Using Screen Brightness to Improve Security in Mobile Social Network Access

Meriem Guerar, Mauro Migliardi, Alessio Merlo, Mohamed Benmohammed, Francesco Palmieri, and Aniello Castiglione, *Member, IEEE*

Abstract—In the today's mobile communications scenario, smartphones offer new capabilities to develop sophisticated applications that seem to make daily life easier and more convenient for users. Such applications, which may involve mobile ticketing, identification, access control operations, etc., are often accessible through social network aggregators, that assume a fundamental role in the federated identity management space. While this makes modern smartphones very powerful devices, it also makes them very attractive targets for spyware injection. This kind of malware is able to bypass classic authentication measures and steal user credentials even when a secure element is used, and can, therefore, perform unauthorized mobile access to social network services without the user's consent. Such an event allows stealing sensitive information or even a full identity theft. In this work, we address this issue by introducing BrightPass, a novel authentication mechanism based on screen brightness. BrightPass allows users to authenticate safely with a PIN-based confirmation in the presence of specific operations on sensitive data. We compare BrightPass with existing schemes, in order to show its usability and security within the social network arena. Furthermore, we empirically assess the security of BrightPass through experimentation. Our tests indicate that BrightPass protects the PIN code against automatic submissions carried out by malware while granting fast authentication phases and reduced error rates.

Index Terms—Smartphone, social networks, mobile-access, malware, authentication, usable security, brightness

1 Introduction

In the last years the attention of social network providers has been more focused on attracting users and building large warehouses of personal information than on securing the access to the infrastructures they have realized, so that state-of-the-art authentication mechanisms provided in most of the common social network facilities are not sufficiently robust. As a consequence, the active users of social networks available on the Internet now amount to several billions people and most of them make use of mobile devices, such as smartphones or tablets, to access the provided services. In addition, many of these users now consider social networks as the preferred way for managing their personal data, and they use their social network access credentials to simplify, through the available social aggregators, the management of their various profiles and accounts on many service portals. According to these trends, social networks are moving from monolithic proprietary applications to fundamental data aggregators as well as hubs in the federated identity management space. Unfortunately, as the strategic importance of these platform grows, the interest of the hackers on them increases as well, so that identity theft and authentication breaches, aiming at several hostile activities, become fundamental problems in the social networking arena, lying at the basis of its most critical security challenges [1].

This is even more important in the mobile communications scenario, where today's smartphones offer new capabilities to develop sophisticated applications that seem to make daily life easier and more convenient for users.

Many attacks are successful in accessing social network accounts since the current password-based authentication paradigms are not efficient and robust enough as well as vulnerable to automated attacks.

A recent study from LinkedIn and Twitter hacks shows that weak passwords and single factor authentication are still the main security weaknesses facing most social networking sites [2]. Accordingly, two-factor authentication seems to be the simplest and most effective protection strategy currently available. Many of the topmost social networking services providers such as Google, Facebook, Yahoo, Twitter, Snapchat, and Dropbox already allow you to optionally require second authentication (e.g., through an additional security code). For example, from the Facebook security settings you can require a security code for accessing your account from unknown browsers, whereas Twitter, if specifically configured, requests to its users, immediately after entering the access password, a six-digit verification code, sent via short text message (SMS) to their cell phone, anytime they try to log in. Similar mechanisms are provided by Google and Dropbox.

- *M. Guerar is with the University of Sciences and Technology of Oran Mohamed Boudiaf, Bir El Djir 31000, Algeria. E-mail: meriem.guerar@univ-usto.dz.*
- *M. Migliardi is with the University of Padua, PD 35122, Italy. E-mail: mauro.migliardi@unipd.it.*
- *A. Merlo is with the University of Genoa, Genova 16126, Italy. E-mail: alessio.merlo@unige.it.*
- *M. Benmohammed is with the University of Constantine, Constantine 25017, Algeria. E-mail: ben_moh123@yahoo.com.*
- *F. Palmieri and A. Castiglione are with the Department of Computer Science, University of Salerno, Fisciano 84084, Italy. E-mail: fpalmieri@unisa.it, castiglione@ieee.org.*

Manuscript received 28 Feb. 2016; revised 24 June 2016; accepted 6 Aug. 2016. Date of publication 19 Aug. 2016; date of current version 6 July 2018. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TDSC.2016.2601603

However, the traditional two-factor authentication mechanisms are not applicable to online social networks because physical token or biometric data cannot be easily (and hence practically) used to log into users' profiles. The simplest alternative is complementing the single factor (password-based) authentication process with additional identification elements, such as one-time PIN codes, generated by the user's own device (e.g., the smartphone) or received via SMS. This can be useful to further confirm the user's identity when signing on from unusual locations or performing specific actions, such as changing or accessing important configuration data within the user's profile. Alternatively, encryption techniques can also be used to complement traditional authentication mechanisms. Unfortunately, the mobile devices used for gaining access are often vulnerable to several kind of malware that can be able to retrieve data such as passwords and PIN codes as they are inserted to perform authentication to the target social network applications. Hence, the presence of such malware in mobile platforms can seriously impact the user's privacy and security, reducing the user's trust in performing mobile access to its preferred social network services.

The importance of security in the authentication process as well as the increase in threat level posed by such malware have attracted many researchers to the field. Hence, numerous authentication methods have been proposed in academic research to prevent malware from performing automatic authentication attempts. Some proposals include an additional secret value or a complex cognitive intelligence test, e.g., CAPTCHA, or a social contact verification [3], to the traditional authentication methods. These, in the most cases, significantly slow the authentication process and, thus, have a low level of acceptance among users.

In this paper we discuss a brightness based authentication mechanism (i.e., *BrightPass*) capable of enhancing the security of identity confirmation PIN codes without asking the user to memorize an additional secret value or to solve a complex cognitive task. This method introduces a new input value that is changed at every usage combining a *something you know* element (i.e., the PIN) with an interface element that cannot be captured by spyware, i.e., a bright or dark circle displayed on the phone screen to tell the user *when* to digit the correct PIN digit and when to digit a fake one.

Unlike the existing authentication schemes, BrightPass does not prevent the spyware from stealing the user's PIN code. On the contrary, it prevents the malware from correctly inserting the PIN code, thereby disallowing the possibility to perform critical operations without the user's agreement. Our experiments show that BrightPass does not hamper usability and provides adequate security for mobile and sensitive applications against different types of spyware that deal with user authentication. Thereby, BrightPass can increase user confidence in accessing social networks. Our scheme has a level of resilience to attacks that makes it usable as a second level of authentication to guard especially sensitive data and operation, but is also shows a level of usability that makes it usable as a candidate to be the only authentication mechanism available. The paper is structured as follows: in Section 2 we present the related work; in Section 3 we introduce the BrightPass scheme; in Section 4 we perform a security analysis of our scheme; in Section 5 we discuss the advantages and disadvantages of our scheme in comparison to other schemes; in Section 6 we describe our experimental usability evaluation and, finally, in Section 7 we provide some concluding remarks.

2 Related Work

Today's smartphones are built on sophisticated mobile operating systems that allow them to run applications with rich functionalities. Most of them are equipped with new communication interfaces that allows smartphones to carry out security-critical operations like access to sensitive personal data in social network applications.

