Future Generation Computer Systems 82 (2018) 617–630

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Completely Automated Public Physical test to tell Computers and Humans Apart: A usability study on mobile devices

Meriem Guerar[a], Alessio Merlo[b], Mauro Migliardi[c,*]

[a] *University of Sciences and Technology of Oran Mohamed Boudiaf, Algeria*
[b] *DIBRIS, University of Genoa, Italy*
[c] *DEI, University of Padua, Italy*

HIGHLIGHTS

- We provide an extended discussion of the methodologies dedicated to cull out automated malicious attacks to authentication mechanisms.
- We provide an extended usability comparison both in terms of time needed to complete the challenge and in terms of error rates (false negatives, false positives are not possible in our scheme).
- We provide an extensive survey of our mechanism usability adopting the same methodology that has been used in the literature to evaluate the usability of CAPTCHAs.
- We discuss the results of our survey and through them we show that our mechanism is both secure and usable.

ARTICLE INFO

Article history:
Received 29 February 2016
Received in revised form
15 January 2017
Accepted 8 March 2017
Available online 10 March 2017

Keywords:
Security
Mobile
Usability
CAPTCHA
CAPPCHA

ABSTRACT

A very common approach adopted to fight the increasing sophistication and dangerousness of malware and hacking is to introduce more complex authentication mechanisms. This approach, however, introduces additional cognitive burdens for users and lowers the whole authentication mechanism acceptability to the point of making it unusable. On the contrary, what is really needed to fight the onslaught of automated attacks to users data and privacy is to first *tell human and computers apart* and then distinguish among humans to guarantee correct authentication. Such an approach is capable of completely thwarting any automated attempt to achieve unwarranted access while it allows keeping simple the mechanism dedicated to recognizing the legitimate user. This kind of approach is behind the concept of *Completely Automated Public Turing test to tell Computers and Humans Apart* (CAPTCHA), yet CAPTCHA leverages cognitive capabilities, thus the increasing sophistication of computers calls for more and more difficult cognitive tasks that make them either very long to solve or very prone to false negatives.

We argue that this problem can be overcome by substituting the cognitive component of CAPTCHA with a different property that programs cannot mimic: the physical nature. In past work we have introduced the *Completely Automated Public Physical test to tell Computer and Humans Apart* (CAPPCHA) as a way to enhance the PIN authentication method for mobile devices and we have provided a proof of concept implementation. Similarly to CAPTCHA, this mechanism can also be used to prevent automated programs from abusing online services. However, to evaluate the real efficacy of the proposed scheme, an extended empirical assessment of CAPPCHA is required as well as a comparison of CAPPCHA performance with the existing state of the art. To this aim, in this paper we carry out an extensive experimental study on both the performance and the usability of CAPPCHA involving a high number of physical users, and we provide comparisons of CAPPCHA with existing flavors of CAPTCHA.

1. Introduction

Nowadays, mobile devices are used to carry out activities that use sensitive personal information that should be accessed only by the legitimate device owner. To protect such information, most of mobile devices support PINs as a way to allow a quick, but weakly

[*] Corresponding author.
E-mail addresses: meriem.guerar@univ-usto.dz (M. Guerar), alessio.merlo@unige.it (A. Merlo), mauro.migliardi@unipd.it (M. Migliardi).

http://dx.doi.org/10.1016/j.future.2017.03.012
0167-739X/

secure, user's authentication mechanism. PINs are both easy to use and time-savings as they require users to remember and digit a very short number sequence, made by four to six digits at most. The key space of these PINs could be still adequate for humans, but not for malware, since they can easily carry out brute force attacks to PIN-based authentication mechanisms, by pretending to be the legitimate user that taps the PIN on the virtual keyboard.

The awareness of this problem has lead some researchers to introduce a new dimension to the authentication taxonomy. This new dimension adopts *Completely Automated Public Turing test to tell Computers and Humans Apart* (CAPTCHA) as an additional preliminary step to recognize the human user before allowing her to insert the PIN digits. The rationale for this additional security step derives from the observation that PIN-based mechanisms, as well as other mobile authentication approaches (i.e., Unlock patterns), have been devised to perform a *human vs. human* selection; hence the need to cull automated programs (or bots) before they may be able to perform the actual attack while being confused with human users by the system. Nonetheless, even if the idea to filter out automated attacks before using a traditional authentication mechanism is effectively targeting the actual nature of the threats, the root of the CAPTCHA concept, that is, the Turing test, targets the difference in cognitive capabilities between humans and computers. Therefore, even if the CAPTCHA concept introduces a new class of authentication methods that is orthogonal to all the ones previously defined (i.e., authentication methods capable of performing a *human vs. automated system separation*), the discerning factor adopted (i.e., a cognitive task) makes CAPTCHAs very often inconvenient and hard to solve even for human subjects. This inconvenience is quickly turning into a lack of usability as the sophistication of programs and the computational power of computers allows them to tackle more and more complex cognitive tasks.

To tackle the open issues related to CAPTCHA, we propose to adopt as the discerning factor telling computers and humans apart not intelligence or cognitive capabilities in general but the physical nature. In the past, computers input from the users was limited to keyboard and mouse, hence the measurement of physical parameters was not convenient; however, the current generation of mobile devices is endowed with a number of sensors capable of capturing several different kinds of physical interaction. For this reason, we argue that it is possible to leverage these sensors to simply tell human and computers apart on the basis of their capability (or inability) to physically interact with the device.

Leveraging smartphones and tablet computers sensors to devise new security mechanisms capable of separating humans from computers is not new; however, in all the past cases, the mechanisms required the users to perform some sort of cognitive task that actually made it an implementation of the traditional Turing test (CAPTCHA). We claim that, in order to tell humans and computers apart, it is possible to leverage the mere physical nature of human subjects without requiring them to tackle a complex cognitive task. Such a test can be defined as a *Completely Automated Public* ***Physical*** *test to tell Computers and Humans Apart* (CAPPCHA). An example implementation of CAPPCHA [1] requires the user to arrange the mobile device in a specific position for being recognized as a human. The rationale is that malware cannot physically move the device. For this reason, if the sensor that recognizes the position (e.g., the accelerometer) is secured (e.g., it is hosted in a Secure Element (SE)), a malware can neither simulate a fake position for the device, nor it can physically move the device.

We claim that CAPPCHA is able to separate humans from computers while maintaining a high level of usability and guaranteeing resilience to automated attacks. Furthermore, CAPPCHA can be used in a twofold mode: (1) in combination with a PIN authentication scheme to allow only human to access the PIN challenge, or (2) it can be used alone, acting as a standard CAPTCHA (e.g., before posting comments on a blog, voting or sending an email).

To prove the efficacy and ease of use of CAPPCHA, we have devised two practical implementations of a sensor-based CAPPCHA and we have combined them with a simple PIN based authentication. Here, our contribution is the discussion of the feasibility of CAPPCHA and a comparison in terms of performance and usability with Password-based authentication methods and CAPTCHA schemes available in literature. Regarding performance, we compare existing mechanisms with CAPPCHA in terms of test completion times and error rates. In terms of usability, we carry out and discuss an extended study (based on more than 200 actual users) of the CAPPCHA scheme. Our findings show that CAPPCHA is a promising solution as it is extremely easy to understand and use, and it shows a high acceptance rate by the users. Furthermore, it provides very high resilience to automated attacks and it does not introduce cognitive overload on the users.

The paper is structured as follows: in Section 2 we provide an overview of the related work; in Section 3 we describe the CAPPCHA scheme; in Section 4 we provide a security analysis of the proposed scheme against some well-known types of attacks; in Section 5 we describe the experimental procedure, the questionnaire we proposed to the users and we discuss our findings; finally, in Section 6 we draw our conclusions and point out potential directions for future work.

2. Related work

Albeit static alphanumeric passwords and PIN codes are vulnerable to spyware attacks [2], most of the current commercial applications still adopt these methods to authenticate the user in sensitive transactions. Current approaches to enhance the security of these widely adopted methods can be divided into two categories, namely *Password-based* and *CAPTCHA-based* authentication methods.

Password-based authentication methods. Proposals in this category may rely either on requiring an additional secret value to remember or on adding some overhead to the classic PIN/password authentication method. Kim et al. [3] proposed a password authentication scheme, called FakePIN (Fig. 1, left). In their scheme, the user has to memorize an alphanumeric text and a password direction as an additional secret value. To authenticate, the user is asked to input a fake dummy key value which results from the combination of the original password with the password direction. Whereas, the location of the keypad letters is changed randomly at each authentication request in order to prevent the attacker from replaying the user input acquired by shoulder-surfing or side channel attacks. However, an attacker can discover the original password by intersecting two sets of data acquired through recording attacks. Thus, this scheme is not resilient against multiple recording attacks. Similar to FakePIN, Yi et al. [4] propose PassWindow (Fig. 1, right), thereby adding an additional secret value, called *pass-icon*, to allow the user to enter their PIN in a secure way. The user has to memorize the location of the pass-icon within a window. The authentication is performed by moving the window on the virtual keypad using multimodal sensors in such a way that the location of pass-icon moves over the PIN. In this way, this mechanism increases the security of PIN codes against side channel attacks and one time recording attacks. However, it is vulnerable to multiple recording attacks and its user study shows that the authentication speed is very low (i.e., 17,86 s on average). Recently, Guerar et al. [5] proposed a secure authentication mechanism against mobile malware, called BrightPass (Fig. 2, left). To authenticate, the user has to follow the alternating circle's brightness displayed on the mobile device to enter his PIN code. He should insert a correct PIN digit when the

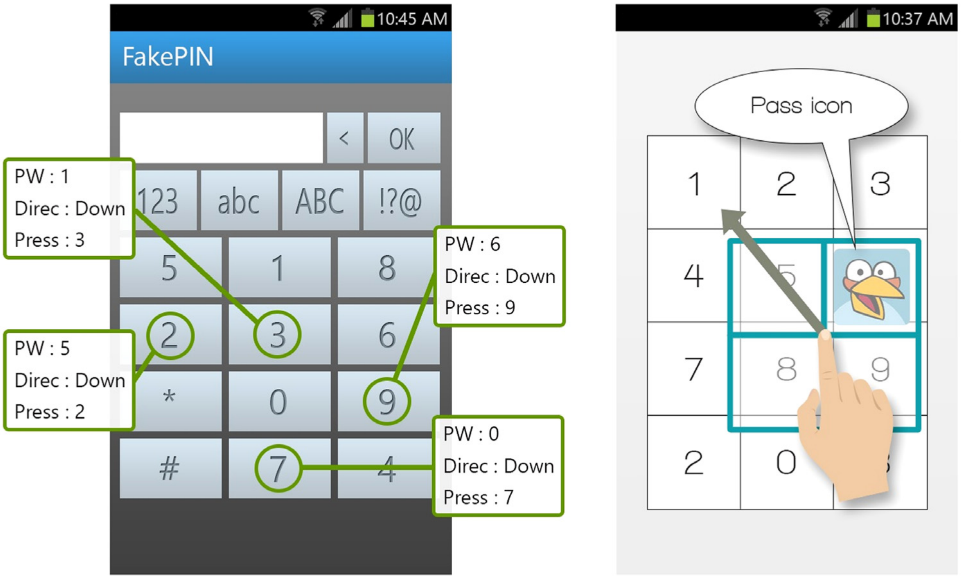


Fig. 1. FakePIN and PassWindow scheme.

