Securing Mobile Device Applications using Biometric Authentication

Andreea Florescu, Cristina Necula, Diana Popa
Computer Science Department, Faculty of Automatic Control and Computers
University Politehnica of Bucharest

***Abstract*—The ever growing advances in accessing any service from a personal mobile device and the extending capabilities that such a device offers, calls for more powerful methods of verifying the identity of users than the traditional ones which were based on knowledge and possession (e.g. passwords, patterns unlocking) making them more vulnerable to fraudulent activities. This paper aims at analyzing two methods for validating users' identity on Android mobile devices: fingerprint scanning and face recognition. We will analyze the performance, study user effort, error and task disruption for each of these biometric authentication methods. We will use the proposed methods as authentication mechanisms for an Android application. The evaluation process consists in gathering a signatures database from multiple subjects. Every subject will register a series of signatures which will be used in building a database with registration templates. Another series of tests will be performed, comparing a new series of signatures coming from the same subjects with the ones registered in the database. Thus, we will calculate the acceptance rate (true and false), user effort and study errors. The purpose of this project is to realize a comparison between the two systems and the security they offer, and also to analyze the possibility of integrating such techniques of validating users' identity for increasing transactions security.**

***Index Terms*—biometrics, mobile devices, security, facial recognition, fingerprint detection**

I. Introduction

In the following sections of this paper we analyze two methods for authenticating users into their personal mobile device from a comparative perspective by taking into account the acceptance rate, user effort and performance errors. The applied measuring technique will also be presented, we will debate around the experimental results and conclude upon the technique that dominates in term of a security point of view.

....

This paper is organized in the following manner: section II provides an overview on previous work on fingerprint scanning and facial recognition with focus on work related to performance issues, challenges and security concerns. Section III presents a complete description of the experimental methodology used to compare these two analyzed techniques on an Android mobile device. Section IV presents, from a comparative perspective, the results obtain for both facial and fingerprint recognition and debates cons and pros around the idea of incorporating these two methods for securing transactions over the Internet.

II. Related Work

In order to be able to compare these two current popular biometric techniques on Android devices, we firstly analyze these methods separately by providing past research regarding each one of these techniques to better grasp the problems encountered when implementing them and be able to identify advantages and disadvantages when forced to choose between them. We will also look at previous research work that offers a comparative overview to these methods by providing interesting statistical data for each one of them.

A. Fingerprint Scanning

The number of Android phones that nowadays integrate the capability of a fingerprint scanner is continuously increasing (HTC One M9+, Samsung Galaxy A8, OnePlus 2, Meizu MX5). Needless to say, the framework used for authenticating a fingerprint has also changed greatly over the years and these adjustments have brought with them major security enhancements.

In this study from 2015 [17], the authors analyze the evolution of the frameworks used for scanning and evaluating fingerprints in order to show the improvements in key security aspects. They outline the main vulnerability of the initial framework of Android phones (including the HTC One Max) as being the fact that the safety of the biometric template was dependent on how secure the kernel was given that the fingerprint drivers resided directly in the kernel. The framework, that was later developed to solve the problem posed by this vulnerability, integrates the ARM TrustZone™[2] technology inside the architecture. This technology is in fact an extension that splits the execution environment into a **Normal** one, which is the one that is not secure (e.q the user or the kernel mode), and into a **Trusted** one which sensitive data (i.e. fingerprint templates) can be kept safe. The TrustZone™ extension is actually the current technology used in Android devices which offers solutions to not only protecting the data involved in scanning and identifying fingerprints, but to a variety of security requirements.

Despite all the popularity that the TrustZone™ technology has gained, the only author of [15] gives a full and detailed description of two major vulnerabilities that he was able to discover in all the Huawei™ devices with some specific chipset. He proves his ability to exploit them to bypass security requirements imposed by TrustZone™ to succeed in making a local application obtain fingerprint images from the scanner. In this way, the local application would be able to change them in order to control the final verification step of the smartphone's biometric system.

