A Completely Automatic Public *Physical* test to tell Computers and Humans Apart: a way to enhance authentication schemes in mobile devices

Meriem Guerar
Univ. Oran Mohamed Boudiaf
Oran, Algeria
and
University of Padua
Padua, Italy
Email: meriem.guerar@univ-usto.dz

Alessio Merlo
University of Genoa
Genoa, Italy
Email: alessio.merlo@unige.it

Mohamed Benmohammed
University of Constantine
Constantine, Algeria
Email: ben_moh123@yahoo.com

Mauro Migliardi
University of Padua
Padua, Italy
Email: mauro.migliardi@unipd.it

Belhadri Messabih
Univ. Oran Mohamed Boudiaf
Oran, Algeria
Email: belhadri.messabih@univ-usto.dz

***Abstract*—Nowadays, data security is one of the most – if not the most important aspects in mobile applications, web and information systems in general. On one hand, this is a result of the vital role of mobile and web applications in our daily life. On the other hand, though, the huge, yet accelerating evolution of computers and software has led to more and more sophisticated forms of threats and attacks which jeopardize user's credentials and privacy. Today's computers are capable of automatically performing authentication attempts replaying recorded data. This fact has brought the challenge of access control to a whole new level, and has urged the researchers to develop new mechanisms in order to prevent software from performing automatic authentication attempts. In this research perspective, the Completely Automatic Public Turing test to tell Computers and Humans Apart (CAPTCHA) has been proposed and widely adopted. However, this mechanism consists of a cognitive intelligence test to reinforce traditional authentication against computerized attempts, thus it puts additional strain on the legitimate user too and, quite often, significantly slows the authentication process. In this paper, we introduce a Completely Automatic Public Physical test to tell Computers and Humans Apart (CAPPCHA) as a way to enhance PIN authentication scheme for mobile devices. This test does not introduce any additional cognitive strain on the user as it leverages only his physical nature. We prove that the scheme is even more secure than CAPTCHA and our experiments show that it is fast and easy for users.**

I. Introduction

The growing dependence of daily activities from information systems of different kinds is making security and data protection a paramount problem, and one of the key aspects of security is authentication and access control. Historically, authentication has been studied as a way to identify with a very high level of certainty a single user among a set of potential users (authentication) and thus guarantee that only a selected set of users could access a given resources (access control). These mechanisms have been part of security long before the advent of "computer" security.

A classic taxonomy of the authentication mechanisms distinguishes between those based on *something you know* (e.g. a password), *something you own* (e.g. a token) and *something you are* (e.g. a biometric parameter). Among these, the most widely and commonly used mechanism is based on *something you know*. In fact, most authentication controls are still based on an secret information such as a password, a PIN, or a sequence of gestures.

The increasing availability of computers, however, has radically changed the landscape; in fact, the capability of computers to perform access tries automatically has forced the security mechanisms to introduce additional constraints on secrets to make them harder to crack in an automated, trial and error way. These constraints (e.g. longer passwords that include non-alphabetic symbols, changing the password often, etc.) make more difficult for humans to remember, force them to stop relying exclusively on their own memory, and, thus, very often completely foil the whole concept behind the *something you know* categorization. In fact, as users write complex passwords in any kind of stable storage, they turn the authentication into an awkward, indirect sort of the *something you own* mechanism.

In order to overcome this problem, some authors have introduced a new dimension to the traditional authentication taxonomy and have adopted Completely Automatic Public Turing test to tell Computers and Humans Apart (CAPTCHA) as an additional security mechanism. We argue that all the

978-1-4673-7813-0/15/$31.00 ©2015 IEEE

mechanisms classified by the traditional authentication taxonomy have been originally devised to perform a human vs. human selection and, because of the advent of automated attempts to overcome them, they have simply grown in complexity often disregarding the actual nature of the threat. At the same time, the root of the CAPTCHA concept, that is, the Turing test, targets the difference in cognitive capabilities between humans and computers; thus, 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 makes CAPTCHAs very often inconvenient and hard to solve even for human subjects.