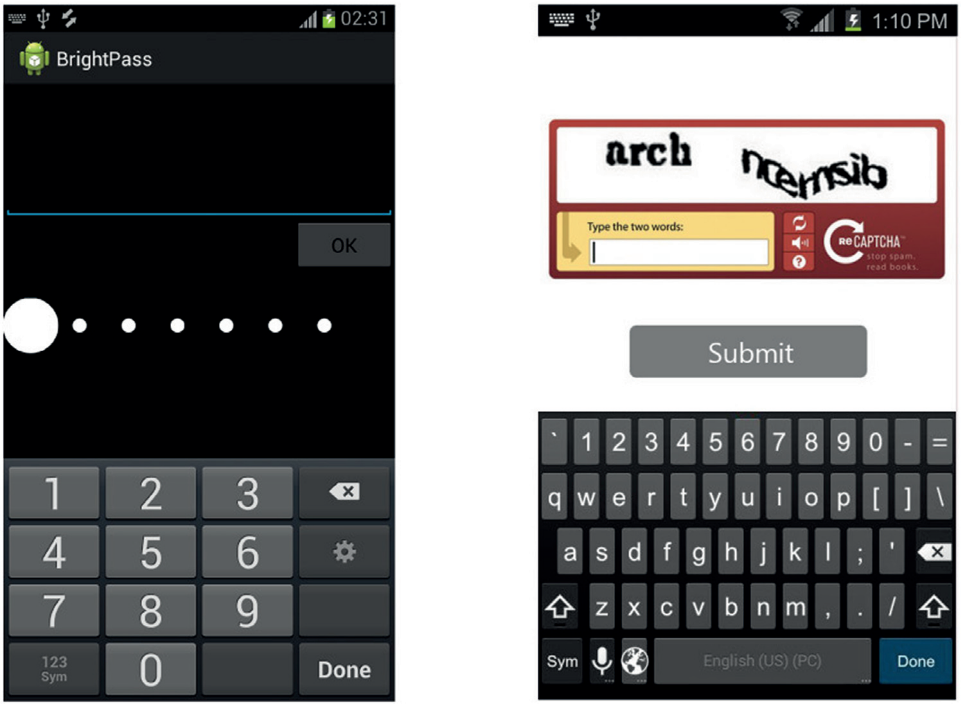
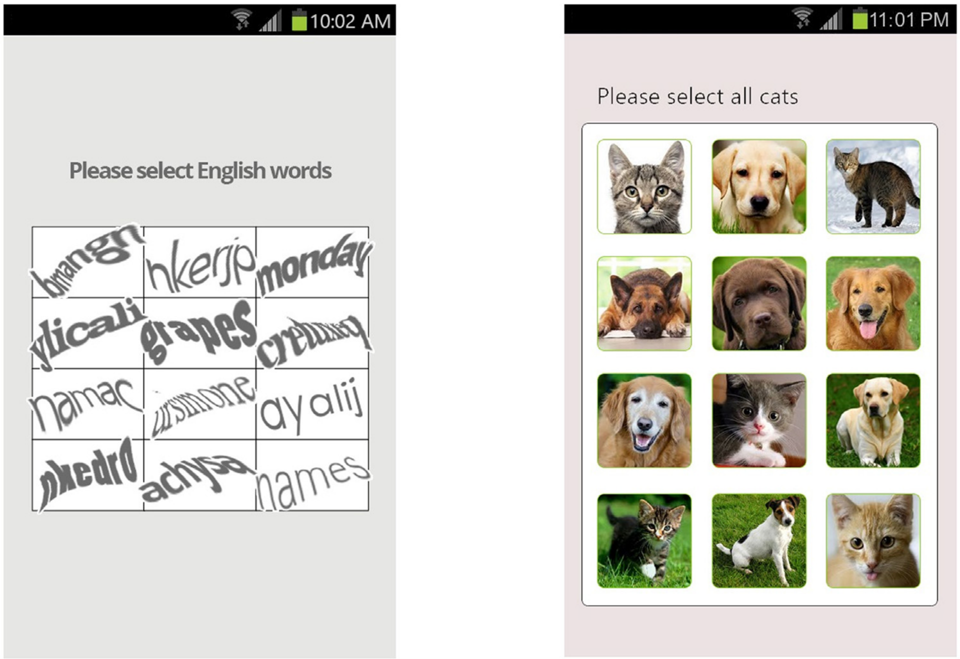


Fig. 2. BrightPass and reCAPTCHA scheme.

circle looks bright and a misleading lie digit whenever it looks dark. The main advantage of this scheme is that the user does not need to memorize an additional secret value nor to solve a complex cognitive task. In addition, their user study shows that it grants short authentication time (i.e., 6.73 s) and low error rates (i.e., 1.81%).

CAPTCHA-based authentication methods. Proposals in this category require the user to solve a challenge–response test, such as CAPTCHA, to filter out malware before they can try to enter passwords or PINs. 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 [6] (Fig. 2, right). Their approach consists in using scanned words from old books that Optical Character Recognition (OCR) programs fail to recognize. Whereas, the challenge is a combination of an unknown word with a control word whose content is known. If the user correctly recognizes the control word, it is assumed that his judgment about the other word is also valid. However, the hardest category of this scheme has been recently broken by Goodfellow et al. [7] with an accuracy of 99.8%. Instead of traditional approaches that distinguish among localization, segmentation, and recognition steps, they introduced a unified approach that integrates these three steps through a deep convolutional neural network that operates directly on the image pixels. Then, they apply it to transcribing synthetic distorted text from reCAPTCHA. In addition to this security issue, a recent research [8] pointed out that existing CAPTCHA schemes, including reCAPTCHA, are not suitable for mobile devices. This is due to the lack of usability that frustrates the user and lead to errors. In [9], authors suggested alternative



Please select English words

Please select all cats

Fig. 3. Clickable CAPTCHA and Asirra scheme.

input mechanisms aimed at improving the usability of ReCaptcha on smartphone. However, the usability comparison results show that participants prefer the existing ReCaptcha scheme, that uses the virtual keyboard as the primary input.

Chow et al. [10] introduced the idea of *Clickable CAPTCHA* (Fig. 3, left) in order to make CAPTCHAs suitable for mobile devices. Their approach consists in combining multiple textual CAPTCHAs into a grid of Clickable CAPTCHAs (e.g. a 3-by-4 grid). The user is required to click on the grid elements that match the challenge requirement. For example, the challenge can be the identification of English words among non-English words in the grid. Thus, it requires the selection of some elements in the grid, instead of traditional textual CAPTCHAs that requires inputting a string of characters using the mobile keyboard, which results to be challenging. Despite showing some advantages, this scheme has not been widely deployed.

Pequegnot et al. [11] proposed an authentication mechanism based on graphical Turing test to increase the confidence in mobile transactions. Their mechanism consists of typing a secure code of three-digit displayed in a CAPTCHA, in addition to the four-PIN digits. The secure code is randomly generated by a Secure Element for each authentication session. In this way, the system prevents mobile transactions submission by malware. However, this scheme is similar to existing commercial text-based CAPTCHAs, which add noise and distortion to CAPTCHAs to make them harder to break. Nevertheless, all of them have been defeated with high percentages of accuracy through object-recognition techniques, e.g., [7,12–15]. In addition, using too much noise and distortions makes them harder for humans to decipher as well, especially on tiny screens.

An alternative to text-based CAPTCHA forms are image-based CAPTCHAs. A typical CAPTCHA of this kind is Drawing CAPTCHA (Fig. 4, left). It has been proposed by Shirali-Shahreza et al. [16] and it has been specifically designed for mobile devices. In this scheme, numerous dots are displayed on a screen with noisy background. To pass the CAPTCHA challenge, the user has to connect the diamond-shaped dots to each other. Albeit easy and straightforward to use, this approach is not secure. In fact, Lin et al. [17] noticed that the sizes of background dots and the *noise square dots* are smaller than the diamond-shaped dots which can be filtered out by several erosion operations. Based on their observation, they proposed an image-processing technique that breaks Drawing CAPTCHA with an accuracy of 75%. Another image-based CAPTCHA is Asirra [18], which display 12 images of cats and dogs (Fig. 3, right) and asks the user to select all cat images among them. Their user study shows that solving the Asirra challenge takes less than 30 s for the 96.6% of humans, thereby making it convenient compared with text-based CAPTCHAs. However, Golle [19] showed that this scheme is vulnerable to machine learning attacks. Another proposal is NuCaptcha [20]. NuCaptcha uses the animation of a series of characters, i.e., codewords, that the user has to identify. The idea behind NuCaptcha is that the eye and the visual cortex leverage the motion of the characters to clearly differentiate them both among each others and from the background. In this way, the challenge proposed to the user can be solved easily. In [9], Reynaga et al. shows that the usability of NuCaptcha on smartphones is promising; however, this scheme has been broken [21,22].