Most of the applications running on these devices still use static alphanumeric passwords or PIN codes (semi-permanent or one-time) as a mean of authentication in accessing sensitive services/data or performing critical transactions, even though these methods are vulnerable to spyware attacks. Indeed, current mobile operating systems (e.g., Android, IoS, etc.) provide proper environments that allow developers to easily create applications and sell them through online marketplaces (e.g., GooglePlay, App Store, etc.). Users access marketplaces and choose the applications to install on their smartphones. Unluckily, this ecosystem also leads to the possibility for mobile malware to spread across online marketplaces and reach smartphones by fooling the user (e.g., a malware can pretend to be a fancy social network application or a popular game).

For these reasons, users may end up installing such applications without realizing that they may include spyware able to track all the activities and authentication transactions carried out in their devices. This can be done by using side-channel attacks [4], [5] or by more sophisticated forms of spyware that are able to record the entire authentication screen along with the user's touch coordinates [6], and then process the recorded data to steal the password and perform unwanted operations on the social platform without the user's awareness.

It is obvious that a spyware steals the user's credential to replay the recorded data and, thus, to gain unwanted access and/or perform specific actions without the user's consent. The methods to deal with such vulnerabilities can be classified into two categories. Approaches in the first category rely on preventing the spyware from stealing the user's credential, while approaches in the second category rely on preventing the spyware from replaying the recorded data.

Yi et al. [7] proposed PassWindow, an authentication method that use PIN digits and a pre-selected image called *Pass-icon* as the password. The basic idea behind this system is that the Pass-icon is displayed to the user with other randomly selected decoy icons on a graphical grid called Pass-Window. The user has to memorize the *pass-location* which is the location of pass-icon within the pass-window. Afterward, the virtual keypad in addition to the pass-window without its images appears in the center of the screen. To authenticate, the user has to move the pass-window on the virtual keypad by tilting it (thus using accelerometers) in such a way that the pass-location moves over the PIN. To enter each digit, the user has to cover the rear camera lens with a finger to hide the input. In this way, it prevents shoulder surfing attacks and increases the security against side channel attacks and one time recording attacks. However, this approach is vulnerable to multiple recording attacks and its user study shows that the authentication speed is very low.

Fig. 1. PassWindow, FakePIN and Kim et al.'s scheme.

Kim et al. [8] proposed a dummy-key based password authentication scheme, called FakePIN. In their scheme, the password consists of an alphanumeric text and a password direction as an additional secret value. During login, instead of directly inserting the original password, the user has to combine it with the password direction in order to fool the observer by pressing a fake dummy key value. Since the location of the keypad letters is changed randomly for each authentication, an observer fails to authenticate with the password acquired by shoulder-surfing or side channel attacks. However, an attacker can discover the original password by intersections between two sets of information acquired through recording attacks. Thus, the scheme is not resilient against multiple recording attacks.

Recently, Kim et al. [6] designed a recall-based graphical password for mobile devices, which is resilient to spyware attacks. Their approach is based on three elements: arrows in the same direction, the omission of authentication values and the inclusion of random errors. The user has to memorize the password's location in a 5×7 grid. During the login, the user selects cells according to the arrows displayed in each password's cell whereas the starting cell position changes randomly each time. This method ensures security against brute force attacks, smudge attacks, side-channel attacks and spyware-based recording attacks. However, while including errors increases the security of this scheme, their user study shows that it decreases its usability.

A common way to increase the security against automated attacks is to ask users to solve challenge-response tests such as CAPTCHA before allowing them to enter their PIN/password. The most widely-deployed form of CAPTCHA is text based, where distorted texts are shown as CAPTCHA images. A well-known example, designed by Ahn et al is ReCAPTCHA [9]. Their approach consists of using scanned words from old books that Optical Character Recognition (OCR) program failed to recognize. However, the hardest category of this scheme has been recently broken by Goodfellow et al. [10] using neural networks with an accuracy of 99.8 percent. In addition to this security issue, a recent research [11] pointed out that existing schemes of CAPTCHA, including reCAPTCHA, are not suitable for mobile devices. This is due to significant usability problems that frustrate users and lead to errors. In [12], authors suggested alternative input mechanisms aimed at improving the usability of ReCaptcha on smartphones. However, their user study results show that the participants prefer the existing ReCaptcha scheme, that uses the virtual keyboard as primary input.

Chow et al. [13] introduced the idea of presenting several textual CAPTCHAs into a grid of clickable CAPTCHAs. Their system does not rely on keyboard input, which can be particularly annoying on mobile devices. Instead, they ask the user to select some elements in the grid that match the challenge requirement. Despite showing some advantages, this scheme has not been widely deployed.

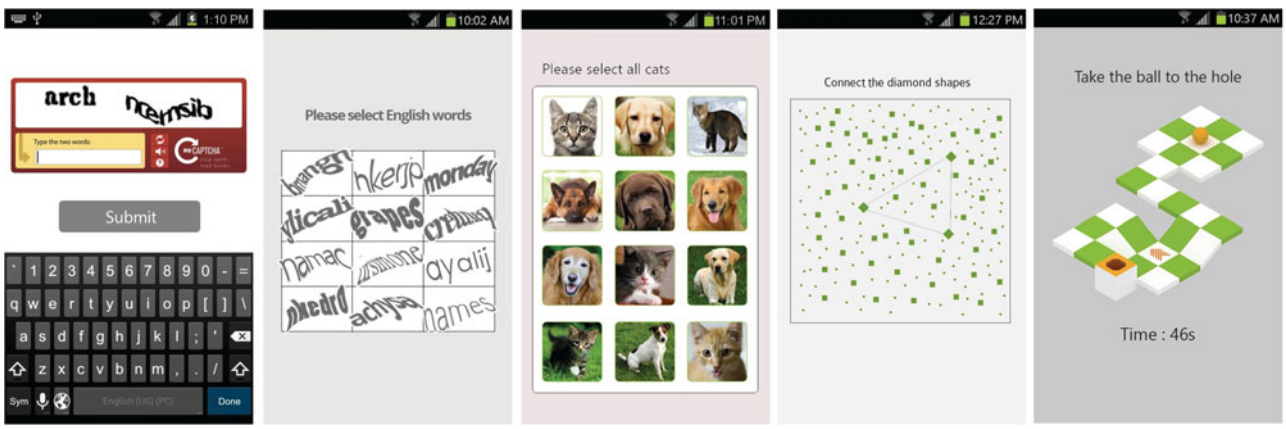

Fig. 2. ReCAPTCHA, Clikable CAPTCHA, Asirra, Drawing CAPTCHA and accCAPTCHA.

Pequegnot et al. [14] proposed an authentication mechanism based on a graphical Turing test. To authenticate, the user has to input the PIN code along with a secure code displayed in a CAPTCHA. This additional code is pseudo-randomly generated for each image by a secure element. This way, it prevents mobile malware from performing unwanted transactions without the user's agreement. However, this scheme is similar to existing commercial text-based CAPTCHAs, which add noise and distortion to the CAPTCHAs to make them harder to break. Nevertheless, all of them have been defeated with high percentages of accuracy through object-recognition techniques, see e.g., [10], [15], [16], [17], and [18]. In addition, using too much noise and distortions makes them hardly understandable by humans as well, especially on tiny screens.