In this paper we propose to adopt as the discerning factor telling computers and humans apart not intelligence or cognitive capabilities in general but their physical nature.

In the past, computers did not lend themselves to the evaluation of physical quantities; 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 interact physically with the device.

In the past, several papers presented different security mechanisms involving the exploitation of smartphones and tablets sensors to involve the human user in an activity that could tell him apart from a computerized program. However, in all the past cases, the activity involved some sort of cognitive action that made it an implementation of the Turing test. 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 could be defined as a Completely Automatic Public Physical test to tell Computers and Humans Apart (CAPPCHA) and is expected to enhance the ease of use of this new generation of authentication methods while guaranteeing resilience to automated attacks.

To prove the efficacy and ease of use of this kind of techniques we have devised a practical implementation of a sensor based CAPPCHA and we have combined it with a simple PIN based authentication. In our experiments this combination provides a very high level of simplicity, resilience to automated attack and does not introduce a cognitive overload on the users.

The paper is structured as follows: in Section 2 we provide an overview of related work; in Section 3 we describe the CAPPCHA scheme we propose; in Section 4 we provide a security analysis of the proposed scheme against some well-known types of attack; in Section 5 we describe our preliminary experimental results and, finally, in Section 6 we draw our conclusions.

II. Related work

Most current commercial applications still use static alphanumeric passwords or PIN codes as a mean of authentication in sensitive transactions, even though these methods are known by their vulnerability to spyware attacks. Current approaches to enhance the security of these widely adopted methods can be categorized into two categories. Approaches in the first category rely on adding additional secret value to the PIN/password. Approaches in the second category ask users to solve challenge-response test such as CAPTCHA to allow them to enter their PIN/password.

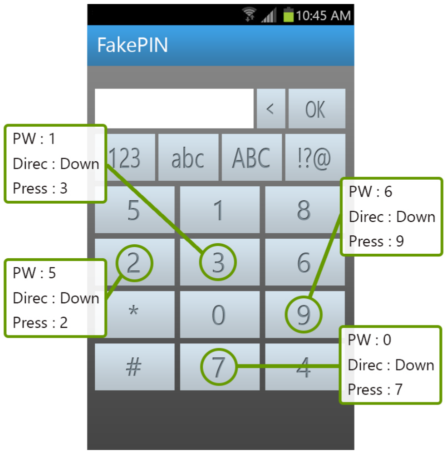


Fig. 1: Screenshot of FakePIN scheme.

Recently, Kim et al. [1] proposed a password authentication scheme, called FakePIN (Fig. 1). In their scheme, the user has to memorize an alphanumeric text and 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 original password with password direction. Whereas, the location of the keypad letters is changed randomly for each authentication in order to prevent attacker from replaying the user input 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, this scheme is not resilient against multiple recording attacks. Similar to FakePIN, Yi et al. [2] add an additional secret value, called *pass-icon* (Fig. 2), to allow the user to enter their PIN in a secure way. The user has to memorize the location of pass-icon within a window. The authentication is performed by moving the window on the virtual keypad using multimodal sensors in such way that the location of pass-icon moves over the PIN. While, this mechanism enhances the security of PIN codes against spyware attacks, their user study showed that the authentication speed is very low (i.e 17,86 sec).

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 [3] (Fig. 3). Their approach consists of using scanned words from old books that Optical Character Recognition (OCR) program failed 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

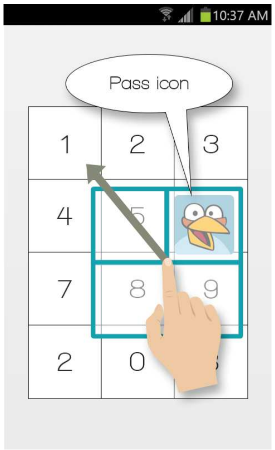


Fig. 2: Screenshot of PassWindows scheme.