Another recently introduced form of CAPTCHA is *game-based CAPTCHA*. Liao et al. [23] proposed accCAPTCHA (Fig. 4, right), a 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, ...). In accCAPTCHA, the motion operations are carried out by moving the device in the case of accelerometer-enabled device or by touching the screen in the other cases. Authors claimed that is difficult for a malware to understand the meaning of these games and thus, to provide the correct response. However, their user study shows that most gaming sessions take a long time to pass the challenge (e.g., the Stack game: 47.3 s., the Rolling ball game: 25.2 s., and the Racing game: 55 s).

3. CAPPCHA in the current hardware context

We now discuss the hardware requirements for a secure implementation of CAPPCHA and we prove its feasibility with current technologies. In order to increase the security of PIN codes – which are used to access sensitive mobile services – against malware attacks, there exist industry-led initiatives pushing the adoption of a Trusted Execution Environment (TEE), a Secure Element (SE) or both.

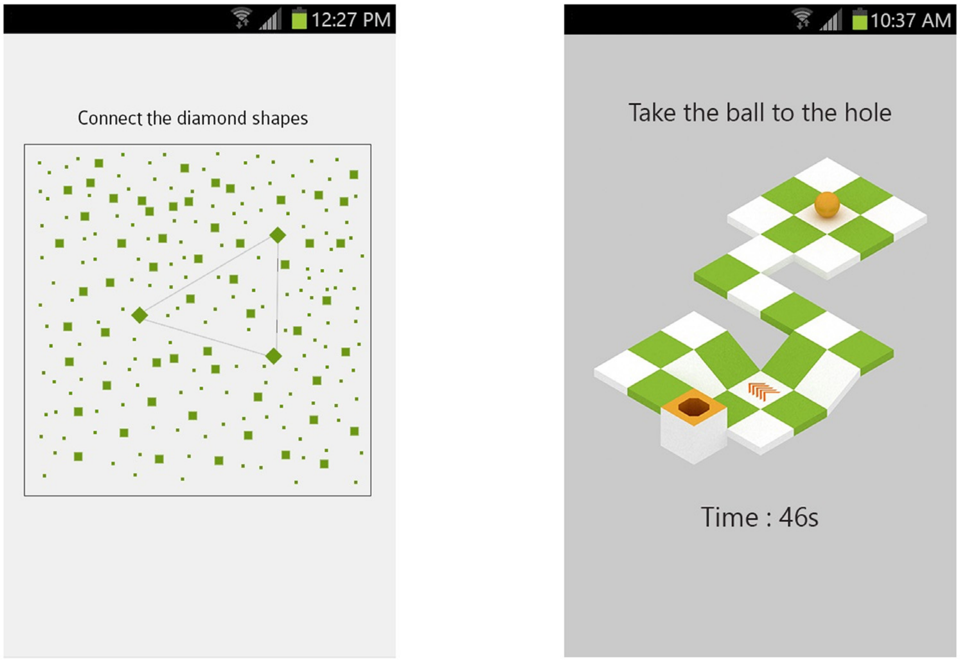


Fig. 4. Drawing CAPTCHA and accCAPTCHA scheme.

Secure Element (SE). The SE is a combination of hardware, software, interfaces and protocols embedded in a mobile handset [24]. This component provides a secure platform which enables an isolated execution environment and tamper proof data storage for applications. This ensures a high level of security and reliable identity management to each application, network and user [25].

SEs come in a variety of form factors. The most common form are *embedded Secure Elements* (eSEs), *Universal Integrated Circuit Cards* (UICCs), and *Secure Memory Cards* (Secure Micro SD) [26]. An eSE is a smart card embedded into the device mainboard. A UICC is an advanced SIM card which can have multiple applications hosted in it (e.g. USIM, CSIM and ISIM applications) and can communicate using the Internet Protocol (IP). In addition, it can support multiple PIN codes. A Secure Micro SD holds an embedded chip which can be used as a SE, along with a Flash memory.

According to Infineon, the adoption of SEs in smartphones is expected to increase from 427 million units in 2015 to 1.6 billion in 2020 [27]. This development is mainly driven by the growth of NFC-enabled applications (e.g., mobile payment, transport and ticketing applications) that strongly rely on a secure element to achieve the expected level of security. Some examples of smartphones that include a secure element are Galaxy S4/S6/S7, Iphone 6/7, Iphone plus 6/7, Iphone SE, Lenovo vibe P1 and Lenovo X3, just to cite a few.

Trusted Execution Environment (TEE). A TEE is a secure, tamper-proof area inside the smartphone CPU. This execution environment runs alongside but isolated from the device main operating system (e.g., Android). It guarantees that sensitive data are stored, processed and protected inside a fully trusted environment. Its ability to offer safe execution of authorized software, known as *trusted applications*, enables the TEE to enforce protection, confidentiality, integrity and access rights on the data belonging to such trusted applications [28]. Hence, the TEE isolates trusted applications and keeps them away from any malware which might be downloaded and executed in the main operating system. Another benefit of TEE is the ability to offer a trusted user interface (UI) which ensures that the information displayed on the screen and entered by the user are secure [29]. An example of a commercial deployment of TEE on mobile devices is the ARM TrustZone technology [30]. In 2012, ARM announced that it would include ARM TrustZone Technology in every processor design they license to manufacturers [31]. As a result, nowadays many smartphones are equipped with a processor with ARM TrustZone Technology (e.g., Galaxy S3/S4/S5/S6, Nexus 4/5, Galaxy Nexus, Galaxy Note 4, Moto G, ...).

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. In contrast, the SE supports physical robustness and high tamper resistance against side channel attacks [32]. The SE can work in combination with the TEE. For example, a SE can be used to store proximity payment applications – that require the highest level of security – in combination with a TEE that filters access to the same applications.

3.1. Assumptions and threat model

Phone architecture. In this paper, we discuss two ways for implementing CAPPCHA, namely `TEE-based CAPPCHA` and `SE-based CAPPCHA`.

In TEE-based CAPPCHA, we assume that the user has a TEE-enabled smartphone which is available in the most today's smartphones. On the other hand, in SE-based CAPPCHA, we assume that the smartphone uses a SE with an embedded accelerometer sensor. SEs equipped with embedded sensors, such as sweep sensors, are already commercially available thanks to the collaboration between Fingerprint Cards (FPC) and Infineon Companies [33]. Furthermore, Michel Willemin [34] recently released a prototype of a smart card (i.e., a SIM card or a memory card) equipped with a motion sensor that acts as an accelerometer. Therefore, it is reasonable to assume that similar products will be commercialized in the very near future. Moreover, this will allow CAPPCHA to be implemented in all mobile devices that are not TEE-enabled.

Infection. To prove the resilience of our scheme, we assume that the user has naively installed a malware from app stores or websites. We also assume that the malware exploits a vulnerability in Android OS and is able to gain root access on the device (see

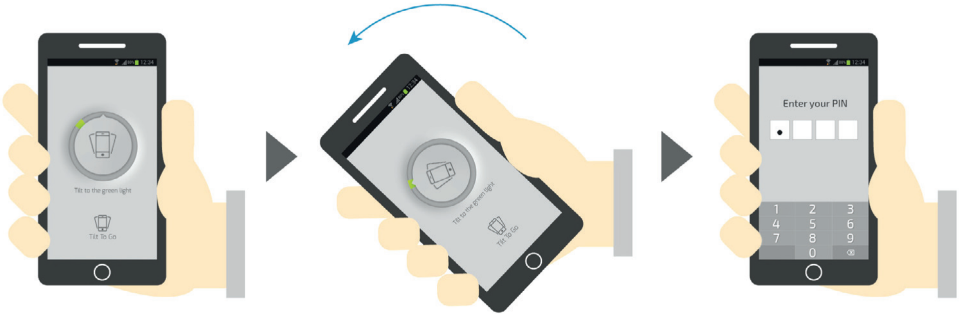


Fig. 5. The CAPPCHA + PIN authentication method.

e.g. [35,36]). As stated above, we also assume that the sensitive application is running in a TEE or a SE, and is protected with a common PIN code. Despite the strong isolation provided by TEE and SE to protect sensitive applications, the malicious code running in Android OS can steal the user's PIN code (see e.g., [37–40]) and replay this PIN in the next authentication session to execute unwanted transactions without the user's consent [11].

3.2. CAPPCHA test concept

Despite the variety of user authentication methods proposed in literature (see Section 2), most of the current applications running on mobile devices still use PIN codes to authenticate the user that aims to access sensitive services/data or to perform security-critical transactions. This is due to their simplicity, ease of remember and input. In order to enhance the security of PIN authentication against mobile malware, we add a simple CAPPCHA test to the PIN entry without affecting its usability. CAPPCHA requires the user only to tilt the device to a specific angle whose value in degrees is displayed in the screen and to hold it still in this position for one second to have access to the PIN pad (see Fig. 5). This approach is expected to be less burdening for the user in comparison to solve complex cognitive tasks or memorizing additional secret values with complicated input processes, as it happen in current literature proposals. The rationale behind CAPPCHA is to use something that the user can do easily while the malware cannot. We analyze the benefit of this approach by discussing a TEE-based and a SE-based implementation of CAPPCHA.

TEE-based CAPPCHA with a random challenge angle (`TEE-random`). This method uses a TEE, such as ARM TrustZone. The random challenge angle is generated by the ARM TrustZone for each authentication session in order to prevent the malware from simulating the current motion value, measured by the phone's accelerometer, in the next transaction.

SE-based CAPPCHA with a fixed challenge angle (`SE-fixed`). This method uses a secure element with an embedded accelerometer sensor. In this case, there is no need to generate a random challenge angle for each authentication session because the sensitive application gets the motion value directly from the trusted hardware (i.e. accelerometer) that cannot be fooled by a rootkit infecting the device. Thus, in this case a user would have the possibility to choose the challenge angle that he is most comfortable with. It is important to notice that, even if a malware could capture the current tilt value by using the non-protected accelerometer in the device, it cannot feed any false value to the software that interacts only with the accelerometer running in the SE.