An alternative to text-based CAPTCHA forms are image based CAPTCHAs. A typical CAPTCHA of this kind is Asirra [19], which displays 12 images of cats and dogs and asks users to select all cat images among them. Their user study shows that solving the Asirra challenge takes time under 30 seconds for 96.6 percent of humans which is advantageous compared with text-based CAPTCHAs. However, Golle [20] showed that this scheme is vulnerable to machine learning attacks. Shirali-Shahreza et al. [21] proposed CAPTCHA mechanism for mobile devices, called Drawing CAPTCHA. In this method, numerous dots are displayed on a screen with a noisy background. To pass the CAPTCHA challenge, the user has to connect specific dots to each other. It sounds straightforward, but it is not secure. In [22], an image processing technique was proposed to breaks the Drawing CAPTCHA with an accuracy of 75 percent.

Another form of CAPTCHA that has been introduced in recent years is game-based CAPTCHA. Liao et al. [23] proposed accCAPTCHA, a new CAPTCHA scheme for mobile device based on game logic and human recognition. In this scheme, the user is asked to play a simple rolling ball game or other well-known games (e.g., enigma, racing game, etc.). However, the user study shows that most of the games take a long time to pass the challenge (e.g., Stack game 47.3 sec, Rolling ball game 25.2 sec and Racing game 55 sec).

Recently, Guerar et al. [24] proposed a Completely Automatic Public *Physical* test to tell Computers and Humans Apart (CAPPCHA). In their scheme, the user is asked to tilt the device to a specific degree displayed on the screen and hold it still in this position for one second to have access to the PIN pad. Their security analysis shows that the proposed scheme is resilient against brute force attacks, side channel attacks and spyware-based recording attacks. However, this mechanism requires a secure element with an embedded accelerometer sensor which is not yet commonly available in the market.

Table 1 provides a recap of the basic methodologies as well as of the motivations and main limitations associated to the existing schemes. In the following we discuss BrightPass as a way to overcome the limitations of the current proposals.

3 Introducing the BrightPass Scheme

This section introduces the basics of BrightPass as well as the technological background required for its implementation on Android-based smartphones.

3.1 Secure Element

Secure Elements (SEs) are a combination of hardware, software, interfaces and protocols embedded in a mobile handset [25] that provide a secure platform enabling isolated execution for applications of different issuers and tamper-proof data storage. This ensures a high level of security and identity management to each application, network and user [26]. Secure Elements come in a variety of form factors. The most common are embedded Secure Elements (eSEs), Universal Integrated Circuit Cards (UICCs), and Secure Memory Cards (Secure Micro SD) [27]. An embedded SE is a smart card embedded into the device main board. An UICC is an advanced SIM card. A Secure Micro SD holds an embedded chip which can be used as a SE, along with a Flash memory.

Initially, secure elements were located only in the SIM, primarily due to operator control requirements and because the technology/process beyond them is almost identical. Indeed, SIM security and SE security are almost exactly the same. In more recent times, secure elements that are specific for NFC started appearing as "embedded" entities in the phone itself since the operators took too long to coordinate their actions in supporting NFC. Thus, device manufacturers in partnership with OS companies, started putting SEs in phones regardless of operator support. In parallel, some chip manufacturers started working on a secure and manageable memory space as part of the regular memory of the phone, initially called TrustZone and then Trusted Execution Environment (TEE). The TEE is a secure area that resides in the main processor of the phone and guarantees that sensitive data is stored, processed and protected within a totally trusted environment. Its ability to offer safe execution of authorized security software, known as trusted applications, enables the TEE to enforce protection, confidentiality, integrity and access rights on the data belonging to those trusted applications [28]. The TEE provides a more powerful processing speed capability and greater accessible memory space than a SE. In addition, it supports more granular user interface capabilities and peripheral connections than a traditional SE. However, the TEE can work together with a SE for providing specific functionalities such as the Trusted User Interface. In contrast, the SE supports physical robustness and high tamper resistance against side channel attacks; therefore, it is certifiable at the highest security levels (EAL4+) [29].

Fortunately, BrightPass does not rely on the SE hardware implementation. Additionally, it assumes that the mobile phone contains a SE which provides security support. In this paper, we implement the BrightPass application by using the Mobile Security Card (MSC) SE 1.0 issued by G&D.

3.2 PIN-Based Mobile Authentication Mechanism for Sensitive Operations

In order to highlight the weakness of PIN-based mechanisms to provide adequate security in mobile authentication, we take the example of a user experience for accessing a Twitter account that has been configured to require a second factor verification by adding his phone number (ideally associated to a device different to the one used for accessing) in its profile configuration. When logging into Twitter, immediately after entering the access password, the user is asked to enter a six-digit PIN verification code received via

TABLE 1
Methodology, Motivation and Weakness of Existing Authentication Methods

Schemes Approach	Methodology	Motivation	Weakness/ drawback
PassWindow [7]	1- The user memorizes the PIN digits and the location of a Pass-icon in the PassWindow 2- He moves the Pass-Window on the virtual keypad to enter the PIN.	- It prevents shoulder surfing attacks and increases the security against side channel attacks and one-time recording attacks.	- It takes a long authentication time (i.e., 17.86 seconds) - It is weak against multiple spyware based recording attacks
FakePIN [8]	1- The user chooses the alphanumeric text and a password direction. 2- He enters a fake dummy key values obtained through the combination of the alphanumeric text with the password direction.	- It prevents shoulder-surfing and side channel attacks	- Weak against multiple spyware-based recording attacks
Kim et al. [6]	1- The user chooses and memorizes the locations of a password in a 5×7 grid. 2- Then, he moves the finger from the "Start" position according to the arrows directions displayed in each password's cell	- It is resistant to multiple spyware-based recording attacks	- High error rates (i.e., 18%)
ReCAPTCHA [9]	1- The user recognizes the challenge which is a combination of an unknown word with a control word whose content is known. 2- The user input the two words on the keyboard. If he correctly recognizes the control word, it is assumed that his judgment about the other word is also valid.	- It prevents automated programs from abusing online services. - It helps to digitize books that are too illegible to be scanned by computers	- It has been broken with an accuracy of 99.8% - Not suitable for mobile devices
Clickable CAPTCHA [13]	1- The user recognizes distorted texts into a grid of clickable CAPTCHAs. 2- He clicks on the grid elements that match the challenge requirement	- It prevents automated programs. - The design text-based CAPTCHA is suitable for mobile devices.	- It requires a long time to solve the challenge (i.e., 18.2 seconds) and high error rates (i.e., 10-20%)
Asirra [19]	1- The user recognizes the cats among 12 images of cats and dogs. 2- Then he selects cats' images only	- It improves the usability of text-based CAPTCHA by using image instead of text.	- It is vulnerable to machine learning attacks
Drawing CAPTCHA [23]	1- The user recognizes specific dots displayed on a screen with a noisy background. 2- Then, he connects the specific dots to each other	- It provides an alternative to text-based CAPTCHA suitable for mobile devices.	- It has been broken with an accuracy of 75%
accCAPTCHA [22]	1- The user plays a simple rolling ball game or other well-known games (e.g., enigma, racing game, etc.).	- It enhances the security without annoying users	- It takes a long authentication time (e.g., Stack game 47.3 sec, Rolling ball game 25.2 sec and Racing game 55 sec)
CAPPCHA [24]	1- The user tilts the device to a specific degree displayed on the screen.	- It provides secure and usable alternative to CAPTCHA in order to improve the authentication on mobile devices.	- It requires specific hardware that is not yet commonly available in the market

SMS on the specified phone to confirm its identity. Once entered this security code, the user has full access to his Twitter session. However, the spyware that has gained root access to the mobile OS is able to steal the user's PIN code [5] as it is entered on the smartphone and thus, can trick the current session and carry out a successive login without the user's agreement. In order to keep the advantages of PIN authentication and increase its security against different spyware attacks, BrightPass introduces screen brightness and the *lie overhead* [30] concept to this common method.