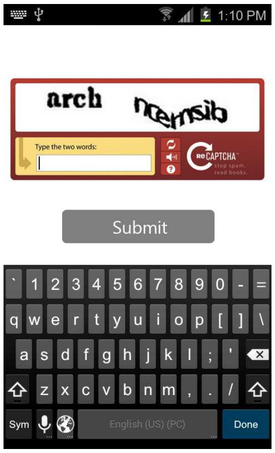


Fig. 3: Screenshot of ReCAPTCHA Test

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. [4] using neural networks with an accuracy of 99.8%. In addition to the security issue, a recent research [5] 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 [28], authors suggested alternative input mechanisms aimed at improving the usability of ReCaptcha on smartphone. However, their usability comparison results show that the participants prefer the existing ReCaptcha scheme, that uses the virtual keyboard as primary input.

Chow et al [6] introduce the idea of clickable CAPTCHAs (Fig. 4) in order to make the CAPTCHAs suitable for mobile devices. Their approach consists of combining multiple textual CAPTCHAs into a grid of clickable CAPTCHAs (e.g. a 3-by-4 grid).The user has 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 the entry of string of characters using the mobile keyboard which is challenging. Despite showing some advantages, this scheme has not been widely deployed.

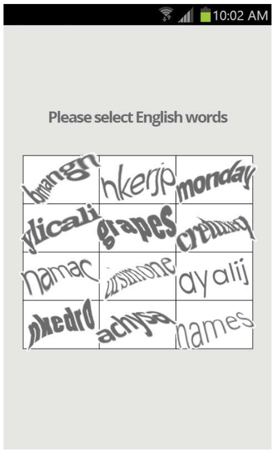
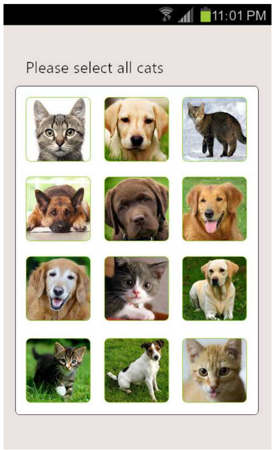


Fig. 4: Example of a clickable CAPTCHA. The user's task is to identify the three valid English words (in this example: monday, grapes and names).

Pequegnot et al. [7] suggested an authentication mechanism based on graphical Turing test to increase confidence in mobile transactions. Their mechanism consists of typing in a secure code of three-digit displayed in a CAPTCHA, in addition to the four-PIN digits. The secure code is randomly generated by the secure element for each authentication session. In this way, the system prevents mobile transactions submission by malwares. 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, e.g., [4], [8], [9], [10] and [11]. In addition, using too much noise and distortions makes them harder for humans to decipher as well, especially on tiny screens.

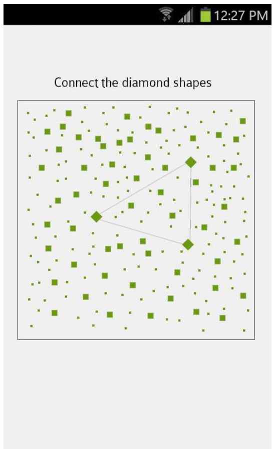
An alternative to text-based CAPTCHA forms are image-based CAPTCHA. A typical CAPTCHA of this kind is Asirra [12] (Fig. 5), which display 12 images of cats and dogs and asks users to select all cat images among them. Their user study shows that solving the Asirra challenge takes time under 30 seconds for 96.6% of humans which is advantageous compared with text-based CAPTCHAs. However, Golle [13] showed that this scheme is vulnerable to machine learning attacks. Shirali-Shahreza et al. [14] proposed CAPTCHA mechanism for mobile devices, called Drawing CAPTCHA (Fig. 6). In this method, numerous dots are displayed on a screen with 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 [15], an image-processing technique was proposed by Lin et al to breaks the Drawing CAPTCHA with an accuracy of 75%.

Another form of CAPTCHA that has been introduced in



Please select all cats

Fig. 5: Screenshot of Asirra challenge.