3.3. Technical feasibility assessment

Both implementations enhance the security of PIN authentication against mobile malware by asking the user to tilt the device to a fix or random angle before entering his PIN code, e.g., when the user wants to either perform sensitive transactions or execute any other mobile service that require authentication. Albeit we are primarily focused on using CAPPCHA test to enhance authentication on mobile devices, it is worth noticing that the same test on a sensors endowed platform such as a smartphone, can be used in all the situations in which CAPTCHA are currently used.

Both TEE-random and SE-fixed implementations are viable in the current mobile ecosystem. In fact, most of the smartphones nowadays embeds either a TEE or SEs. Thus, assuming the adoption of a TEE-enabled smartphone or a smartphone that is equipped with a SE containing an embedded accelerometer sensor is not a far-fetched. In the case of smartphones without TEE, a removable SE such as SIM or Secure Memory Card with embedded accelerometer sensor can be used. However, as this hardware currently is not yet commonly available in the market, an alternative can be adopting the trusted sensor proposed recently by Haider et al. [41]. In order to ensure integrity, authenticity and non-repudiation guarantees on the sensed data, they uses physically unclonable functions (PUFs) which is already applied commercially (e.g., Intrinsic ID [42], Verayo [43,44], Microsemi Smartfusion2 FPGAs [45], and NXP SmartMX2 [46]). A PUF is an integrated circuit that serves as a hardware security primitive thanks to its complexity and the high level of unpredictability of the output [47]. In [41], authors argue that their PUF-based design is lightweight, has improved physical security, and requires no modifications in the sensor hardware as it happens for TPM-based solutions (e.g., [48,49]). However, evaluating the security granted by CAPPCHA implementation with PUF is out of the scope of this paper.

4. Security analysis

Since CAPPCHA is basically a way to enhance the PIN authentication on mobile devices, in this section we discuss the treat model for CAPPCHA with PIN, according to different classes of threats related to the authentication on mobile devices.

4.1. Brute Force attacks

One of the main security issues of PIN authentication are *brute force attacks* where an attacker tries all possible character combinations until the password is found. The proposed scheme involves presenting a problem challenge that humans can solve easily but would be impossible for a computer program. In the TEE-random case, the randomization of the challenge angle for each authentication session prevents the automated process of iterating through the entire circle space.

In the SE-fixed case, the use of the secure element with an embedded sensor prevents the automated process from any authentication attempt, thereby avoiding successful brute force attacks. This is due to the fact that malicious codes cannot feed false data into the authentication process as it relies on the sensor embedded into the secure element. Thus, the proposed scheme is resilient against brute force attack.

4.2. Side channel attacks

Side channel attacks aim at stealing user's keystrokes, even when strong isolation is applied, by trying to get or infer the user input from other components. More specifically, these attacks leverages resources that are shared between the mobile OS and the trusted OS, such as the accelerometer [37], the camera and the microphone [38], the Gyroscope [39,40], etc.

In the TEE-random case, the usage of TEE prevents side channel attacks through the randomization of the challenge angle for each authentication session. In this way, the current motion value, measured by the accelerometer, cannot be used in the next transaction. Thus, stealing the response in the current challenge is useless since the malicious code is unable to solve the challenge.

In the SE-fixed case, the SE provides protection against side channel attack even without randomizing the challenge angle. This is due to the fact that CAPPCHA measures the motion value from trusted hardware and this value cannot be changed by the malicious code even if it has root access to the device: instead, it can be changed only when the user physically moves the device. Hence, the proposed scheme grants security against side channel attacks.

4.3. Recording attacks

The problem of mobile malware has been aggravated by the introduction of recording attacks. In addition to inferring the user's touch coordinates, current recording attacks are able to record the entire authentication screen [50], thereby making hard to protect against this class of attacks. However, in the TEE-random, leveraging a TEE-enabled device when users interact with the Trusted OS, the Android OS cannot access the screen [38] and, thus, a secure input path to the user is provided and the malicious code cannot take screenshots to reveal the challenge angle. Therefore, the TEE-random is resilient against recording attacks. In the SE-fixed case, when using the secure element instead of TEE, a malware having root access to the device could take a screenshot and reveal the challenge angle since the SE does not provide a secure access to the screen. However, this would be useless because, even if it knows the value of the challenge, the malware cannot bypass the measurements taken by the accelerometer inside the secure element, thus it cannot fool the authentication process into believing that the phone has been tilted to the correct angle.

5. Experimental results

The time granularity at which a security check has to be done, is not an intrinsic of the check itself: it depends on the level of security you want to enforce. As an example, in a computer system you only provide your credentials at the beginning of the session, but if you engage in a critical activity (e.g., banking) your credentials are tested again (maybe even with a stronger mechanism). Obviously, a very awkward and intrusive mechanism will not call for frequent use possibly jeopardizing the whole system security, while a very simple and pleasant scheme lend itself to frequent usage, with a better chance to secure the system from intrusions. For these reasons, designing a new authentication mechanism requires taking into consideration both computer security and usability/user friendliness.

Therefore, in this section we describe an experiment that allows (1) to evaluate the usability of CAPPCHA and (2) to compare it to other existing mechanisms. Before we describe the experiment in details, it is important to clarify some important facts. First, CAPPCHA, exactly as CAPTCHA, is not, in itself, an authentication mechanism; on the contrary, it is a bot-culling mechanism, i.e., a mechanism targeted at pointing out automated programs (bots) among human users. Therefore, we compared CAPPCHA directly with other bot-culling schemes, but in order to make a fair comparison with full authentication schemes that declare to be bot-proof by themselves, we have measured a two-stages activity composed by solving CAPPCHA first and then providing a PIN; this two-stage activity constitutes an actual bot-proof authentication mechanism and is thus comparable with other authentication schemes. Second, average usage time and error rates of existing authentication schemes are extracted from the relevant literature; we have checked the references publications and the sources get reliable values; however, average values may not have been calculated exactly the same way for each scheme (e.g., inclusion or exclusion of outliers). Nonetheless, even with these differences, the experiments described in the relevant literature provide a basis for a meaningful comparison of the CAPPCHA approach with other existing solutions. Third, in order to evaluate the usability of CAPPCHA, the underlying hardware requirements are not relevant while the important part is only what changes in the user perception. Hence, the fact that CAPPCHA is TEE or SE based is meaningless from the usability point of view, while it is important to distinguish between CAPPCHA that proposes a random angle from the one that proposes an a-priori known angle. Nonetheless, to maintain a uniform naming, we will continue to identify the different schemes as TEE-random and SE-fixed (see Section 3.2 for details).

Participants. We recruited 200 volunteers among students at the University of Genoa. The age of the participants to the study ranges from 19 to 23, among the participants 47 are females, while 153 are males. Some participants preferred to use their own smartphone, while many others used the smartphones that we provided for the test. The reason behind the choice to use a smartphone different from the user's own is the desire to avoid installing additional (in the users perception potentially disruptive) software on a device considered a precious tool both for business and for daily life in general.

Procedure. Before starting the test, the CAPPCHA mechanism was explained in details to all users. Then, they were asked to test the application until they felt familiar with the concept. Due to the simplicity of the challenge, most participants did not have to perform more than two tests to get acquainted with the mechanism. Afterwards, participants were asked to solve the challenge ten times for both CAPPCHA implementation configurations (i.e., TEE-random and SE-fixed). Thus, our results are based on 2000 CAPPCHA solution + PIN entry for each configuration. The challenge time (i.e. CAPPCHA test time) has been measured from the start of the application to the successful solution of the challenge, while the authentication time has been taken in two steps: from the start to the solution of the challenge and the from the solution of the challenge to the completion of PIN entry. The logged data stored on the smartphones were used to calculate the average challenge time per user. Relevant statistical parameters, including confidence intervals, of the measured timings are presented. Once participants have completed the test, a satisfaction survey was provided to each participant in order to collect the participants' evaluation for the CAPPCHA test. Then, the users were invited to answer to the questionnaire and note their comments as well as any problem they encountered. The whole set of ratings and comments constitutes our evaluation of the global usability of CAPPCHA.

The user study has been carried out in a controlled environment in order to observe the participants spontaneous behavior in

Table 1
Values and statistical parameters for SE-fixed CAPPCHA.

	CAPPCHA	PIN	CAPPCHA + PIN
Average	1.80	1.00	2.79
Min	1.57	0.53	2.16
Max	2.09	1.35	3.30
Var	0.01	0.03	0.04
StDev	0.10	0.17	0.20
Conf Int alpha 0.05	0.01	0.02	0.03
Conf Int alpha 0.01	0.02	0.03	0.04

Table 2
Values and statistical parameters for TEE-random CAPPCHA.

	CAPPCHA	PIN	CAPPCHA + PIN
Average	2.86	1.00	3.86
Min	2.29	0.53	3.16
Max	3.50	1.35	4.71
Var	0.06	0.03	0.10
StDev	0.25	0.17	0.32
Conf Int alpha 0.05	0.03	0.02	0.04
Conf Int alpha 0.01	0.04	0.03	0.06

holding the smartphone while solving the challenge. Furthermore, the controlled environment allowed us to collect the paper-based questionnaires. The whole anonymous collection of the votes as well as the average CAPPCHA, PIN and CAPPCHA + PIN solution time is available at http://geas.dei.unipd.it/CAPPCHASurvey2016.pdf.

Equipment and software. Each CAPPCHA implementation has its own hardware requirements, i.e., a TEE-enabled smartphone (for TEE-random) or a SE with embedded accelerometer (for SE-fixed). However, the actual presence of the hardware that is required to make the solution secure is meaningless from the usability point of view; thus, we implemented an app able to work on any smartphone with no dependencies from the actual presence of secure hardware. When implementing the CAPPCHA test, it is important to take into consideration that humans cannot hold their hands perfectly still; this is why we used a small green line representing an angle range (see Fig. 5) as a challenge.

5.1. Performance comparison

Tables 1 and 2 show the experimental results together with their statistical parameters, while Fig. 6(a) and (b) show the box plots of the average timings. As we stated before, the numbers provided in these table and figure summarize the result of an experimental campaign comprising 4000 CAPPCHA and PIN solving instances.