3.3 Brightness as a Security Mechanism

During the design phase of BrightPass, we noticed that screen capture and screen recording techniques do not take the display brightness setting into account (i.e., a white pixel will come out as white in the screen captures regardless of the screen brightness level at which the screen was captured). The simplest but most effective test consisted in taking two screen captures when the brightness seekbar is displayed on the phone screen to show in which brightness level the screen capture has been token and then compare between them visually. Fig. 3 shows that even though the screen captures were taken at different values of brightness as the seekbar indicates (i.e., the brightness level, highlighted by blue color in the seekbar, is adjusted to a low value in Fig. 3a and to a high value in Fig. 3b), they look exactly the same in the pictures. In order to empirically demonstrate this, a comparative study between these screen captures was conducted by using the Mean Squared Error (MSE) algorithm [31]. The average squared difference between the pictures is computed pixel-by-pixel by adding up the squared differences of all the pixels and dividing it by the total pixel count. For screenshots

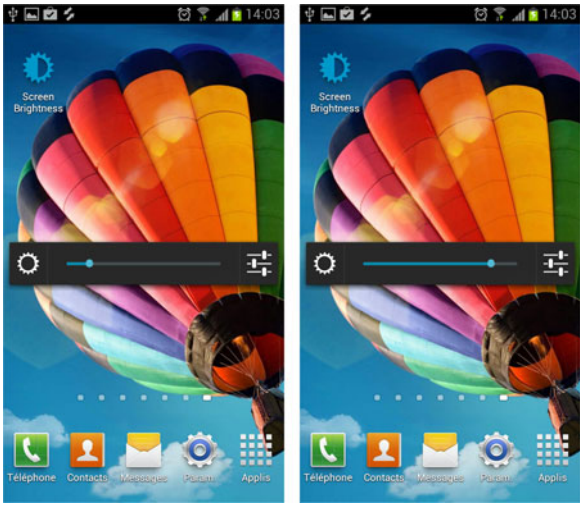

Fig. 3. Screen captures of an Android desktop at different levels of screen brightness, showing the seekbar adjusted respectively to a low (a) and a high (b) brightness value.

$A = \{a_1 \dots a_M\}$ and $B = \{b_1 \dots b_M\}$, where M is the number of pixels

$$MSE(A, B) = 1/M \sum_{i=1}^{M} (a_i - b_i)^2. \quad (1)$$

Results are shown in the Fig. 4. The area with black color means that we get zero as result and thus there is no difference between the screen captures in this area. The blue line means that we get a different value and thus the only difference between the screen captures is at the seekbar which is mainly used to illustrate that the screen captures were taken at different brightness values. This highlights the fact that the user notices the screen brightness change, whereas the mobile malware is unable to detect it by using screen capture or screen recording techniques. On the other hand, since the mobile malware that has gained root access to the mobile OS is able to access the system's brightness value easily through the Android API, we used the brightness of the BrightPass application without changing the system's brightness value. As the BrightPass application is stored secretly in the secure element, when the user interacts with the application to enter the PIN, the Android OS (and hence the rootkit) cannot access the screen [5], hence it cannot reveal the brightness of the BrightPass activity. In addition, the Android platform does not allow a service running in the background to access the window parameter (e.g., brightness value) of an activity. In order to prove this, we tried to create a service that is able to get the brightness value of the current activity (i.e., the BrightPass activity), which is running in the foreground. This was not possible due to the fact that the service does not have window, thereby resulting in the possibility for the service to access the current brightness level. This means that introducing brightness in our security mechanism, as a communication channel that is invisible to the mobile malware, provides adequate security against such attacks.

3.4 The BrightPass Concept

Achieving higher levels of security for mobile social network access on untrusted platforms requires to enhance

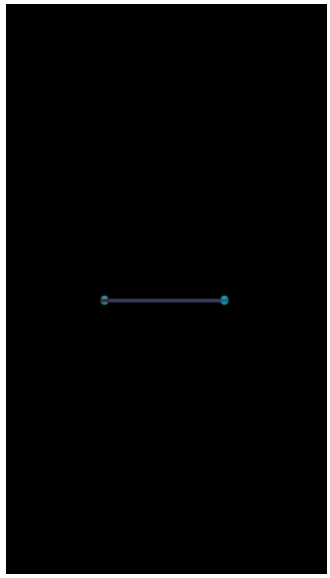

Fig. 4. The results of a comparison between the screenshots using the MSE algorithm.

and secure the classic widespread PIN authentication method. To this aim, we introduce the *lie overhead* concept. The idea consists in inserting a combination of the PIN digits and some misleading values, i.e., the lies. The order of the PIN digits' positions is randomly generated by the SE and then secretly shared with the user via an alternating circle's brightness displayed on the mobile device. If the circle's brightness value is high, the user must insert a correct PIN digit. Whenever it looks dark to the user, he is required to enter a misleading lie digit. In this way, only the legitimate user and the SE know the real PIN digits along with its positions in the currently generated sequence. Thereby, even if a mobile malware can steal the PIN by analyzing the differences between inputs recorded through repeated side-channel attacks [32], [33], [34], it will not be able to authenticate in the next operation. This is due to the randomization of PIN digit positions in the sequence generated for each authentication, and to the use of different screen brightness level each time. In our scheme brightness is adopted as a secure channel to secretly tell the user when to input a correct PIN digit and when to input a misleading lie digit.

A use case example of our Brightpass mechanism is described below. This sequence may be requested every time a user is going to perform an action in a sensitive context or once per session. The whole process is depicted in Fig. 5 and works as follows:

- When the user initiates the needed operation, a request is sent by the social network Service Provider (SP) to the smartphone and through the OS to the SE component possibly together with other information (e.g., SP identifiers, date, etc.).
- Then, the SE generates a random sequence of lies based on the lie overhead. An example sequence of lies generated by the SE for a four-digit PIN is 1101001, where 1 means displaying a circle with a high brightness value to tell the user to input the correct PIN digit and 0 means displaying a circle with a

Fig. 5. The proposed authentication mechanism.

low brightness value to tell the user to input a misleading lie digit.

- Circles are displayed sequentially on the phone's screen with their corresponding brightness.
- The user follows the circle's brightness to enter the PIN code. According to this example, if we assume that the real PIN is 3972, the user's input should be $39R7RR2$ where R means a random number from 0 to 9.
- When the user clicks the OK button to validate his action, the input is sent to the SE for verification and to make a decision on acceptance or rejection of the requested action. The different steps of the authentication method are summarized in Fig. 6.