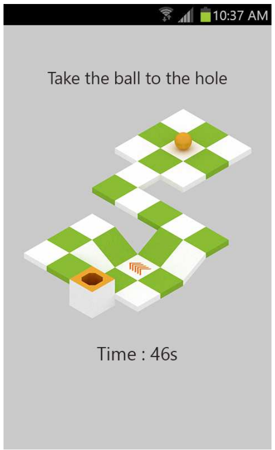


Connect the diamond shapes

Fig. 6: Screenshot of drawing CAPTCHA.

recent years is game-based CAPTCHA. Liao et al [16] proposed accCAPTCHA (Fig. 7), 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). In this scheme, the motion operations are performed through moving the device in the case of accelerometer-enabled device or by touching the screen in the other case. Authors claimed that is difficult for computer to understand the meaning of these games and thus, to provide the correct response. However, their user study shows that the most 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).

In this paper, we propose a sensor based CAPPCHA to enhance the security of PIN code without asking user to memorize an additional secret value or solving complex cognitive task that, in the most cases, are not suitable for mobile device and take long time which add annoyance to users.



Take the ball to the hole

Time : 46s

Fig. 7: Screenshot of accCAPTCHA

III. Proposed scheme

A. Context

Today's smartphone platforms give the user the ability to customize their device through the million applications available on the market or traditional websites. However, these possibilities come with potential risks of installing malicious apps that may steal sensitive user data or gain root access to their device (e.g [29]) . In order to increase the security of PIN codes which are used to access sensitive mobile services against software attacks, generated in a Rich OS environment such as Android, there are industry-led initiatives to use a Trusted Execution Environment (TEE) or a Secure Element (SE). **Secure Element (SE)** is a combination of hardware, software, interfaces and protocols embedded in a mobile handset [17]. This component provides a secure platform which enables an isolated execution to 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 [18].

Trusted Execution Environment is a secure processing environment which is isolated from the Rich Execution Environment (REE) where the device operating system and applications run. This execution environment ensures that sensitive applications (e.g. payment, banking, corporate emails, etc.) are stored, processed and protected in a trusted environment which enforces the protection, confidentiality, integrity and data access rights [19]. An example of commercial deployment of TEE on mobile devices is based on ARM TrustZone technology. This technology has gained wide acceptance and development in recent times, we can find it shipped in many phones on the market today (e.g., Samsung Galaxy S3/S4).

B. Assumptions and threat model

Phone architecture: In this paper, we assume a TEE-enabled smartphone such as samsung Galaxy S3/S4. Otherwise, we assume that the smartphone uses a secure element with an embedded accelerometer sensor. Secure elements with

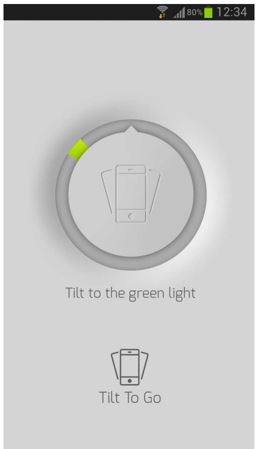


Fig. 8: Screenshot of CAPPCHA Test

embedded sensor such as sweep sensor are already commercially available thanks to the collaboration between Fingerprint Cards (FPC) and Infineon Companies [20]. Furthermore, Michel Willemin [21] recently showed the prototype of a smart card (i.e. SIM card or memory card) equipped with motion sensor that acts as an accelerometer.

Infection: To prove the resilience of our scheme, we assume that the user has naively installed a malicious application from app stores or other website that has exploited a vulnerability in Android OS and has gained root access on the device. As stated above, we also assume that the sensitive application is running in one of the over mentioned components and is protected with the common PIN code. Despite the strong isolation provided by these components to protect sensitive application, the malicious code running in Android OS can steal the user's PIN code and replay this PIN in the next authentication session to perform unwanted transaction without the user's awareness [7]. In order to solve this problem, we propose an authentication mechanism for both TEE-enabled smartphones and the other devices without TEE.