Table 3 allows comparing the average authentication time and error rate of the proposed authentication scheme (i.e., CAPPCHA + PIN) to other authentication methods that try to enhance the security of the PIN code by adding an additional secret value or an overhead. Table 4, on the contrary, allows comparing the average challenge time and the error rate of CAPPCHA to other schemes targeted at preventing automated attacks from reaching a subsequent stage by means of challenge/response tests.

5.1.1. Authentication schemes

In this subsection we analyze the performance of the authentication schemes on the basis of the time that users need to successfully complete the authentication and the rate of failed authentications. FakePin and PassWindow both have an average solution time that is three to four times longer than the coupling of CAPPCHA and a PIN. BrightPass, while much faster that the other two schemes, still requires a solution time that is, in the average, twice the one required by SE-fixed CAPPCHA + PIN and about 170% than the one required by TEE-random CAPPCHA + PIN. Furthermore, even if we have no information about the confidence interval of the average values for the other schemes, we can observe that the values are all and by far out of the confidence interval of CAPPCHA + PIN measurements.

Table 3
User test results from different systems. Results are taken from referenced papers.

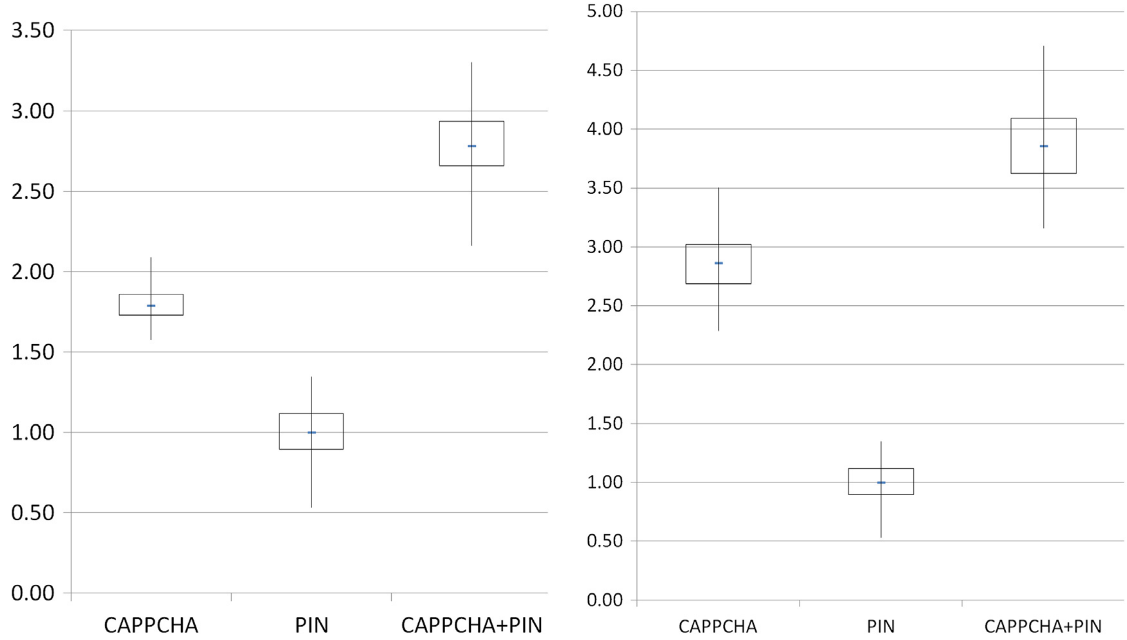
Authentication method	Authentication time (s)	Error rate (%)
FakePin [3]	14.13	4.7
PassWindow [4]	17.86	4
BrightPass [5]	6.73	1.81
TEE-random CAPPCHA + PIN	3.86	0
SE-fixed CAPPCHA + PIN	2.79	0

In terms of error rates, CAPPCHA + PIN is again stronger than the other three schemes and the users in our experiment never failed an authentication attempt, while all the other showed a non-zero error rate. The absolute absence of errors in the CAPPCHA + PIN scheme is due to the combination of two factors: first, our implementation of CAPPCHA cannot fail, it may only take a longer time to complete; second, entering a PIN has become a very natural operation for the users.

5.1.2. Bot-culling schemes

We now analyze the performance of the different bot-culling schemes. NuCaptcha [20], despite its security issues, requires a short time to solve the challenge and guarantees a low error rate, w.r.t. the other forms of CAPTCHAs. In the case of Clickable Captcha [10], we notice that familiarity with the English language is a discriminating parameter when evaluating both the time required to complete the challenge and the error rate; this is rather obvious when we consider the linguistic nature of the challenge. Similarly to Clickable Captcha, ReCaptcha [6] suffers from language dependencies, as well as usability issues related to the addition of distorted text and the use of a small keyboard for the input. Asirra [18] solves some usability issues related to Text-based CAPTCHAs; however, it is ill suited to mobile devices as visualization problems when displaying images on small screens generate longer solution times and increase the error rate. Similar to CAPPCHA, AccCaptcha [23] uses the physical interaction with the device to solve the challenge. However, it takes a long time and has a high error rate compared to CAPPCHA; this is due to the fact that CAPPCHA does not require any cognitive tasks and one simple gesture is enough to solve the challenge. Furthermore, it is important to notice that the CAPPCHA mechanism we implemented cannot lead to authentication errors (false negatives); it may only delay the successful completion of the challenge. The experimentation with both implementations (i.e., TEE-random and SE-fixed) allowed us to notice that setting the authentication angle to the user's preferred value results in shorter challenge times. However, we also observed that allowing users to select a preferred angle resulted always in a request for a small rotation. Therefore, it is possible that the shorter time is not due to a motor memory effect but just to a gesture that is simpler to perform.

As shown in Table 4, CAPPCHA has by far the fastest solution time and the lowest error rates among all the evaluated schemes. We argue that the main reason at the basis this results must be found in the simplicity and in the nature of the CAPPCHA challenge. In fact, the user has only to perform a simple motor task, namely tilting the device to a specific degree, instead of solving complex cognitive tasks. In addition to that, the CAPPCHA test is designed to overcome the usability issues of Text-based CAPTCHA or Image-based CAPTCHA. For example, CAPPCHA is language independent and does not require text-entry in a small button virtual keyboard or zooming images to solve the challenge. Furthermore, it is based



(a) SE-fixed CAPPCHA.

(b) TEE-based CAPPCHA.

Fig. 6. Box plot for the average times of CAPPCHA solution, PIN entry and full CAPPCHA + PIN authentication.

Table 4
User test results taken from the referenced papers.

Challenge/response tests	Test features	Avg. challenge time (s)	Error rate (%)
Clickable Captcha [10]	Subjects familiar with English	11.1	0
	Subjects not familiar with English	18.2	10–20
Acc Captcha [23]	Stack Game	47.3	33
	Rolling Ball Game	25.2	22
	Racing Game	55	4
		8.5	2
NuCaptcha [9]			
ReCaptcha [9]		25.5	9
Assira [9]		29.2	19
CAPPCHA test	TEE-random	2.9	0
	SE-fixed	1.8	0

on a simple gesture rather than cognitive tasks that humans perceive as hard and annoying to solve. For all of these reasons and because of the results of the security analysis we performed in Section 4, we claim that the proposed schema improves the global level of security while remaining fast and easy to use.

5.2. Survey of participants perception of our scheme

The questions used in this user study has been inspired from a recent survey performed by Reynaga et al. [9] on usability of CAPTCHAs on smartphones. The questions were provided to participants as follows:

1. Acc: It was easy to accurately tilt the device to the challenge degree.
2. Und: The challenge was easy to understand.
3. Mem: If I did not use this mechanism for a few months, I would still remember how to solve the challenge
4. Pleas: This CAPPCHA mechanism was pleasant to use.
5. Solv: I found it easy to tilt the device to the challenge degree.
6. Suit: I found CAPPCHA mechanism well suited for the smartphone.
7. Pref: On a smartphone, I would prefer using the CAPPCHA mechanism compared to other CAPTCHAs.
8. Input: The input mechanism is better than traditional input mechanisms (e.g., virtual keyboard).

Participants were asked to rate each statement from 1 (Strongly Disagree) to 10 (Strongly Agree). Table 5 shows the average value of the ratings of participants and the statistical parameters of our collection of data. In Fig. 7 it is possible to see the box plot of the scores we collected.

We notice that all statements were scored positively from all participants. The participants generally found the CAPPCHA mechanism very easy to understand and to use. All of them stated that they would be able to remember the concept behind the challenge easily even months later because the challenge is simple and does not require complex cognitive tasks. In addition, they found it well suited to the smartphone form factor, as it does not require text input using small virtual keyboard or to zoom the screen to recognize images or distorted text. All these features let participants express high ratings in preferring the CAPPCHA mechanism w.r.t. other CAPTCHA schemes.

In order to know which configuration the users prefer and to further enhance CAPPCHA, we added the following questions:

1. Preferred angle: which angle do you prefer to tilt the device to it?
2. Randomization: do you prefer the CAPPCHA test with a fixed preferred angle as challenge, a random angle or there is no difference?

Most participants stated that they would prefer small fixed angles for the challenge (e.g. ±30, ±45, ±60). However, the main

Table 5
Average users' score and statistical parameters.