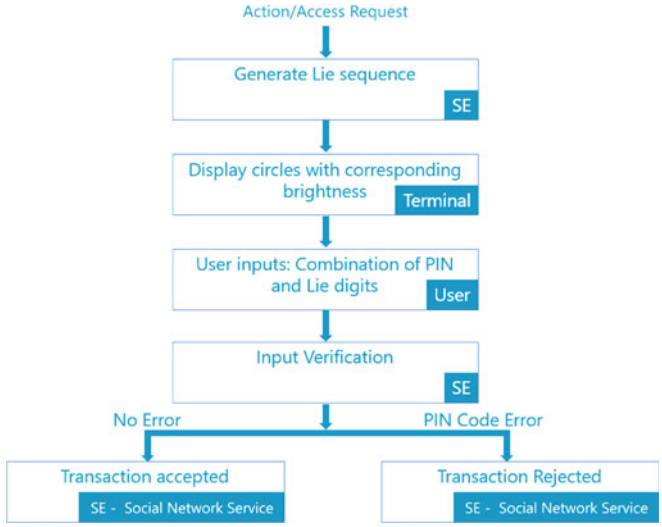

Fig. 6. BrightPass authentication process.

4 Security Analysis

In this section, the security of the BrightPass against Brute Force attacks, Dictionary attacks, Side Channel attacks, Spyware-based Recording attacks and smartphone theft is analyzed.

4.1 Brute Force and Dictionary Attacks

A Brute Force Attack is a password cracking method that uses an automated process to try all possible character combinations until the password is found. In contrast to this type of attack, a dictionary attack creates a dictionary that contains the most commonly used words as a password, and then tests all these words until either no word is left in the dictionary or until the password is found [35]. In the proposed scheme, the randomization of PIN digit positions in the sequence generated by the secure element for each authentication session leads the user to input a new authentication trial value each time. Similar to CAPTCHA test, this prevents the automated process of iterating through the entire password space and from testing all dictionary words. Each time, only one combination or a single dictionary word can be tested by this process to crack the current password with a success probability of 1 in 10,000, which is considerably low. BrightPass allows only three attempts before the SE is locked. Therefore, if the input is wrong, a new challenge must be regenerated by the SE and the number of remaining attempts is decremented.

4.2 Side Channel Attacks

Using a side-channel attack, spyware can steal the user's keystrokes even when the secure element ensures strong isolation to protect a sensitive input. This kind of attack uses the shared resources between the mobile OS and the secure element, such as the accelerometer [33], the camera and the microphone [5], the Gyroscope [32], [34], etc.

The proposed scheme has two elements for protection against side-channel attacks. The first element is the randomization of the PIN digit positions generated by the SE for each authentication. The second one is the use of screen brightness as a secure way to share these positions with the user. Since this scheme is the first work that uses screen brightness for mobile authentication, side-channel attacks through light sensor were not proposed in the literature. Nonetheless, in order to evaluate the security of our system, we classify these attacks according to the two categories and we discuss their potential impact on the security of BrightPass.

In the first category, we assume that the mobile malware uses the light sensor which is available in most modern smartphones as a side channel to acquire the PIN digit positions of the current authentication session. In this way, if the malware has acquired the PIN digits, it can perform unwanted actions without the user agreement. In order to prove that BrightPass is resilient against this side channel attack, we carried out a simple test in which we recorded the brightness value (in Lux) captured by the light sensor during the user authentication. To obtain accurate results that are not affected by the ambient light, we executed the test on a Galaxy-i9300 smartphone in a dark room. The results obtained are all equal to zero either when the circle brightness is adjusted to high or low, which confirms that the light sensor captures only the ambient light rather than the screen brightness. Therefore, the mobile malware is unable to acquire the PIN digit positions and thus, it cannot perform unwanted actions without the user's consent.

In the second category, we assume that the mobile malware predicts the user input through different side-channel attacks existing in the literature (e.g., [5], [32], [33], [34]). Using the combination of PIN digits with misleading lie ones increases the guessing entropy which mitigates side-channel attacks without affecting the memorability (i.e., the user has to memorize only a four-digit PIN). Furthermore, although the malware can catch the user's entire keystrokes

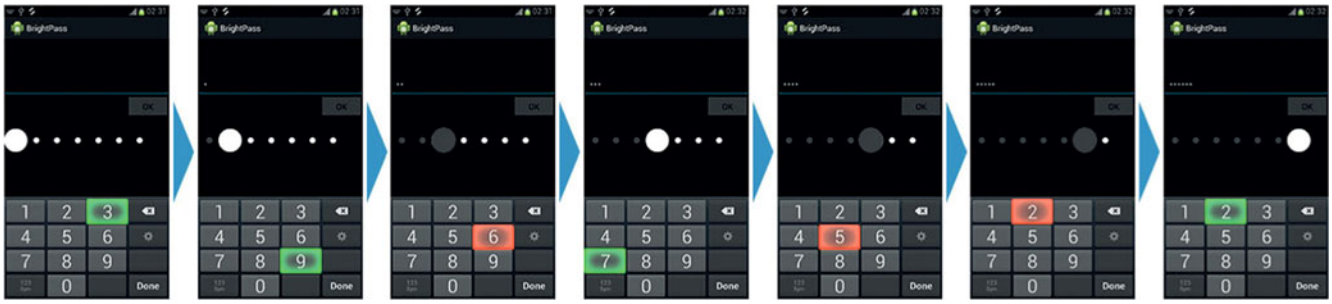

Fig. 7. An illustration of a BrightPass authentication session.

through these kinds of attacks, it still cannot differentiate between the PIN digits and the misleading lie ones. Instead, it can succeed to infer PIN digits by analyzing the differences between the recorded input from several authentication sessions. However, the knowledge of PIN digits without their positions in the current authentication session is not sufficient to authenticate successfully without the user's consent. Hence, this indicates that BrightPass may be resilient against side-channel attacks.

4.3 Spyware Based Recording Attacks

Mobile malware has taken a severe form with the introduction of recording attacks. This type of attack is not limited to leaking touch coordinates, but can also record the entire authentication screen [6] which makes hard to efficiently counter this attack. Let us assume that a smartphone is infected by such spyware. Fig. 7 shows a BrightPass authentication session, where the user inputs the PIN and lie digits according to the circle's brightness. A bright circle tells the user to input a correct PIN digit (highlighted in green) while a dim circle means to enter a misleading lie digit (highlighted in red). Note that we used the gray color to show the dim circles for presentation purposes, since the screenshot cannot capture the device's brightness as shown in Fig. 3.

Fig. 8 shows the same authentication session screenshots and input captured by a recording-based spyware, which is unable to capture the circle brightness and, therefore the spyware is unable to infer the lie sequence generated by the SE. This means that it cannot reveal the correct digits among the misleading lie ones though a one-time recording attack. Although it can infer the PIN digits by performing intersection between user's keystrokes recorded through multiple recording attacks, it cannot authenticate without the knowledge of the randomized positions in the next authentication session. Thus, the knowledge of PIN digits is useless. Hence, the proposed scheme provides adequate security against one-time and multiple spyware-based recording attacks.

4.4 Theft of Smartphone

BrightPass is designed to protect PIN code from automatic action/operation approval by malware. Since this is achieved using the screen brightness as a communication channel that is invisible to the mobile malware, it is obvious that humans can solve this challenge easily. However, if an attacker steals the user's mobile device, he is still unable to authenticate without the knowledge of the PIN code. In addition to that, BrightPass allows only three attempts to enter the PIN digits in the right position before the SE is locked. Thus, our security mechanism is secure against smartphone's theft unless the PIN has been previously stolen through different channels (e.g., side channel attacks) or the PIN is generated (one time PIN) by the smartphone itself through a proper application.