C. *CAPPCHA test concept*

Despite the variety of user authentication methods proposed in academic research, most of the current commercial applications still use PIN codes though they are known to be insecure. This is due to their simplicity, ease of remember and input. In order to enhance the security of this common method against mobile malware, we add a simple CAPPCHA test to the PIN entry without affecting its usability. In our proposed scheme, instead of existing schemes that ask users to solve complex cognitive task or memorizing additional secret value with complicated input processes, users have only to tilt the device to a specific degree displayed in the screen and hold it still in this position for one second to have access to the PIN pad (see the Fig. 8). The basic idea behind this is to use something the user can do it easily while the spyware cannot. The challenge degree is generated randomly 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.

In the case of smartphones that use a secure element with embedded accelerometer sensor, there is no need to generate random challenge degree for each authentication session because the sensitive application gets the motion value of device from trusted hardware (i.e. accelerometer) that can't be fooled by a rootkit infecting the device. Thus, in this case a user would have the possibility to choose the degree of challenge that he finds himself more comfortable to turn the device to it. It is important to notice that, even if a malware could grasp the tilt value by using the non-protected accelerometer in the device, it cannot fool the accelerometer inside the secure element.

IV. Security analysis

A. *Brute Force attack*

One of the main security issues of PIN authentication is the *brute force attack* where attackers try 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 very difficult or impossible for an automated computer. In the first case using ARM TrustZone, the randomization of the challenge degree for each authentication session prevents the automated process of iterating through the entire circle space. Whereas, the second case using the secure element with embedded sensor, the automated process is unable to perform a brute force attack. This is due to the fact that malicious codes can't simulate the motion values stored securely in the secure element and are unable to move the device for solving the challenge. Thus, the proposed scheme is resilient against brute force attack.

B. *Side channel attack*

Recently, a new attack trend targeted at stealing user's keystrokes even when strong isolation protects sensitive input has emerged. This kind of attack leverages resources that are shared between the mobile OS and the trusted OS, such as the accelerometer [22], the camera and the microphone [23], the Gyroscope [24], [25], etc.

The proposed scheme, running in the TEE-enabled device, prevents side channel attacks through the randomization of the challenge degree for each authentication session. In this way, the current motion value of device, measured by accelerometer, can't be used in the next transaction. Thus, stealing the response to the current challenge (the equivalent of the PIN code) is useless because the malicious code is unable to solve the challenge. In the other case, when the proposed security mechanism is running in the secure element with embedded accelerometer, it provides protection against side channel attack even without randomizing the challenge degree. This is because the proposed scheme measures the motion value from trusted hardware and this value cannot be changed by the malicious code even if it has the root access to the device: instead, it can be changed only when the user physically moves the device. Hence, the proposed scheme ensures the security against side channel attacks in the both cases.

C. Recording attack

The problem of mobile malware has been greatly aggravated with 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 [26] which makes defense against this attack very difficult. However, in the TEE-enabled device, when users interact with the Trusted OS, the Android OS cannot access the screen [23] providing a secure input path to the user. Hence, the malicious code cannot take screenshot when the proposed scheme is running in a TEE such as ARM TrustZone in order to reveal the challenge degree. Therefore, our system is resilient against recording attack. When using the secure element instead of TEE, the malwares that have root access to the device can take a screenshot and reveal the challenge degree since the secure element does not provide a secure access to the screen. However, this would be useless because the malware cannot fool the accelerometer inside the secure element into believing that the phone has been tilted to the correct angle: only the real user may apply physical movement thus efficiently separating computer programs from humans.

D. Theft of smartphone

Similarly to existing CAPTCHA schemes, a CAPPCHA test is devised to separate humans from machines. Thereby, it is obvious that humans are able to perform such test. Thus, if an attacker steals the mobile device, he will be able to bypass the first step (i.e. the CAPPCHA test) of the proposed authentication mechanism easily; however, the attacker still has to overcome the second authentication step. Yet, even in the presence of a human thief and a physical acquisition of the device, an authentication mechanism that requires human intervention at each try slows down significantly the rate of the attack.

