input the color wheel was already positioned correctly (i.e., 1.62 seconds). The longest authentication times were experienced when the oldest participants (i.e., two participants aged 59 years) needed to watch the colors table for a longer time (i.e., 2 seconds) than the default setting of one second; that additional time affected significantly the authentication time (i.e., 6.76 Second). In Addition, we noticed that there is a correlation between input time and how much the user had to move the wheel to get to the correct position. Hence, CirclePIN in some cases takes less time than the regular PIN and in average, a delay of two seconds in return of an improved security and reduced error rate still reasonable. Similar to the above-mentioned studies [11, 26], the main limitation of this work is the limited number of participants. Nevertheless, as a proof-of-concept, we have shown that CirclePIN provides a novel usable way of PIN authentication on smartwatch through the digital crown. Our results are consistent with the previous results that show that the rotatable bezel as well as the digital crown represent usable alternatives to touch on smartwatches [34].

6.5 Survey of Participants Perception of Our Scheme

In order to further evaluate the usability of CirclePIN, we asked the participants to rate each statement from 1 (Strongly Disagree) to 10 (Strongly Agree) . Thus, a higher score refers to a positive opinion. The questions were provided to participants as follows:

Accurate input: Was it easy to accurately turn the crown and thus the color to match the PIN?

Understandability: Was the concept easy to understand?

Memorability: If you did not use this mechanism for a few months, do you expect would you still remember how it works?

Pleasant: Was this mechanism pleasant to use?

Suitability: Is this mechanism suitable for the smartwatch?

Preference: On a smartwatch, do you prefer using the CirclePIN mechanism compared to PIN/ Pattern lock?

Input mechanism: Was the input mechanism (crown) better than traditional input mechanisms (e.g., Touch screen)?

Figure 8 shows the average value of the usability ratings of participants. We notice that all statements were scored positively by participants. The participants generally found the CirclePIN mechanism easy to understand and to use. All of them stated that they would be able to remember the concept easily even months later. In addition, they found it well suited to the smartwatch form factor, as it does not require text input using small virtual keypad. All these features made the participants express their preference for the CirclePIN mechanism over both PIN and Pattern Lock methods. In contrast to the users perception in [26], our results suggest that users tend to favor a locking method that has both a reasonable authentication time and a low error rate in comparison with the PIN and pattern lock. In order to know which version of CirclePIN the users prefer, we added the following questions:

CirclePIN Version: Do you prefer the first version of CirclePIN or the second which requires that the user click on OK button or swipe his finger to pass to the input step?

Most participants stated that they would prefer the first version of CirclePIN in which the wheel displays automatically after a configurable time.

