NFC4Sure

Mobile Ticketing System

Diogo Antunes, diogo.antunes@ist.utl.pt

Technical University of Lisbon - Taguspark Campus,
Av. Prof. Doutor Aníbal Cavaco Silva 2744-016 Porto Salvo, Portugal
http://www.ist.utl.pt/en/

Abstract. NFC4sure is a ticketing solution that integrates seamlessly with existent ticketing technologies, without using a hardware secure element in the mobile phone and not requiring a permanent online interaction. The scope of this thesis is presenting NFC4sure architecture and implementing an implicit authentication system capable of identifying and authenticate a mobile device user into the NFC4sure system. The way we achieve this is by using a mechanism that makes use of the user interaction with a mobile device screen to create a distinguishable fingerprint.

Keywords:
pubic transportation, ticketing system, implicit authentication, behavioral biometrics

1 Introduction

Most public transportation services (PT) (metro, train, etc.) feature electronic ticketing systems, to handle both the payment for the services by the customers and the management of the transport infrastructure. Smartcards are on the user's side of the electronic ticketing systems, enabling him to use the infrastructure. However, while smartcards were once a convenience for many people, they are now becoming a burden. Both in terms of the financial cost of acquiring a smartcard, but also due to the simple practical fact that smartcards do pileup in people's wallets.

To increase the usability and security of the ticketing systems, companies turned their attention to a device that most people have and use on their daily lives, the smartphone.

Has this happens, many companies around the world invest in making their business or product, that is available through a smartphone, more secure. This security may pass by authenticating the correct user, storing less sensitive information (token), and having a Secure Element (SE) to safely store the data.

In ticketing systems the communication usually occurs via Near Field Communication (NFC) [1].

[1] http://www.gsma.com/digitalcommerce/wp-content/uploads/2012/08/GSMA-Mobile-NFC-Infrastructure-v1-01.pdf

Typical smartphone NFC-enabled transactions (e.g. enter a PT), usually require the existence of an hardware SE (HSE), containing the necessary credentials. One example of such hardware SEs is the Universal Integrated Circuit Card (UICC) (also know as SIM card), provided by Mobile Network Operators (MNO).

Despite the security guarantees offered by these solutions, having a hardware SE provided by an MNO requires a certain degree of cooperation between companies and institutions, for instance the MNO may charge additional fees.

To address some of the issues posed by NFC, and the use of HSE, Host Card Emulation (HCE) was introduced. HCE emulates a virtual smartcard into an application, allowing the payment application to communicate directly with the NFC controller, and store the payment credentials on the application, instead of on the SE. The application must implement the security mechanisms necessary to support the transaction.

However, most developed HCE solutions come at a cost. In most cases HCE security is paired with a remote SE, which stores the payment credentials, introducing non-negligible usability problems.

This thesis is driven by the motivation that ticketing systems using NFC with HCE technology are not secure. With the usage of HCE a mobile device can emulate the functionality of a smartcard but exhibits a number of new security threats. The two biggest problems when using HCE is handling the communication of privileged data where there is always a necessity of verifying the owner of the data, or by other words, identify and authenticate the user.

2 Objectives

The scope of this project is substantially vast, in this thesis we present a ticketing architecture capable of providing a secure environment for the usage of HCE in PT. We also implement one authentication method used in section 4.2.2 that allows the identification and authentication of the user. The mechanism chosen is an implicit authentication method [9], based on the user normal interaction with a mobile device. After a thorough research in the area we settled on the work done by Mario Frank et Al, Touchalytics [7], that calculates a set of biometric features and applies machine learning techniques to generate a user fingerprint [4][12][10].

3 Related Work

In this section we start by explaining some ticketing solutions that make use of HCE. Finally in sections 3.2 3.3 3.4 we converge to the main scope of our project, and lay out out important work in the field of implicit authentication using mobile devices.

3.1 Public Transportation

In the domain of PT ticketing systems several HCE systems have been proposed. However, most of these systems are proprietary, and lack a comprehensive, community-accepted security analysis. For instance, Bytemark [2], developed an innovative mobile ticketing solution for transports, attractions, and events using a visual aid in fare control [8]. Contrarily to other companies, they do not offer a standard product, but yet a customizable solution, that range from trip planning, to securing tickets in a cloud system which may be shared with other users' devices.

A Spanish company Aditium launched an application called TickTrack [3], which is another mobile ticketing solution, but employs geolocation and other metrics to provide better security, in order to solve the problem of the device authentication.

3.2 Touchalytics: On the Applicability of Touchscreen Input as a Behavioural Biometric for Continuous Authentication

Touchalytics [7] is a system built to continuously authenticate users based on the way they use their smartphone screen. They accomplish this by extracting touch data, computing a set of 30 biometric features, and with the help of a classifier authenticate the user. Two types of classifiers were used, k-nearest-neighbors (kNN) [5] and a support-vector-machine with and rbf-kernel (SVM). This system is built on the premise that the data retrieved from the movements of the user on the touchscreen, is sufficiently different to serve as behavioural biometric. Their approach was able to achieve a Equal-Error-Rate (EER) of $<= 3\%$ by using a sample size of 13 strokes.