V. Experimental results

Designing a new authentication mechanism requires to take in consideration both computer security and human-friendliness. Therefore, in this section, we evaluate the usability of the proposed scheme and we compare it to some existing authentication schemes. 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.

To test our security mechanism, we need a TEE-enabled smartphone or secure element with embedded accelerometer. Unfortunately, Samsung Galaxy S3 features a TrustZone-based TEE called MobiCore developed by G&D [27], but it is not allowed at the moment to develop on it without certification from G&D. On the other hand, a secure element with embedded accelerometer is not available yet in the market. Hence, we used Samsung Galaxy S3 (1.4 GHz Dual-Core CPU, 1 GB RAM) equipped with a Giesecke&Devrient Mobile Security Card (MSC) SE 1.0 to test our mechanism. Furthermore, the communication between the third-party application and the MSC SE 1.0 is performed via the Seek-for-android API. The following development tools have been used for implementation: Eclipse KEPLER SR2, android SDK 4.04 and JAVA 1.7.0. 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 8) as a challenge. We recruited 10 volunteers to participate in the study. The average age was 24 years, ranged from 22 to 32 years. Before starting the test, the proposed mechanism was explained in detail and a random PIN was assigned to each participant. They were asked to test the application until they felt familiar with the system. Afterwards, participants were required to enter a PIN for five times. Thus, the results are based on 50 authentication sessions performed by 10 participants. Authentication time (i.e. time of CAPPCHA test plus time of PIN-entry) was measured from starting the application to releasing the last PIN digit.

TABLE I: User test results for the different systems. Related work results taken from the original papers.

Authentication method	Authentication time [s]	Error rate [%]
FakePIN [1]	14.13	4.70
PassWindow [2]	17.86	4
Proposed scheme	4.12	0

Table I shows the average authentication time and error rate of our scheme along with two schemes from the first category mentioned in the related work. Table II summarizes the average challenge time and the error rate of some related works from the second category.

Despite the fact that the challenge timings showed in the second table represent only part of authentication time, we see that our proposed scheme has by far the fastest authentication time and lower error rates among all the evaluated schemes. We argue that the main reason of this is the fact that, once automated attacks have been made impossible with the CAPPCHA part (where the user only has to move the device to a specific degree),our scheme adopts a familiar four-digit PIN without adding additional secrets value with complicated input processes or asking users to solve complex cognitive tasks. In this way, our scheme improves the global level of security and remains fast and easy to use. Furthermore, it is important to notice that the CAPPCHA mechanism we implemented does not allow errors, it may only delay the completion of the first part of the authentication. Thus, the error rate in our experiments is actually the error rate of the PIN entry.

VI. Conclusion

In this paper, we proposed a new authentication mechanism that increases the security of common PIN codes against different malware attacks and remains easy and fast to use. The proposed scheme uses a CAPPCHA test which asks users to move their devices to a specific degree to prove that they

TABLE II: User test results taken from the original papers.

Authentication method		Avg. challenge time [s]	Error rate [%]
Clickable captcha [6]	- Subjects familiar with English	11.1	0
	- Subjects not familiar with English	18.2	10 – 20
AccCaptcha[16]	- Stack Game	47.3	33
	- Rolling ball game	25.2	22
	- Racing game	55	4

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. From a usability point-of-view, the results of our experiments show that the proposed scheme offers a short authentication time and a zero error rate (albeit the testbed must be extended once the prototype implementation will be stabilized).

The comparison with existing schemes which are resilient to automated attacks shows that our scheme provides the same security strength with considerably low authentication time and error rates. Thus, it has the potential to replace current authentication systems.

VII. Acknowledgement

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] Kim, S., Yi, H., Yi, J.H.: FakePIN: Dummy Key Based Mobile User Authentication Scheme. In Ubiquitous Information Technologies and Applications, Volume 280 of Lecture Notes in Electrical Engineering , pp.157-164, Springer, Berlin (2014).