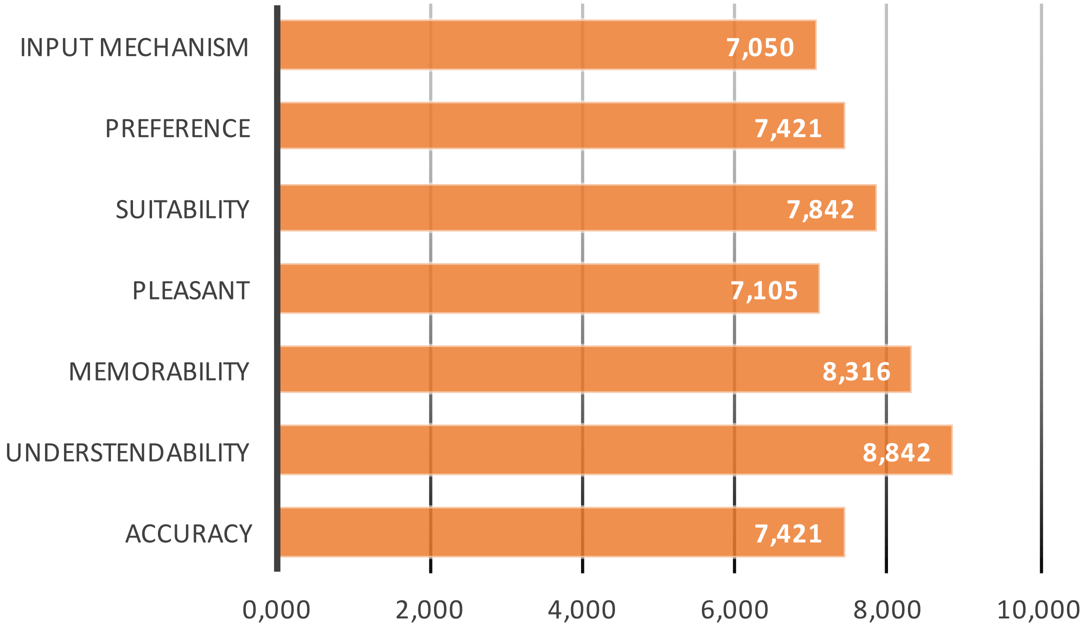

Fig. 8. Usability Average User's Score of CirclePIN method

6.6 User comments

We invited participants to provide us comments regarding their experience with CirclePIN concept and test application by means of a free text section in the survey. We also asked to provide suggestions on how to improve the CirclePIN test. Selected comment samples are shown in Table 4.

Table 4. Some Examples of User Comments

Sex	Comments
M	Interesting mechanism, I would probably use it when I need extra security.
M	A gear like the one in the Samsung Galaxy S3 would be more comfortable and precise, I think it is more secure.
M	the crown is too sensitive. The principle is enjoyable!
M	I have never seen an unlock system like this but I was fascinated. I think it is more secure with respect to a classic PIN unlock sys but convenient as well.
M	Being a left-handed person I prefer to wear a watch on the right arm, so the crown on the right side of the watch is not ideal.
F	The concept is easy and practical, but the two green colors makes me confused sometimes which one of them I have to use for th

The Participants comments were very interesting and helped us improve the usability of CirclePIN. For example, one participant expressed an issue with the two green colors, she was able to distinguish them but sometimes had some problems remembering which green color she saw on the color table. This issue can be addressed by choosing more distinguishable colors in the implementation. Some participants had issues with the rotation of the wheel, they initially found that it was too sensitive then they got used to it during the training phase. The crown sensitivity can be an adjustable parameter in further implementations. One participant expressed annoyance toward the fact that the crown is on the right side of the smartwatch and it felt awkward to him operate the crown with his left hand (being left-handed he wears watches on his right arm). The best option for left-handed people will be to use the rotating bezel instead of the crown or to have smartwatches specifically built for left handed people.

7 CONCLUSION

The merging of Internet of Things devices into critical infrastructures is changing the security scenario. While critical infrastructures used to be isolated islands to be protected mainly from physical threats, the introduction of the IoT paradigm stresses the interconnection of the systems and the reliance of each part on data and services provided by other parties.As an example, crowd-sourcing data from personal devices allows gathering real-time car traffic data to provide up-to-date routing with traffic-jam avoidance.

However, the very same mechanism exposes a critical infrastructure such as the navigation of self-driving cars to new types of attacks such as a sibyl-like attack injecting fake data and generating false traffic-jam alarms. Hence, the need for strong and usable authentication in personal devices is of paramount importance. At the same time, a new category of personal devices, namely the smartwatch, is quickly gaining traction and visibility in the panorama of personal devices for its unobtrusiveness and the rich set of sensors it can integrate in a very small footprint, thus turning from a high-tech gadget with very specific uses (e.g., health-monitoring) into an Internet connected proxy for user's identity and position.

In this paper, we have presented CirclePIN, an innovative, smartwatch dedicated authentication methodology that is endowed both with resilience to the most common attacks and a high degree of usability. To prove our claims, we performed security analysis and a usability study; our findings show that, indeed, CirclePIN provides enhanced security against side channel, shoulder surfing and single recording attacks, while it still provides a level of usability that is comparable with less secure and more widely known authentication methodologies such as PIN or lock pattern. For future work, we plan to test CirclePIN as well as the most common authentication methods on smartwatches with a large number of participants.