	Acc	Und	Mem	Plea	Solv	Suit	Pref	Input
AVERAGE	9.11	9.56	9.72	9.44	8.97	9.65	9.56	9.16
MIN	6.00	6.00	7.00	6.00	6.00	6.00	6.00	6.00
MAX	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
VAR.P	1.28	0.76	0.39	0.82	1.59	0.56	0.67	1.27
STDEV.P	1.13	0.87	0.63	0.90	1.26	0.75	0.82	1.13
Conf Int alpha 0.05	0.16	0.12	0.09	0.13	0.17	0.10	0.11	0.16
Conf Int alpha 0.01	0.21	0.16	0.11	0.16	0.23	0.14	0.15	0.21

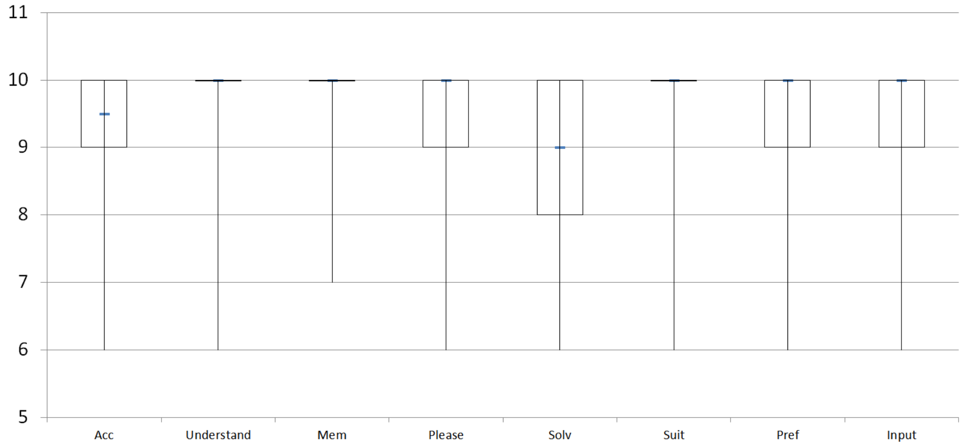


Fig. 7. Box plot for average users' score.

reason to choose the fixed angle challenge was the difficulty encountered when tilting the smartphone to a challenge degree above 90° or below −90°. Thus, a focused comparison between the fixed angle case and the random angle case limited to small angles should be performed to gain a more in depth understanding of this issue.

5.3. User comments

We invited participants to provide us comments regarding their experience with CAPPCHA 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 CAPPCHA test. Selected comment samples are shown in Table 6.

Many participant comments were related to the difficulty to tilt the smartphone to angles above 90° or less than −90° and they suggested that the generated challenge angle should be within this range or even smaller. Two participants suggested using arrows to indicate the rotation direction to help them go directly in the right direction instead of trying both directions, as noticed on their spontaneous behavior when solving the challenge. One participant in the survey provided an interesting insight on the capability of CAPPCHA to overcome sight problems, stating that it could be very suitable for sight impaired people. To evaluate his comment we asked the sight impaired participants to take an additional round and solve the challenge without wearing their glasses. All of them were able to solve the challenge successfully, while, on the contrary, they stated that they need to wear their glasses to input text using virtual keyboard. Hence, they recommend CAPPCHA as a promising alternative to existing CAPTCHA schemes even for aging people.

5.4. Observations on participants' behavior

Since the CAPPCHA mechanism is based on tilting the smartphone, the observation of how participants spontaneously handle their smartphones to solve the challenge is important as it directly impacts the ease with which some challenges may be solved. According to our observations, participants basically uses three different positions to handle their smartphones as depicted in Fig. 8.

Participants who used position (a) to handle their smartphones encountered difficulties in tilting it to directions over 30° to left direction while holding the smartphone in left hand, and vice-versa. This is due to the fact that their wrist is already bent in order to hold the phone vertically. However, tilting the phone to the right while holding it in the left hand, regardless of the challenge angle, was comfortable for all participants in position (a).

Participants who used the second position (b) felt generally comfortable with turning the device in both directions until 90°. We noticed that this was the most comfortable way to use CAPPCHA. We also noticed that participants that used both position (a) and (b), had to twist their entire arm when the challenge angle is more than +90° or less than −90° which was not comfortable and annoying to all participants.

Participants who used the third position (c) to handle their smartphones used both their hands to tilt the smartphone to any challenge angle comfortably, and without having to twist their arms. It is worth noticing that most of them reported that the concept is more game-like or similar to a steering wheel.

5.5. A focus on age related differences

As our sample was very homogeneous in terms of users' age, we cannot use it to formulate any significant observation about the differences in usability that users of different age categories may find in our scheme. To have a first level evaluation of this topic, we decided to perform an additional set of tests with a limited number of users with significant differences in age. The size of the sample is 35 divided as follows: 3 users are in age range 15–19, while 13 are in the 20–29, 10 in the 30–39 and 7 in the 40–58 ones.

As this experiment is focused on detecting possible differences in users of different ages we proposed only the longest and apparently most difficult challenge, i.e., the TEE-random one. The

Table 6
Some examples of user comments.

Sex	Comments
F	Very easy and effortless Unusable on a PC One cannot make a mistake with this mechanism
M	The less tilting, the more comfortable users would be Angles above 90° and less than −90° are non practical as well as annoying
M	Quick, generally easy to understand and use It's easy between −30° and 30°
M	It seems like a game to me There is no way to make a mistake
M	Less tilting would be better The easiest mechanism I used
F	Above 90° is definitely hard to solve Very intuitive Add arrows to indicate the rotation direction Unfortunately, it cannot be used to authenticate on a PC
M	Easy to use I will prefer this concept than other captchas
M	Fast and easy, it seems like steering wheel to me Very good idea for people who have sight problems and don't like to use virtual keyboard on smartphone

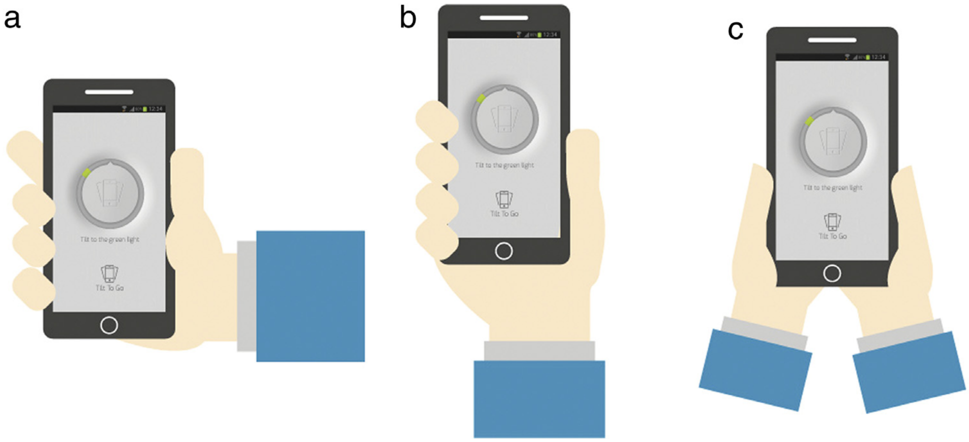


Fig. 8. Positions of phone handling.

procedure adopted for this second experiment was exactly the same we used for the previous one, thus we do not repeat here a detailed description (see paragraph *Procedure* in Section 5 for the specific details).

In Table 7 we show both the averaged measurements of the time needed to solve the CAPPCHA challenge and the averaged votes provided by the users, together with the relevant statistical parameters.

In Figs. 9 and 10, it is possible to observe the graphical representation of the votes separated by age range, where the error bars are dimensioned according to the confidence intervals with α equals to 0.01. Some of the voting categories (e.g., suitability) are extremely regular and all the values for the different age ranges are contained in the confidence intervals; furthermore, even the voting category that shows the largest difference between the maximum and the minimum value (i.e., solvability) has such a strong overlap in the confidence intervals of the different age ranges that we cannot claim any statistical evidence of an age-related difference. For this reason, this focus study does not allow us to identify any age-related significant difference in the perception that the users have of the CAPPCHA scheme.

The only statistically significant difference we can spot with the data collected in this focus is given by the CAPPCHA challenge solution timings. In Fig. 11 you can observe the values for the different age ranges. In fact, while the increase in solution time is not significant for all the users ages 20 and more, teenagers showed the capability to solve the challenge in a time that is 33% shorter that the one needed by the other users (i.e., one second less); furthermore, the confidence intervals of these mean values are very well separated and corroborate the significance of this difference.

These facts allow us to conclude that, even if teenagers are definitely quicker in solving the proposed CAPPCHA challenge, their perception of its usability does not show any statistically significant difference from the one of older users. Our experiments do not allow us to draw a conclusion about this phenomenon and we plan to investigate it in more depth in future work.

6. Conclusion and future work

Authentication is always the first step in any security architecture, in fact, until a system has managed to guarantee the identity of the user it cannot perform any kind of access control. Traditional authentication mechanisms, such as the static PIN/Password secret that is still the most commonly adopted one, greatly suffer from the growing sophistication of malicious software and the increasing computational power of computing platforms; thus, to mitigate their vulnerability researchers have proposed a preliminary form

Table 7
Votes and solution times separated by age range.

	Acc	Und	Mem	Plea	Solv	Suit	Pref	Input	Time
15–19									
AVERAGE	8.80	9.40	10.00	9.00	9.40	10.00	9.80	8.80	1.95
MIN	8.00	7.00	10.00	8.00	9.00	10.00	9.00	8.00	1.82
MAX	10.00	10.00	10.00	10.00	10.00	10.00	10.00	9.00	2.04
VAR.P	0.56	1.44	0.00	0.40	0.24	0.00	0.16	0.16	0.01
STDEV.P	0.75	1.20	0.00	0.63	0.49	0.00	0.40	0.40	0.08
ConfInt95	0.66	1.05	N.A.	0.55	0.43	N.A.	0.35	0.35	0.07
ConfInt99	0.86	1.38	N.A.	0.73	0.56	N.A.	0.46	0.46	0.09
20–29									
AVERAGE	8.92	9.54	9.92	9.31	8.69	9.77	9.31	9.00	3.07
MIN	7.00	8.00	9.00	7.00	7.00	8.00	7.00	9.00	1.98
MAX	10.00	10.00	10.00	10.00	10.00	10.00	10.00	9.00	4.11
VAR.P	1.15	0.56	0.07	0.83	0.67	0.33	0.98	0.00	0.47
STDEV.P	1.07	0.75	0.27	0.91	0.82	0.58	0.99	0.00	0.68
ConfInt95	0.58	0.41	0.14	0.49	0.45	0.31	0.54	N.A.	0.37
ConfInt99	0.77	0.53	0.19	0.65	0.59	0.41	0.71	N.A.	0.49
30–39									
AVERAGE	8.60	9.90	9.80	9.10	7.80	9.80	9.50	8.80	2.98
MIN	7.00	9.00	8.00	7.00	5.00	9.00	8.00	8.00	1.81
MAX	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	3.71
VAR.P	1.44	0.09	0.36	1.29	1.96	0.16	0.65	0.36	0.38
STDEV.P	1.20	0.30	0.60	1.14	1.40	0.40	0.81	0.60	0.61
ConfInt95	0.74	0.19	0.37	0.70	0.87	0.25	0.50	0.37	0.38
ConfInt99	0.98	0.24	0.49	0.93	1.14	0.33	0.66	0.49	0.50
40–60									
AVERAGE	8.00	9.57	9.71	9.14	8.29	9.29	8.71	8.71	3.29
MIN	6.00	8.00	9.00	8.00	7.00	8.00	7.00	8.00	2.30
MAX	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	4.51
VAR.P	1.43	0.53	0.20	0.41	1.06	0.78	1.35	0.49	0.61
STDEV.P	1.20	0.73	0.45	0.64	1.03	0.88	1.16	0.70	0.78
ConfInt95	0.89	0.54	0.33	0.47	0.76	0.65	0.86	0.52	0.58
ConfInt99	1.16	0.71	0.44	0.62	1.00	0.86	1.13	0.68	0.76