Besides the challenges related to keeping the fingerprint data safe, there are several other challenges that this particular biometric method involves. One problem when it comes to developing an efficient algorithm, is how to deal with the impartially collected fingerprints. Due to design reasons, a mobile device usually incorporates a small rectangular scanner which, depending on the user's physical traits, may not collect a full fingerprint, and this is a challenge that was researched in the past. One very recent algorithm that tries to surpass this obstacle is described in [11]. An important mentioning is made by the authors which state that previous methods mainly knew how to deal with the scenario where the user would input a partial fingerprint when trying to authenticate into the mobile device but the database contained full fingerprints (gathered at time of enrollment). So, basically the authors find a solution to be able to identify current partial fingerprints from pre-registered partial fingerprints. The key idea of their algorithm is guiding the user at enrollment time into inputting his fingerprint into two positions: one horizontal and one vertical. The results obtained upon comparing their method with other two popular methods (VeriFinger™ [12] and SIFT™ [13]), show performance gain as their implementations gives the highest true acceptance rate on all databases tested.

A very large variety of algorithms that can match fingerprints can be identified, but as per [5], the most used one in Android devices and in mobile devices in general is the minutiae-based algorithm because they are more efficient and they have proven their performance in time. The minutiae represent the discriminatory points of a fingerprint and they can be classified into: *ridge ending* (some papillary that ends abruptly) and *ridge forking* (some point where the papillary bifurcates). Given that all fingerprint matching algorithms on Android devices are minutiae-based, a relevant study is represented by [9] , where the authors work at obtaining the performance results of several such algorithms. The metrics that they use to perform this comparative analysis are: matching score (the sum of total correct correlation between template data and input testing data), execution time (the duration of the matching in milliseconds). It is interesting that in the end, there are two algorithms that show best results depending on database used. One of them performs the best on low quality images while the other one performs best on medium and high quality images.

B. Facial Recognition

Google introduced Face Unlock as a method to unlock an Android mobile device since version 4.0. The method uses the mobile phone frontal camera to record an image of the person who possesses the device. The recorded picture is then used for facial recognition, utilizing the technology developed by the PittPatt company. The code Google used to implement Face Unlock is proprietary and there are no public details about how data is stored or the implementation of the face recognition algorithm [6]. Even though, we can consider that data is not easy to obtain as a warning screen informs the user that "The data that's used to recognize your face is kept private on your phone" when activating Face Unlock.

A lot of research have been made in the area of face recognition and face detection. Guillaume et all [3] perform a comparative study of algorithms for face recognition on mobile phones, having in mind the tradeoff between accuracy and computational complexity because of the limited resources of a mobile device. Eigenface [8] and Fisherface [10] were used for face recognition, experimental results showing equal error rate of 35% for Eigenface and 25% for Fisherface. Fisherface algorithm performed better for fluctuating lightning conditions.

One problem that raises when using frontal images to secure devices or applications is the high availability of pictures of device owners from social media or internet. Rainhard et all [7] propose a new approach to Face Unlock: usage of both front and profile face information, which will be captured during a pan shot around the device owner's head, thus making it more secure than regular face detection. The method was tested using a set of 4-5 images for each subject, recorded at different angles: 90°, 45°, 0°, -45°and -90°, 0°representing the frontal image. The issue with their proposed system is that results showed 90.5% detection rate and 55.8% recognition rate for profile images.

Paper [14] identifies the security issues in Face Unlock. Spoof attacks can be performed using printed pictures, video replays and 3D masks. Considering printed and video attacks more accessible and low cost, authors perform a study on spoofing detection, using multiple existing spoofing detection databases. A series of factors were analyzed: influence of image acquisition device, influence of different image regions, influence of color channel, influence of IPD or influence of database size. They implemented a new method for spoofing detection that combines previous presented aspects. Tested in real-life scenarios, the method proved to have an accuracy of 96.0% accuracy in detecting faces on a Google Nexus 5. Regarding the spoof attack, the application had an accuracy of 77.5%.

Both Face Unlock and Fingerprint scanning used to unlock Android devices require an additional authentication mechanism (PIN or pattern), in case biometric authentication fails. When activating Face Unlock on a device, Android system warns the user that "Face-matching is less secure than a pattern, PIN or password" and that "Someone who looks like you could unlock your phone".

C. Comparative studies

Research comparing face recognition and fingerprint identification as authentication methods for android devices is rather scarce. Regarding the security as it is perceived by the users, we found users to be more inclined to use fingerprints rather than face recognition. In this survey [1], people invoke reasons like security, convenience and future usage when choosing fingerprint as a better authentication method for mobile devices.