5 Comparison with Existing Approaches

In Section 2 we introduced several mobile authentication systems in which users have to remember different kinds of passwords. Some of them can withstand sophisticated forms of spyware that leak the entire authentication screen as well as the touch coordinates, while others are robust only against simple attacks that predict the user's input through side channels. However, few of them resist multiple recording attacks. We compare our system with Kim et al.'s (KHS for short) [6], Kim et al.'s (FakePIN) [8] and Yi et al.'s (PassWindow) [7] in terms of i) amount of data that the user is required to remember and ii) security strength. Detailed comparisons are listed in Tables 2 and 3.

Our system has the advantage of using common PIN-entry. The user has only to remember a four-digit PIN. From a security point of view, it is possible to observe that our system provides adequate security against different

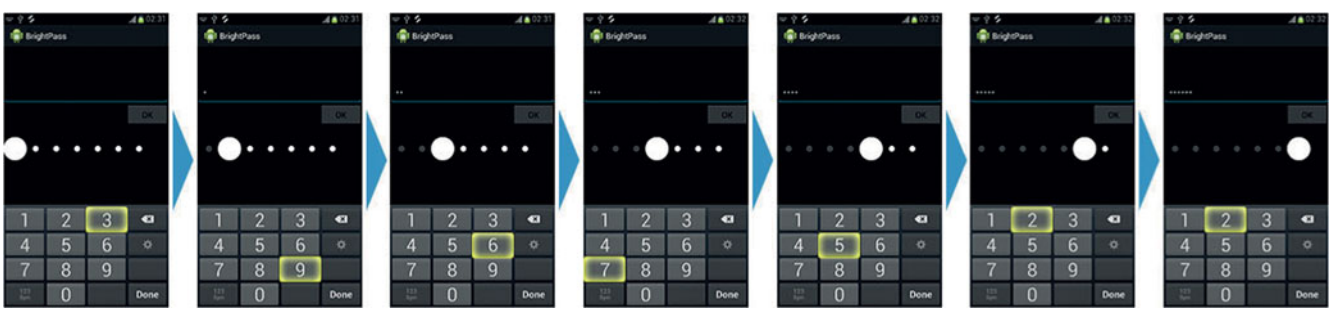

Fig. 8. Malware recorded screenshots during a BrightPass authentication session.

TABLE 2
Comparison of Password Memorability of Related Systems

Approach	KHS	FakePIN	PassWindow	BrightPass
Password number	x	√	√	√
Password icon	x	x	√	x
Password location	√	x	x	x
Password direction	x	√	x	x

TABLE 3
Comparison of the Security Strength of Related Systems

Approach	KHS	FakePIN	PassWindow	BrightPass
Resist side-channel attacks	√	√	√	√
Resist one-time recording attacks	√	√	√	√
Resist multiple recording attacks	√	x	x	√

TABLE 4
CogTool Measurement Results

Authentication method	Authentication time [s]
Kim et al.'s scheme [3]	10.72
FakePIN [5]	10.90
PassWindow [4]	18.12
BrightPass	8.2

spyware attacks without an additional secret that a user has to remember. In the FakePIN scheme, the user has to memorize a four-digits PIN in addition to four directions. In the PassWindow scheme the user has to remember four-digit PIN in addition to a preselected image called *Pass-icon*. Despite the fact that FakePIN and PassWindow increase the amount of data that has to be remembered by the user, it is still insecure against an attack using multiple recordings. Similar to our system, Kim et al.'s scheme [6] resists different spyware attacks.

6 Experimental Evaluation

In this section, we analyze and compare the usability of BrightPass with some existing authentication schemes using two different experiments. The first experiment is based on a user interface evaluation tool while the second one is based on an experiment based on actual users. The times and error rates of the existing authentication schemes are extracted from the original publications and may not have been calculated in exactly the same way for each scheme. They do, however, provide a basis for a general comparison.

6.1 GOMS Model Test

GOMS [36] is the most commonly used cognitive modeling technique to evaluate usability. We used an evaluation tool, called CogTool [37], [38], which is based on this model to predict the authentication time of BrightPass. As shown in Table 4, the test result estimated by CogTool 1.2.2 is 8.2 seconds, which is the fastest result compared with existing schemes. Although FakePIN and PassWindow are similar to the proposed scheme since they are based on a four-digit PIN, their test showed slow results. The reason is that they require a more complicated password input process than BrightPass.

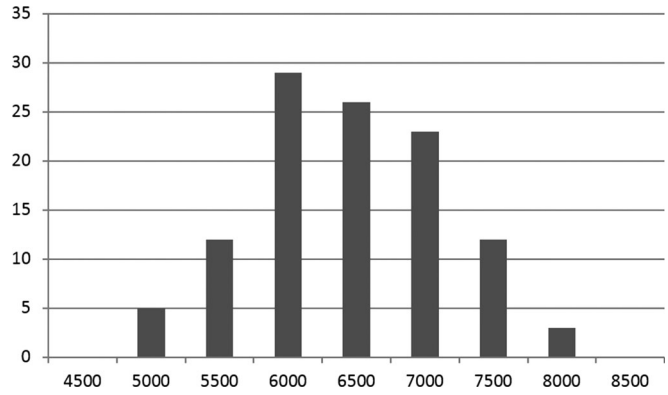

Fig. 9. Distribution of authentication times (milliseconds).

TABLE 5
User Test Results for the Existing Schemes

Authentication method	Authentication time [s]	Error rate [%]
Kim et al.'s scheme [3]	11.47	18
FakePIN [5]	14.13	4.70
PassWindow [4]	17.86	4
BrightPass	6.73	1.81

6.2 User Test

Regarding the experiment with actual users, BrightPass has been implemented using the following development tools: Eclipse KEPLER SR2, Android SDK 4.04 and JAVA 1.7.0. The test equipment consisted of a Galaxy-i9300 smartphone (1.4 GHz Dual-Core CPU, 1 GB RAM) equipped with a Giesecke & Devrient Mobile Security Card (MSC) SE 1.0. This MSC consists of a secure smart card that conforms to the Common Criteria EAL 5+ security level on top of a standard flash memory mass storage. It has 2 GBytes of memory, and uses version 5.0 of the Sm@rtCafé Expert smart card operating system, which has a common criteria EAL 4+ and is certified by Germany's Federal Office for Information Security (BSI) [39]. Moreover, it incorporates a crypto controller and supports multiple applications as well as the complete Java Card API and the Global Platform API. The communication between the third-party application and the MSC SE 1.0 is carried out via the Seek-for-Android API.

The study was conducted on 22 participants with an average age of 23 years (range: 17-25). At the beginning, detailed explanations were given to each participant together with a random PIN. They were asked to train with the prototype until they felt familiar with the system. Due to the popularity of PIN, most participants did not have to perform more than one test. Afterwards, each participant repeated the authentication process five times. Thus, the measured authentication time and error rates are based on 110 authentication sessions performed by 22 participants. Authentication time was measured from the first key press to releasing the OK button. The histogram in Fig. 9 shows the distribution of authentication times. Table 5 shows the average authentication time and error rate of BrightPass along with three schemes from the first category mentioned

TABLE 6
User Test Results of the Existing Schemes

Authentication method		Avg. challenge time [s]	Error rate [%]
Assira [16]		15 - 20	-
Clickable captcha [10]		18.2	10 - 20
AccCaptcha [19]	- Stack Game	47.3	33
	- Rolling ball game	25.2	22
	- Racing game	55	4

in the related work. Table 6 summarizes the average challenge time and the error rate of existing schemes.