REFERENCES

[1] N. Gobbo, A. Merlo, and M. Migliardi, "A denial of service attack to gsm networks via attach procedure," in *Security Engineering and Intelligence Informatics*, A. Cuzzocrea, C. Kittl, D. E. Simos, E. Weippl, and L. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 361–376.

[2] M. Migliardi and A. Merlo, "Modeling the energy consumption of distributed ids: A step towards green security," in *2011 Proceedings of the 34th International Convention MIPRO*, May 2011, pp. 1452–1457.

[3] P. C. van Oorschot, A. Somayaji, and G. Wurster, "Hardware-assisted circumvention of self-hashing software tamper resistance," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 82–92, April 2005.

[4] M. Jakobsson, "Secure remote attestation," 2018. [Online]. Available: https://eprint.iacr.org/2018/031.pdf

[5] H. Tschofenig, "Fixing user authentication for the internet of things (IoT)," *Datenschutz und Datensicherheit - DuD*, vol. 40, no. 4, pp. 222–224, apr 2016. [Online]. Available: https://doi.org/10.1007/s11623-016-0582-1

[6] M. Guerar, A. Merlo, and M. Migliardi, "Completely automated public physical test to tell computers and humans apart: A usability study on mobile devices," *Future Generation Computer Systems*, vol. 82, pp. 617 – 630, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17303709

[7] M. Guerar, A. Merlo, M. Migliardi, and F. Palmieri, "Invisible cappcha: A usable mechanism to distinguish between malware and humans on the mobile iot," *Computers & Security*, vol. 78, pp. 255 – 266, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404818307557

[8] J. Yang, Y. Li, and M. Xie, "Motionauth: Motion-based authentication for wrist worn smart devices," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, March 2015, pp. 550–555.

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

[10] A. H. Johnston and G. M. Weiss, “Smartwatch-based biometric gait recognition,” in *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Sept 2015, pp. 1–6.

[11] T. Nguyen and N. Memon, “Tap-based user authentication for smartwatches,” *Computers & Security*, vol. 78, pp. 174 – 186, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404818303778

[12] M. Ehatisham-ul Haq, M. A. Azam, J. Loo, K. Shuang, S. Islam, U. Naeem, and Y. Amin, “Authentication of smartphone users based on activity recognition and mobile sensing,” *Sensors*, vol. 17, no. 9, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/9/2043

[13] M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, F. Palmieri, and A. Castiglione, “Using screen brightness to improve security in mobile social network access,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 621–632, July 2018.

[14] E. von Zezschwitz, A. De Luca, B. Brunkow, and H. Hussmann, “Swipin: Fast and secure pin-entry on smartphones,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI ’15. New York, NY, USA: ACM, 2015, pp. 1403–1406. [Online]. Available: http://doi.acm.org/10.1145/2702123.2702212

[15] M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, and B. Messabih, “A completely automatic public physical test to tell computers and humans apart: A way to enhance authentication schemes in mobile devices,” in *2015 International Conference on High Performance Computing Simulation (HPCS)*, July 2015, pp. 203–210.

[16] M. Guerar, A. Merlo, and M. Migliardi, “Completely automated public physical test to tell computers and humans apart: A usability study on mobile devices,” *Future Generation Computer Systems*, vol. 82, pp. 617 – 630, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17303709

[17] A. Bianchi, I. Oakley, V. Kostakos, and D. S. Kwon, “The phone lock: Audio and haptic shoulder-surfing resistant pin entry methods for mobile devices,” in *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI ’11. New York, NY, USA: ACM, 2011, pp. 197–200. [Online]. Available: http://doi.acm.org/10.1145/1935701.1935740

[18] T. Kwon and S. Na, “Tinylock: Affordable defense against smudge attacks on smartphone pattern lock systems,” *Computers & Security*, vol. 42, pp. 137 – 150, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404813001697