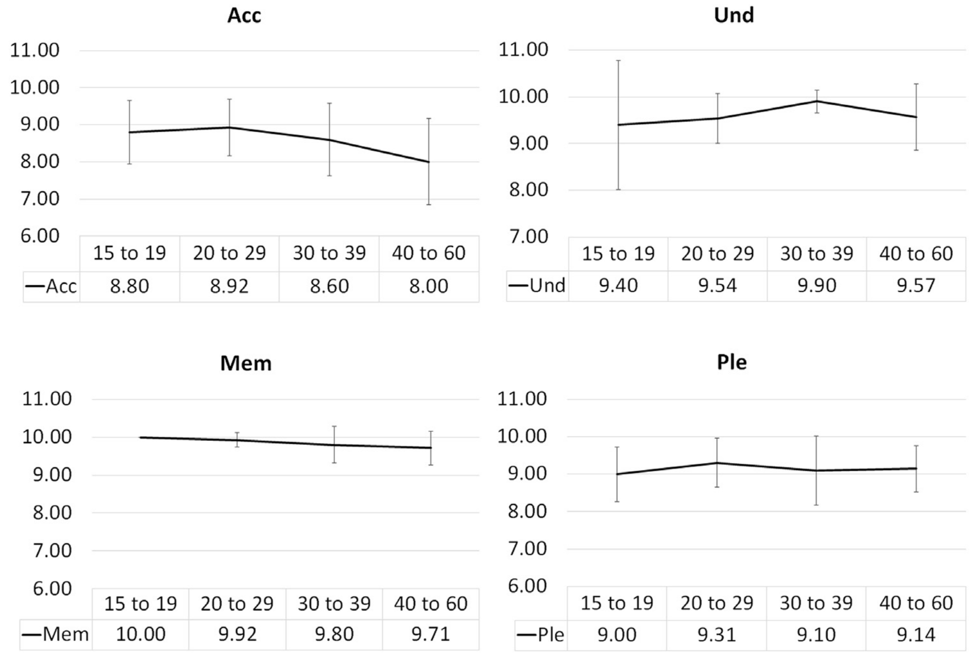


Fig. 9. Average votes for different age ranges (accuracy, understandability, memorability, pleasantness).

of a Turing test (CAPTCHA) to cull out the malware access attempts. CAPTCHA, however, have proved to be cumbersome and generally ill suited to the task, as the growing sophistication of the malicious software requires also increasing the complexity of the cognitive task to be solved. We proposed a new way of culling out malware access attempts (CAPPCHA) that, leveraging the physical nature of humans instead of their cognitive capabilities, does not introduces any cognitive overload for the users. To test the feasibility of our approach we developed a CAPPCHA scheme which asks users to tilt their devices to a specific angle to prove that they are humans before giving them access to the screen where they can enter their PINs.

The security analysis showed that the proposed scheme is resilient against brute force attacks, side channel attacks and spyware-based recording attacks. To evaluate the usability of our scheme we decided to perform an experiment similar to the one

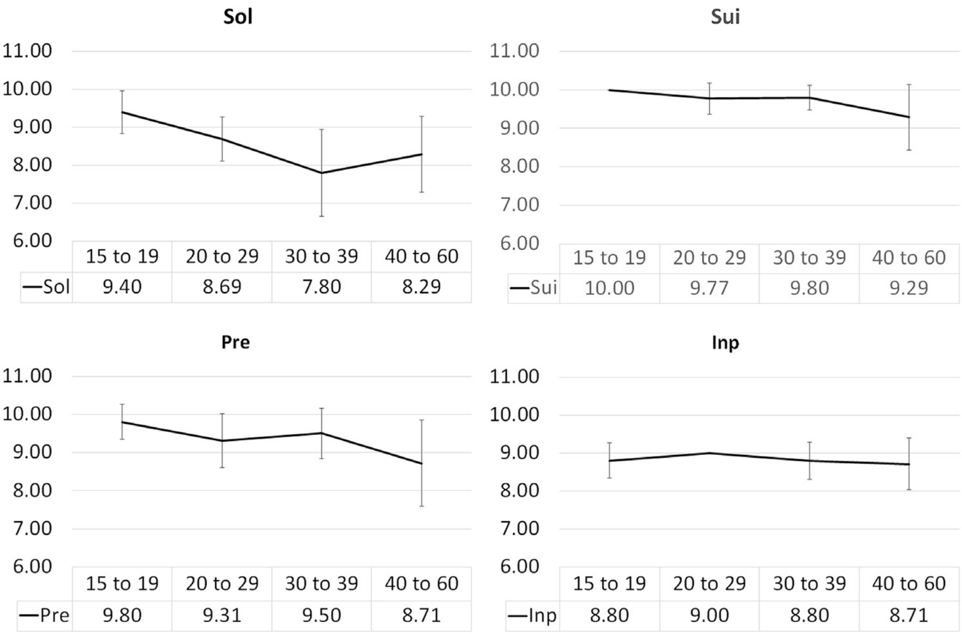


Fig. 10. Average votes for different age ranges (solvability, suitability, preference, input mechanism).

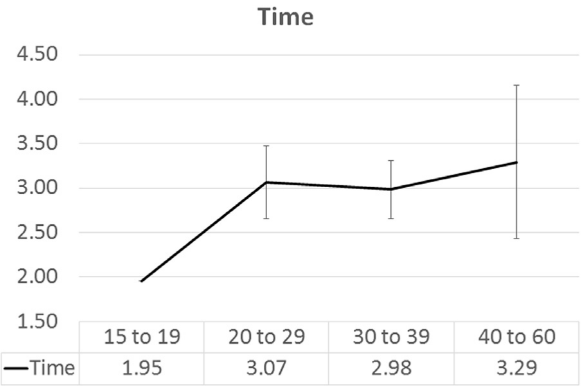


Fig. 11. Average CAPPCHA challenge solution timings for different age ranges.

used in usability surveys for CAPTCHA schemes available in the literature, namely we performed the experiment described in [9] with a significantly larger sample (more than 200 users). From a usability point-of-view, all the users involved in the experiment described our scheme as very intuitive and easy to use, thus CAPPCHA has the potential to replace CAPTCHA as the mechanism to cull out malware before traditional authentication mechanisms are used.

Despite the success of our experiment, the survey indicates that there are some new directions that need to be explored in future work. First, while the sample adopted was larger than the one used in other surveys, we should provide one with a larger statistical significance also when different age ranges are taken into account; in fact, the limited focus experiment we performed in addition to the large one to try identifying correlations between age and perception of our CAPPCHA scheme did not allow us to draw significant conclusions. Secondarily, the observation that a statically set, preferred angle of tilting resulted in faster authentication has to be compared with randomization of the angle in a smaller range to check if the effect we observed was due to some sort of motor memory or simply to easier gestures. Furthermore, while CAPPCHA currently targets mobile devices for the availability of sophisticated sensors on such a platform, several users suggested that porting it to traditional PCs would greatly enhance its usefulness; thus, we need to explore how the combination of a Trusted Execution Environment with physical sensors already present in the PC (e.g. the movement sensor in the mouse) could be leveraged to implement a CAPPCHA scheme. Finally, we need to study the usability of CAPPCHAs for users with impairments, as an example, serious sight problems and parkinsonisms can prevent users from successfully complete the CAPPCHA test we have implemented; thus, we need to study ways to test the physical nature of the user that can cope with physical impairments due to age or illness.

Acknowledgments

This work was partially sponsored by the ERASMUS MUNDUS Action 2 “AL IDRISI” project, Grant Agreement: 2013-2401/001-001-EMA2. The project is coordinated by Universidad de Granada (Spain) and participated by Università degli Studi di Padova (Italy).

References

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

[2] M. Raza, M. Iqbal, M. Sharif, W. Haider, A survey of password attacks and comparative analysis on methods for secure authentication, World Appl. Sci. J. 19 (4) (2012) 439–444.

[3] S. Kim, H. Yi, J.H. Yi, FakePIN: Dummy key based mobile user authentication scheme, in: Ubiquitous Information Technologies and Applications, in: Lecture Notes in Electrical Engineering, vol. 280, Springer, Berlin, 2014, pp. 157–164.

[4] H. Yi, Y. Piao, J.H. Yi, Touch logger resistant mobile authentication scheme using multimodal sensors, in: Advances in Computer Science and its Applications, in: Lecture Notes in Electrical Engineering, vol. 279, Springer, Berlin, 2014, pp. 19–26.

[5] M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, F. Palmieri, A. Castiglione, Using screen brightness to improve security in mobile social network access, IEEE Trans. Dependable Secure Comput. PP (99) (2016) 1–1. http://dx.doi.org/10.1109/TDSC.2016.2601603.

[6] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, M. Blum, reCAPTCHA: Human-based character recognition via web security measures, Science (2008).