By comparison it is possible to notice that BrightPass has the fastest authentication time and the lower error rates among all the evaluated schemes. The reason is that it uses common four-digit PINs without adding additional secrets value with complicated input processes nor it requires users to solve complex cognitive tasks. The user has only to check the circle brightness to enter the PIN along with random lie digits. In this way, it improves the security while remaining fast to use, similar to traditional PIN input. One among the three necessary elements that has been used in the Kim et al.'s [6] scheme to increase the security against spyware-based recording attacks is the included errors. However, their user study showed that this element leads to an increased error rate. Thus, in their scheme, the user has to choose between security and usability. Unlike Kim et al's scheme, FakePIN and PassWindow have significantly lower error rates but they took longer to authenticate. Most of the existing CAPTCHA schemes are known by their usability problems that frustrate users and lead to errors, which explain the high value of average challenge time and error rate of the mentioned CAPTCHA scheme in the Table 6.

7 Conclusion

Nowadays, mobile access to social networks has become very popular in many sectors of our society, due to the huge amount of personal data and useful information aggregated and made available by applications such as Facebook, Twitter, Google Plus, and so on, that can be seen as the on-line interfaces of our own lives. However, since such data may contain and/or expose very sensitive information that can be prone to several kind of misuses (personal data leakage, identity theft etc.), controlling the access to these facilities through proper authentication procedures is now a fundamental challenge. However, the authentication phase is often considered as the weakest element in mobile access security due to the increase of malware threats that are able to track and capture the secure codes entered by the users.

This work introduced a novel authentication method (BrightPass) that prevents malware from being able to compromise mobile access to social network and subvert user-authenticated operations. The proposed scheme uses screen brightness as a secure communication channel to communicate a random sequence generated by the secure element to the user. This sequence is used to tell the user when to input correct PIN digits and when to input misleading lie digits. Thus, the user authenticates with a new trial for each authentication. The security analysis shows that the proposed scheme is resilient against brute force attacks, dictionary attacks, side channel attacks and spyware based recording attacks. From a usability point-of-view, the results of our experiments suggest that the proposed scheme offers a short authentication time and low error rates. Thus, it increases the security while maintaining good usability properties in the social scenario.

The comparison with existing schemes which are resilient to multiple recording attacks shows that BrightPass has similar security strength with considerably lower authentication time and error rates. Therefore, this technology may introduce a positive impact in the social networking environment by changing the associated business dynamics, together with the way of accessing and publishing information on social media, with the obvious consequences in the political and professional sectors, that are extremely dependent on such media. Finally, it should be considered that the same mechanism can be used also to secure transactions protected by PIN verification codes in electronic payment applications involving contactless proximity payment systems [27], when NFC-enabled smartphones are used in conjunction with NFC POS terminals. This can be, obviously, the subject of a future work.

Acknowledgements

Authors gratefully thank Dr. Alexander de Luca for his valuable suggestions and comments.

References

[1] L. Caviglione, M. Coccoli, and A. Merlo, "A taxonomy-based model of security and privacy in online social networks," *Int. J. Comput. Sci. Eng.*, vol. 9, no. 4, pp. 325–338, 2014. Doi: 10.1504/IJCSE.2014.060717.

[2] E. Ikhalia and C. O. Imafidon, "The need for two factor authentication in social media," in *Proc. Int. Conf. Future Trends Comput. Commun.*, 2013, pp. 76–82.

[3] H. Kim, J. Tang, and R. Anderson, "Social authentication: Harder than it looks," in *Financial Cryptography and Data Security*. Berlin, Germany: Springer, 2012, pp. 1–15.

[4] A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith, "Practicality of accelerometer side channels on smartphones," in *Proc. 28th Annu. Comput. Secur. Appl. Conf.*, 2012, pp. 41–50. Doi: 10.1145/2420950.2420957.

[5] L. Simon and R. Anderson, "PIN skimmer: Inferring PINs through the camera and microphone," in *Proc. 3rd ACM Workshop Secur. Privacy Smartphones Mobile Devices*, 2013, pp. 67–78. Doi :10.1145/2516760.2516770.

[6] T. Kim, J. H. Yi, and C. Seo, "Spyware resistant smartphone user authentication scheme," *Int. J. Distrib. Sensor Netw*, vol. 2014, 2014, Art. no. 7. Doi:10.1155/2014/237125.

[7] H. Yi, Y. Piao, and J. H. Yi, "Touch logger resistant mobile authentication scheme using multimodal sensors," in *Advances in Computer Science and its Applications*, vol. 279. Berlin, Germany: Springer, 2014, pp. 19–26.

[8] S. Kim, H. Yi, and J. H. Yi, "FakePIN: Dummy key based mobile user authentication scheme," in *Ubiquitous Information Technologies and Applications*, vol. 280. Berlin, Germany: Springer, 2014, pp. 157–164.

[9] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: Human-based character recognition via web security measures," *Science*, vol. 321, pp. 1465–1468, Sep. 2008.

[10] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *Proc. Int. Conf. Learning Representations*, *arXiv preprint arXiv:1312.6082*, 2014.

[11] G. Reynaga and S. Chiasson, "The usability of captchas on smartphones," in *Proc. SECRYPT*, 2013, pp. 427–434.