[19] M. Guerar, A. Merlo, and M. Migliardi, “Clickpattern: A pattern lock system resilient to smudge and side-channel attacks,” *JoWUA*, vol. 8, no. 2, pp. 64–78, 2017. [Online]. Available: http://isyou.info/jowua/papers/jowua-v8n2-4.pdf

[20] M. Guerar, M. Benmohammed, and V. Alimi, “Color wheel pin: Usable and resilient ATM authentication,” *J. High Speed Networks*, vol. 22, no. 3, pp. 231–240, 2016. [Online]. Available: https://doi.org/10.3233/JHS-160545

[21] A. D. Luca, E. von Zezschwitz, and H. Hußmann, “Vibrapass: secure authentication based on shared lies,” in *CHI*. ACM, 2009, pp. 913–916.

[22] A. De Luca, K. Hertzschuch, and H. Hussmann, “Colorpin: Securing pin entry through indirect input,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’10. New York, NY, USA: ACM, 2010, pp. 1103–1106. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753490

[23] D. Nyang, A. Mohaisen, and J. Kang, “Keylogging-resistant visual authentication protocols,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 11, pp. 2566–2579, Nov 2014.

[24] I. Oakley, J. H. Huh, J. Cho, G. Cho, R. Islam, and H. Kim, “The personal identification chord: A four buttonauthentication system for smartwatches,” in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ser. ASIACCS ’18. New York, NY, USA: ACM, 2018, pp. 75–87. [Online]. Available: http://doi.acm.org/10.1145/3196494.3196555

[25] T. V. Nguyen, N. Sae-Bae, and N. Memon, “Draw-a-pin: Authentication using finger-drawn pin on touch devices,” *Computers & Security*, vol. 66, pp. 115 – 128, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404817300123

[26] T. Nguyen and N. Memon, “Smartwatches locking methods: A comparative study,” in *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*. Santa Clara, CA: USENIX Association, 2017. [Online]. Available: https://www.usenix.org/conference/soups2017/workshop-program/way2017/nguyen

[27] A. Sarkisyan, R. Debbiny, and A. Nahapetian, “Wristsnoop: Smartphone pins prediction using smartwatch motion sensors,” in *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, Nov 2015, pp. 1–6.

[28] A. Maiti, M. Jadliwala, J. He, and I. Bilogrevic, “(smart)watch your taps: Side-channel keystroke inference attacks using smartwatches,” in *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, ser. ISWC ’15. New York, NY, USA: ACM, 2015, pp. 27–30. [Online]. Available: http://doi.acm.org/10.1145/2802083.2808397

[29] C. Wang, X. Guo, Y. Chen, Y. Wang, and B. Liu, “Personal pin leakage from wearable devices,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 646–660, March 2018.

[30] C. Wang, X. Guo, Y. Wang, Y. Chen, and B. Liu, “Friend or foe?: Your wearable devices reveal your personal pin,” in *AsiaCCS*, 2016.

[31] C. X. Lu, B. Du, H. Wen, S. Wang, A. Markham, I. Martinovic, Y. Shen, and A. Trigoni, “Snoopy: Sniffing your smartwatch passwords via deep sequence learning,” *IMWUT*, vol. 1, pp. 152:1–152:29, 2017.

[32] HP, “Internet of things security study: Smartwatches,” 2015. [Online]. Available: https://www.ftc.gov/system/files/documents/public_comments/2015/10/00050-98093.pdf

[33] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, “Smudge attacks on smartphone touch screens,” in *Proceedings of the 4th USENIX Conference on Offensive Technologies*, ser. WOOT’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–7. [Online]. Available: http://dl.acm.org/citation.cfm?id=1925004.1925009

[34] F. Kerber, T. Kiefer, M. Löchtefeld, and A. Krüger, “Investigating current techniques for opposite-hand smartwatch interaction,” in *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI ’17. New York, NY, USA: ACM, 2017, pp. 24:1–24:12. [Online]. Available: http://doi.acm.org/10.1145/3098279.3098542