Accuracy is another important matter in biometric authentication. In paper [16], fingerprint recognition is said to have an accuracy of 99% while face recognition is correct in 95% of the cases. This comparison is based on literature as stated in the article and is not strictly related to Android devices.

In study [4], the advantages and challenges of many biometric authentication means are presented. Both fingerprint and face recognition have security advantages over the classical passwords and identification theft is reduced as they require the presence of the user. The issue that is not yet addressed in fingerprint authentication is the failure rate when the mark is modified due to external factors (for example wet), thus making the user unable to login. Face recognition can also become unreliable when the subject has the face covered with different objects like hats or glasses. The performance can be negatively influenced by the removal of these objects.

III. Experimental Methodology

A. Fingerprint evaluation

B. Facial evaluation

IV. Experimental Results

Here will put graphs showing results in a comparative manner and talk about transactions...

V. Conclusions

References

[1] Noam Ben-Asher, Niklas Kirschnick, Hanul Sieger, Joachim Meyer, Asaf Ben-Oved, and Sebastian Möller. On the need for different security methods on mobile phones. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 465–473. ACM, 2011.

[2] Rob Coombs. Securing the future of authentication with arm trustzone-based trusted execution environment and fast identity online (fido). *ARM White Paper*, 2015.

[3] Guillaume Dave, Xing Chao, and Kishore Sriadibhatla. Face recognition in mobile phones. *Department of Electrical Engineering Stanford University, USA*, 2010.

[4] Krishna Dharavath, FA Talukdar, and RH Laskar. Study on biometric authentication systems, challenges and future trends: A review. In *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*, pages 1–7. IEEE, 2013.

[5] Octavian Dospinescu and Ilinca Lîsîi. The recognition of fingerprints on mobile applications–an android case study. *Journal of Eastern Europe Research in Business and Economics*, 2016.

[6] Nikolay Elenkov. *Android Security Internals: An In-Depth Guide to Android's Security Architecture*. No Starch Press, 2014.

[7] Rainhard D Findling and Rene Mayrhofer. Towards face unlock: on the difficulty of reliably detecting faces on mobile phones. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, pages 275–280. ACM, 2012.

[8] Bahadir K Gunturk, Aziz Umit Batur, Yucel Altunbasak, Monson H Hayes, and Russell M Mersereau. Eigenface-domain super-resolution for face recognition. *IEEE transactions on image processing*, 12(5):597–606, 2003.

[9] Mohamed Lahby, Yassine Ismaili, Abdelbaki Attioui, and Abderrahim Sekkaki. Performance analysis of minutia-based fingerprint matching algorithms. In *Intelligent Systems: Theories and Applications (SITA), 2016 11th International Conference on*, pages 1–5. IEEE, 2016.

[10] Shang-Hung Lin. An introduction to face recognition technology. *Informing Science*, 3(1):1–8, 2000.

[11] Surbhi Mathur, Ankit Vjay, Jidnya Shah, Shreyasi Das, and Adil Malla. Methodology for partial fingerprint enrollment and authentication on mobile devices. In *Biometrics (ICB), 2016 International Conference on*, pages 1–8. IEEE, 2016.

[12] Ltd. Neurotechnologija. VeriFinger SDK fingerprint identification for stand-alone or web solutions. http://www.neurotechnology.com/verifinger.html. Accessed: 2016-12-10.

[13] opencv dev team. OpenCV sift in opencv. http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html#sift-in-opencv. Accessed: 2016-12-10.

[14] Keyurkumar Patel, Hu Han, and Anil K Jain. Secure face unlock: spoof detection on smartphones. *IEEE Transactions on Information Forensics and Security*, 11(10):2268–2283, 2016.

[15] Di Shen. Exploiting trustzone on android. In *Black Hat conference*, 2015.

[16] JA Unar, Woo Chaw Seng, and Almas Abbasi. A review of biometric technology along with trends and prospects. *Pattern recognition*, 47(8):2673–2688, 2014.

[17] Yulong Zhang, Zhaonfeng Chen, Hui Xue, and Tao Wei. Fingerprints on mobile devices: Abusing and eaking. In *Black Hat Conference*, 2015.