[12] G. Reynaga, S. Chiasson, and P. C. van Oorschot, "Exploring the usability of CAPTCHAS on smartphones: Comparisons and recommendations," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Feb. 2015, pp. 8–11.
[13] R. Chow, Golle, P. M. Jakobsson, X. Wang, and L. Wang, "Making CAPTCHAs clickable," in *Proc. 9th Workshop Mobile Comput. Syst. Appl.*, Feb. 2008, pp. 25–26.
[14] D. Pequegnot, C.-L. Lamy, A. Thomas, T. Tigeon, J. Iguchi-Cartigny, and J.-L. Lanet, "A security mechanism to increase confidence in M-transactions," in *Proc. 6th Int. Conf. Risks Secur. Internet Syst.*, 2011, pp. 9–16. Doi:10.1109/CRiSIS.2011.6061836.
[15] G. Moy, N. Jones, C. Harkless, and R. Potter, "Distortion estimation techniques in solving visual CAPTCHAs," in *Proc. IEEE Comput. Society Conf. Comput. Vision Pattern Recognition*, 2004, pp. 23–28.
[16] A. A. Chandavale, A. M. Sapkal, and R. M. Jalnekar, "Algorithm to break visual CAPTCHA," in *Proc. 2nd Int. Conf. Emerging Trends Eng. Technol.*, 2009, pp. 258–262.
[17] J. Yan and A. S. El Ahmad, "A Low-cost Attack on a Microsoft CAPTCHA," in *Proc. 15th ACM Conf. Comput. Commun. Secur.*, Feb. 2008, pp. 543–554.
[18] M. Korakakis, E. Magkos, and Ph. Mylonas, "Automated CAPTCHA solving: An empirical comparison of selected techniques," in *Proc. 9th IEEE Int. Workshop Semantic Social Media Adaptation Personalization*, 2014 , pp. 44–47.
[19] J. Elson, J. R. Douceur, J. Howell, and J. Saul, "Asirra: A CAPTCHA that exploits interest-aligned manual image categorization," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 366–374.
[20] P. Golle, "Machine learning attacks against the Asirra CAPTCHA," in: *Proc. 15th ACM Conf. Comput. Commun. Secur.*, 2008, pp. 535–542.
[21] M. Shirali-Shahreza and S. Shirali-Shahreza, "Drawing CAPTCHA," in *Proc. 28th Int. Conf. Inf. Technol. Interfaces*, 2006, pp. 475–480.
[22] R. Lin, S. Huang, G. B. Bell, and Y. Lee, "A new CAPTCHA interface design for mobile devices," in *Proc. 12th Australasian User Interface Conf.*, 2011, pp. 3–8.
[23] C. J. Liao, C. J. Yang, J. T. Yang, H. Y. Hsu, and J. W. Liu, "A game and accelerometer-based CAPTCHA scheme for mobile learning system," in *Proc. World Conf. Edu. Multimedia Hypermedia Telecommun.*, 2013, pp. 1385–1390.
[24] 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 *Proc. Int. Conf. High Performance Comput. Simulation*, 2015, pp. 203–210.
[25] B. Choudhary and J. Risikko, "Mobile financial services business ecosystem scenarios & consequences," Mobey Forum Mobile Financial Services Ltd, 2006.
[26] Simalliance White Paper, Secure Authentication for Mobile Internet Services, Critical Considerations V1.1, 2011.
[27] EMV Mobile Contactless Payment, Technical Issues and Position Paper, 2007.
[28] G. Bernabeu, "Accessing GlobalPlatform secure component from a web application," *Web Cryptography Next Steps, W3C Workshop on Authentication, Hardware Tokens and Beyond*, Silicon Valley (Mountain View), California, pp. 10–11, Sep. 2014, http://www.w3.org/2012/webcrypto/webcrypto-next-workshop/papers/webcrypto2014_submission_34.pdf
[29] Global Platform's White Paper, The Trusted Execution Environment: Delivering Enhanced Security at a Lower Cost to the Mobile Market, 2011.
[30] De A. Luca, E. von Zezschwitz, and H. Hussmann, "Vibrapass: Secure authentication based on shared lies," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2009, pp. 913–916.
[31] Z. Wang, A. C. Bovik, "Mean squared error: love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009. Doi: 10.1109/MSP.2008.930649.
[32] L. Cai and H. Chen, "TouchLogger: Inferring keystrokes on touch screen from Smartphone motion," in *Proc. 6th USENIX Conf. Hot Topics Secur.*, 2011, pp. 9–10.
[33] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "ACCessory: Password inference using accelerometers on smartphones," in *Proc. 12th Workshop Mobile Comput. Syst. Appl.*, 2012, pp. 1–6. Doi:10.1145/2162081.2162095.
[34] Z. Xu, K. Bai, and S. Zhu, "Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proc. 5th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, 2012, pp. 113–124. Doi:10.1145/2185448.2185465
[35] M. Raza, M. Iqbal, M. Sharif, and W. Haider, "A survey of password attacks and comparative analysis on methods for secure authentication," *World Appl. Sci. J.*, vol. 19, no. 4, pp. 439–444, 2012.
[36] S. K. Card, T. P. Moran, and A. Newell, *The Psychology of Human-Computer Interaction*, Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1983.
[37] B. E. John, K. Prevas, D. D. Salvucci, and K. Koedinger "Predictive human performance modeling made easy," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2004, pp. 455–462.
[38] L. Teo, John, B. E. John, and M. H. Blackmon, "CogTool-Explorer: A model of goal-directed user exploration that considers information layout," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2012, pp. 2479–2488.
[39] S. Waldenmaier, "G&D makes mobile terminal devices even more secure with new version of smart card in microSD format," *Press release*. Munich, Germany, 2010.

Meriem Guerar received the master's degree in information systems and networks from the University of Sciences and the Technology of Oran, Algeria, in 2011, where she is currently working toward the PhD degree. Her main research interests include the areas of authentication and identity management, security and usability, smartphone security, and payment systems security.

Mauro Migliardi received the PhD degree in computer engineering, in 1995. He was a research associate and assistant professor with the University of Genoa and research associate, Emory University; currently he is an associate professor with the University of Padua, adjunct professor with the University of Genoa, member of the Steering Committee of the Center for Computing Platforms Engineering and of the Scientific Board of Circle Garage s.r.l. start-up. His main research interests include distributed systems engineering, with a focus on security, pervasive systems, human memory support services, and energy awareness. He has won the 2013 Canada-Italy Innovation Award, tutored more than 100 among Bachelor, Master and PhD students at the Universities of Genoa, Padua and Emory, and (co-)authored more than 120 scientific papers.

Alessio Merlo received the MSc degree in computer science from the University of Genova, in 2005. He received the PhD degree in computer science from the University of Genova, Italy, in 2010 where he worked on performance and access control issues related to Grid Computing. He is currently serving as an assistant professor with the Universita degli Studi di Genova, Italy, where he collaborates in the CSec Lab, DIBRIS. His currently research interests include performance and security issues related to web, distributed systems (Grid, Cloud), and mobile (Android platform). He participates to program committees of international conferences and is member of the Editorial Board of an International Journal (*Journal of High Speed Networks*). He is a member of the IEEE Computer Society and the ACM.

Mohamed Benmohammed received the PhD degree from the University of Sidi Bel-Abbes, Algeria, in 1997. He is a professor of computer science with the University of Constantine 2, Algeria. His research interests include: microprocessors, computer architectures, embedded systems, and computer networks.

Francesco Palmieri received the MS and PhD degrees in computer science from the University of Salerno. Currently, he is an associate professor with the University of Salerno. Previously, he has been an assistant professor with the Second University of Naples, and the director in the Telecommunication and Networking Division, Federico II University, Naples. He has been closely involved in the development of the Internet in Italy as a senior member of the Technical-Scientific Advisory Committee and of the CERT of the Italian NREN GARR. His research interests include advanced networking protocols and architectures as well as network security. He has published a significant number of papers (more than 100) in leading technical journals, books, and conferences, and currently serves as the editor-in-chief of an international journal the *Journal of High Speed Networks* and is part of the Editorial Board (associate editor) of several other indexed ones (e.g., Applied Soft Computing, Soft Computing, Mobile Information Systems, etc.). He also achieved several formal appreciations and service awards for the organization of international conferences and scientific events, where he covered several key roles (Program Chair, Honorary General Chair, Workshop Chair, etc.).

Aniello Castiglione (S'04-M'08) received the PhD degree in computer science from the University of Salerno, Italy. Actually he is an adjunct professor with the University of Salerno, Italy, and with the University of Naples "Federico II", Italy. He received the Italian national qualification as associate professor of computer science. He published more than 130 papers in international journals and conferences. He served as program chair and TPC member in around 90 international conferences. One of his papers has been selected as "Featured Article" in the IEEE Cybersecurity initiative. He served as a reviewer for several international journals and he is the managing editor of two ISI-ranked international journals. He acted as a guest editor in several journals and serves as editor in several editorial boards of international journals. His current research interests include information forensics, digital forensics, security and privacy on cloud, communication networks, and applied cryptography. He has been involved in forensic investigations, collaborating as a consultant with several law enforcement agencies. He is a member of several associations, including IEEE and ACM.