[7] I.J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, V. Shet, Multi-digit number recognition from street view imagery using deep convolutional neural networks, in: ICLR, 2014.

[8] G. Reynaga, S. Chiasson, The usability of captchas on smartphones, in: Proceedings of SECRYPT, SciTePress, 2013, pp. 427–434.

[9] G. Reynaga, S. Chiasson, P.C. Van Oorschot, Exploring the usability of CAPTCHAS on smartphones: Comparisons and recommendations, in: USEC 2015.

[10] R. Chow, P. Gollé, M. Jakobsson, X. Wang, L. Wang, Making CAPTCHAs clickable, in: Ninth Workshop on Mobile Computing Systems and Applications, HotMobile 2008, Napa, CA, 2008 February 25–26.

[11] D. Pequegnot, L. Cart-Lamy, A. Thomas, T. Tigeon, J. Iguchi-Cartigny, J.-L. Lanet, A security mechanism to increase confidence in M-transactions, in: CRiSIS: 6th International Conference on Risks and Security of Internet and Systems, IEEE, Timisoara, Romania, 2011, pp. 9–16. http://dx.doi.org/10.1109/CRiSIS.2011.6061836.

[12] G. Moy, N. Jones, C. Harkless, R. Potter, Distortion estimation techniques in solving visual CAPTCHAs, IEEE CVPR, 2004.

[13] A.A. Chandavale, A.M. Sapkal, R.M. Jalnekar, Algorithm to break visual CAPTCHA, in: ICETET 2009 Proceedings of the 2009 Second International Conference on Emerging Trends in Engineering and Technology, pp. 258–262.

[14] J. Yan, A.S. El Ahmad, A low-cost attack on a microsoft CAPTCHA, School of Computing Science Technical Report, Newcastle University, England, 2008.

[15] M. Korakakis, E. Magkos, Ph. Mylonas, Automated CAPTCHA solving: An empirical comparison of selected techniques, in: 9th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), IEEE, Corfu, Greece, 2014, pp. 44–47.

[16] M. Shirali-Shahreza, S. Shirali-Shahreza, Drawing CAPTCHA, in: Proceeding of the 28 International Conference on Information Technology Interfaces, Cavtat, Croatia, 2006, pp. 475–480.

[17] R. Lin, S. Huang, G.B. Bell, Y. Lee, A new captcha interface design for mobile devices, in: ACSW 2011: Australasian User Interface Conference, Curtin, Australia, 2011.

[18] J. Elson, J.R. Douceur, J. Howell, J. Saul, Asirra: a CAPTCHA that exploits interest-aligned manual image categorization, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS, 2007.

[19] P. Golle, Machine learning attacks against the Asirra CAPTCHA, in: Proceedings of the 15th ACM Conference on Computer and Communications Security, 2008, pp. 535–542.

[20] NuCaptcha. Nucaptcha security features. last retrieved on February 27th 2016 from http://www.nucaptcha.com/security-features.

[21] E. Bursztein, How we broke the NuCaptcha video scheme and what we proposed to fix it, Mar. 2012. See http://elie.im/blog/security/how-we-broke-the-nucaptchavideo-scheme-and-what-we-proposeto-fix-it/.

[22] Yi Xu, G. Reynaga, S. Chiasson, J.M. Frahm, F. Monrose, P.C. van Oorschot, Security analysis and related usability of motion-based CAPTCHAs: Decoding codewords in motion, IEEE Trans. Dependable Secure Comput. 11 (5) (2014) 480–493.

[23] C.J. Liao, C.J. Yang, J.T. Yang, H.Y. Hsu, J.W. Liu, A game and accelerometer-based CAPTCHA scheme for mobile learning system, in: Jan Herrington et al. (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2013, 2013, pp. 1385–1390.

[24] B. Choudhary, J. Risikko, Mobile Financial Services Business Ecosystem Scenarios & Consequences, Mobey Forum Mobile Financial Services Ltd., 2006.

[25] Sim alliance White Paper. Secure Authentication for Mobile Internet Services, Critical Considerations V1.1, 2011.

[26] EMV Mobile Contactless Payment, Technical Issues and Position Paper, 2007.

[27] Karin Braeckle, Infineon Security and RF Components support Samsung Galaxy S7 and Galaxy A smartphone series, 2016. http://www.infineon.com/cms/en/about-infineon/press/press-releases/2016/INFXX201602-032.html.

[28] Gil Bernabeu, Accessing GlobalPlatform Secure Component from a Web Application, in: W3C Workshop Synopsis:Referencing and Applying WCAG 2.0 in Different Contexts W3C Workshop, Brussels, Belgium, 2013.

[29] Global Platform, Trusted Execution Environment (TEE) Guide. http://www.globalplatform.org/mediaguidetee.asp.

[30] ARM TrustZone. https://www.arm.com/products/security-on-arm/trustzone.

[31] J. Mick, ARM to bake on-die security into next gen smartphone, tablet, PC cores, April 2012. http://www.dailytech.com/ARM+to+Bake+OnDie+Security+Into+Next+Gen+Smartphone+Tablet+PC+Cores/article24372.htm.

[32] Global Platform's White Paper. The Trusted Execution Environment: Delivering Enhanced Security at a Lower Cost to the Mobile Market, 2011.

[33] http://www.fingerprints.com/blog/2012/11/12/fpc-completes-integration-of-fingerprint-authentication-for-secure-element-targeting-secure-and-convenient-nfc-transactions-in-mobiles-and-tablets/.

[34] Patent. Smart card with additional electronic means EP 2128805 B1.

[35] A. Armando, A. Merlo, L. Verderame, An empirical evaluation of the android security framework, in: Security and Privacy Protection in Information Processing Systems, vol. 405, IFIP Advances in Information and Communication Technology, Springer, 2013, pp. 176–189.

[36] A. Armando, A. Merlo, M. Migliardi, L. Verderame, Breaking and fixing the android launching flow, Comput. Secur. (ISSN: 0167-4048) 39 (Part A) (2013) 104–115. http://dx.doi.org/10.1016/j.cose.2013.03.009.

[37] E. Owusu, J. Han, S. Das, A. Perrig, J. Zhang, ACCessory: password inference using accelerometers on smartphones, in: Proceedings of the 12th Workshop on Mobile Computing Systems & Applications, San Diego, California, USA, 2012, pp. 1–6. http://dx.doi.org/10.1145/2162081.2162095.

[38] L. Simon, R. Anderson, PIN skimmer: Inferring PINs through the camera and microphone, in: SPSM'13: Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices, ACM, New York, NY, USA, 2013, pp. 67–78. http://dx.doi.org/10.1145/2516760.2516770.

[39] L. Cai, H. Chen, TouchLogger: Inferring keystrokes on touch screen from smartphone motion, in: HotSec'11: Proceedings of the 6th USENIX Conference on Hot Topics in Security, San Francisco, California, USA, 2011, pp. 9–9.

[40] Z. Xu, K. Bai, S. Zhu, Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors, in: WISEC'12: Proceedings of the fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks, Tucson, Arizona, USA, 2012, pp. 113–124. http://dx.doi.org/10.1145/2185448.2185465.

[41] Ihtesham Haider, Michael Höberl, Bernhard Rinner, Trusted sensors for participatory sensing and IoT applications based on physically unclonable functions, in: Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security, (IoTPTS'16), ACM, New York, NY, USA, 2016, pp. 14–21. http://dx.doi.org/10.1145/2899007.2899010.

[42] Intrinsic-ID Physical Unclonable Functions Technology. https://www.intrinsic-id.com/physical-unclonable-functions/.

[43] Verayo, Physical Unclonable Functions (PUF). http://verayo.com/tech.php.

[44] Verayo, Introduction to Verayo. http://www.rfidsecurityalliance.org/docs/Verayo_Introduction_RFIDSA_July_9_08.pdf.

[45] Microsemi, SmartFusion2. http://www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2.

[46] NXP adds PUF Anti-Cloning technology to its next generation SmartMX2 microcontroller, September 2016. https://www.intrinsic-id.com/nxp-adds-puf-anti-cloning-technology-next-generation-smartmx2-microcontroller/.

[47] M. Potkonjak, V. Goudar, Public physical unclonable functions, Proc. IEEE 102 (8) (2014) 1142–1156. http://dx.doi.org/10.1109/JPROC.2014.2331553.

[48] A. Dua, N. Bulusu, W.C. Feng, W. Hu, Towards trustworthy participatory sensing, in: Proceedings of the 4th USENIX Workshop on Hot Topics in Security, USENIX, 2009.

[49] S. Saroiu, A. Wolman, I am a sensor, and I approve this message, in: Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications, ACM, 2010.

[50] T. Kim, J.H. Yi, C. Seo, Spyware resistant smartphone user authentication scheme, Int. J. Distrib. Sens. Netw. 2014 (2014) 7. http://dx.doi.org/10.1155/2014/237125.

Meriem Guerar received the Master degree in Information Systems and Networks from the University of Sciences and the Technology of Oran (USTO), Algeria, in 2011, and her Ph.D. in 2017. Her main research interests include the areas of authentication and identity management, security and usability, smartphone security and payment systems security.

Alessio Merlo got a M.Sc. in Computer Science in 2005 at University of Genova. He received his Ph.D. in Computer Science from 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 at the Universita' degli Studi di Genova, Italy, where he collaborates with the CSec Lab at DIBRIS. His currently research interests are focused on performance and security issues related to Web, distributed systems (Grid, Cloud) and mobile (Android platform). He is a member of the IEEE Computer Society and ACM. He is participates to program committees of international conferences (IFIP-SEC, AINA, ARES, HPCS,…) and he is member of the Editorial Board of an International Journal (Journal of High Speed Networks).

Mauro Migliardi got his Ph.D. in Computer Engineering in 1995. He was a Research Associate and Assistant Professor at the University of Genoa and Research Associate at Emory University; currently he is Associate Professor at the University of Padua, Adjunct Professor at 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. startup. His main research interest is 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 Ph.D. students at the Universities of Genoa, Padua and Emory, and (co-)authored more than 120 scientific papers.