[2] Yi, H., Piao, Y.,Yi, J.H.: Touch Logger Resistant Mobile Authentication Scheme Using Multimodal Sensors. In: Advances in Computer Science and its Applications, Volume279 of Lecture Notes in Electrical Engineering , pp.19-26.Springer, Berlin (2014).

[3] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science,September 2008.

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

[5] Reynaga, G., Chiasson, S.: The Usability of Captchas on Smartphones. In: Proceedings of SECRYPT pp.427-434,SciTePress (2013).

[6] R Chow, P Gollé, M Jakobsson, X Wang, L Wang. “Making CAPTCHAs clickable”. Ninth Workshop on Mobile Computing Systems and Applications (HotMobile 2008). 2008 February 25-26; Napa, CA.

[7] Pequegnot, D., Cart-Lamy, L., Thomas, A., Tigeon, T.,Iguchi-Cartigny, J. ,Lanet, J-L.: A Security Mechanism to Increase Confidence in M-Transactions. In CRiSIS: 6th International Conference on Risks and Security of Internet and Systems, pp.9-16. IEEE, Timisoara, Romania (2011). Doi:10.1109/CRiSIS.2011.6061836

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

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

[10] J Yan and A S El Ahmad. “A Low-cost Attack on a Microsoft CAPTCHA”, School of Computing Science Technical Report, Newcastle University, England. Feb, 2008.

[11] M. Korakakis, E. Magkos and Ph. Mylonas.“Automated CAPTCHA Solving: An Empirical Comparison of Selected Techniques“. In 9th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), pp.44-47. IEEE, Corfu, Greece (2014).

[12] J Elson, JR Douceur, J Howell and J Saul. “Asirra: a CAPTCHA that exploits interest-aligned manual image categorization”. Proceedings of the 14th ACM conference on Computer and communications security (CCS), 2007.

[13] P. Golle, “Machine learning attacks against the Asirra CAPTCHA,” In: Proceedings of the 15th ACM conference on Computer and communications security, pp. 535-542, 2008.

[14] M. Shirali-Shahreza and S. Shirali-Shahreza, Drawing CAPTCHA. Proceeding of the 28 international conference on information technology interfaces, Cavtat, Croatia, 2006, pp. 475-480.

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

[16] Liao, C.J., Yang, C.J., Yang, J.T., Hsu, H.Y. and Liu, J.W. (2013). 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 (pp. 1385-1390).

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

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

[19] http://www.globalplatform.org/mediaguidetee.asp

[20] 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/

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

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

[23] Simon, L., Anderson, R., 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, pp. 67–78. ACM, New York, NY, USA(2013). Doi :10.1145/2516760.2516770

[24] Cai, L., Chen, H.: TouchLogger: Inferring Keystrokes On Touch Screen From Smartphone Motion. In HotSec’11: Proceedings of the 6th USENIX conference on Hot topics in security, pp.9–9. San Francisco, California, USA (2011).

[25] Xu, Z., Bai, K.,Zhu, S.: Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. InWISEC ’12 : Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks, pp.113-124. Tucson, Arizona, USA (2012). Doi:10.1145/2185448.2185465

[26] Kim, T., Yi, J.H., Seo, C.: Spyware Resistant Smartphone User Authentication Scheme. Int.J.DistributedSens.Netw. Vol 2014, P.7 (2014). Doi:10.1155/2014/237125

[27] Giesecke& devrient: Creating confidence.”http://www.gi-de.com/en/index.jsp.

[28] Reynaga, G., Chiasson, S., and Van Oorschot, PC.: ”Exploring the Usability of CAPTCHAS on Smartphones: Comparisons and Recommendations.” (USEC 2015).

[29] Bickford J., O?Hare R., Baliga A., Ganapathy V., and Iftode L., ?Rootkits on smart phones: attacks, implications and opportunities?. In HotMobile?10: Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications, pp. 49?54. ACM, New York, NY, USA (2010).