3.3 Touch me once and I know its you! Implicit Authentication based on Touch Screen Patterns

Touch me once and i know its you [6], was inspired by lock-screens in the Android mobile operating system. Using an implicit authentication approach and the user interaction on the smartphone touchscreen, the authors extract a set of behavioral biometric features capable of recognising different users. Their motivation comes from a security point of view, where lock-screens are very easy to bypass, for instance using Smudge-Attacks [2], or shoulder-surfing. In a first time study they reach a minimum accuracy of 38%, but after trying a different approach and simulating a real world usage, the values of the accuracy reached a minimum of 77%.

[2] https://www.bytemark.co/
[3] http://ticktrack.aditium.com/

3.4 Secure Unlocking of Mobile Touch Screen Devices by Simple Gestures —You can see it but you can not do it

In this paper a system called GEAT [1] is proposed. GEAT is a gesture based user authentication scheme for secure smartphone unlocking, implemented on a Samsung Focus smartphone, using Windows Mobile operating system. Instead of just using a typical unlock screen, where the user has to input a defined pattern to gain access to the smartphone, they add an extra layer of security that also checks how the user inputs this patterns. In brief GEAT offers an extra layer of security when unlocking smartphones by calculating biometric features capable of distinguish different users. They report an average equal error rate (EER) of 0.5% with 3 gestures and using 25 training samples.

4 Architecture

In this section there is presented a ticketing system used as a reference (section 4.1), following by a proposed solution (section 4.2) that aims at implementing HCE using NFC technology in the reference ticketing architecture.

4.1 Reference ticketing architecture

The proposed solution aims to fit on a standard ticketing system without significant changes. Implementing a completely new ticketing system incurs in enormous costs, which would prevent the uptake of the solution.

Figure 1 depicts the relations between the elements of the reference ticketing architecture. The Master Secure Access Module (SAM), is responsible for the generation and storage of the various systems keys [3]. The SAM only supports symmetric keys that, by definition, must be shared by every communication party, including the users, which are not completely trustworthy to not leak the keys. To prevent it, the SAM implements a set of security management measures, such as, restricting keys to specific applications.

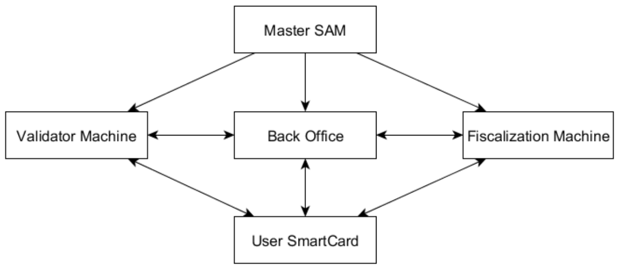


Fig. 1. Various elements involved in the ticketing architecture.

The ticketing architecture, just described, executes a wide variety of protocols. In Figure 2 is presented a succinct general architecture of a ticketing system and their operations.

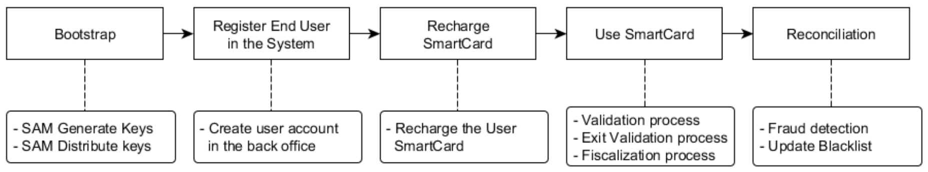


Fig. 2. Sequence of protocols run by the general ticketing architecture.

4.2 Proposed Solution

The cornerstone of our solution is a VirtualCard per client maintained by a central server in the cloud (CS). The main objective of the VirtualCard is to enable the secure usage of a mobile device in a ticketing system, which is achieved by a combination of a tokenization strategy and the maintenance of revision files per token.

1. **Tokenization:** The process by which value is transferred in different types of digital tokens which gives access to PT systems. Tokenization is used to create a low value token containing only the amount necessary to use the service on a defined period of time, therefore reducing the impact of a potential attack.
2. **Revision File:** Revision files contain historic information on each issued token. After being issued, tokens are never changed, although they may be partially consumed. A "revision file entry" not only consumes part of the token but also extends its validity. Revision files are created on every operation originated by a Validation/Fiscalization.

Figure 3 details the overview of the architecture used an the various components involved. The proposed solution has a group of protocols executed during the system life-cycle that assure the security and integration of the system with the reference ticketing architecture and the user mobile device.

4.2.1 Registration The registration protocol starts by establishing a secure communication channel with the CS, using Transport Layer Security (TLS) and requesting for a new registration. After receiving the registration request the CS asks the user to define a password and picks a random unused identifier from a sparse ID domain, to be used as a user ID. Both the chosen ID and a derivation (PBKDF2 [11]) of the chosen password are stored in an account for the user, together with a new VirtualCard. Finally, the CS sends a notification to the user informing him of his user ID.

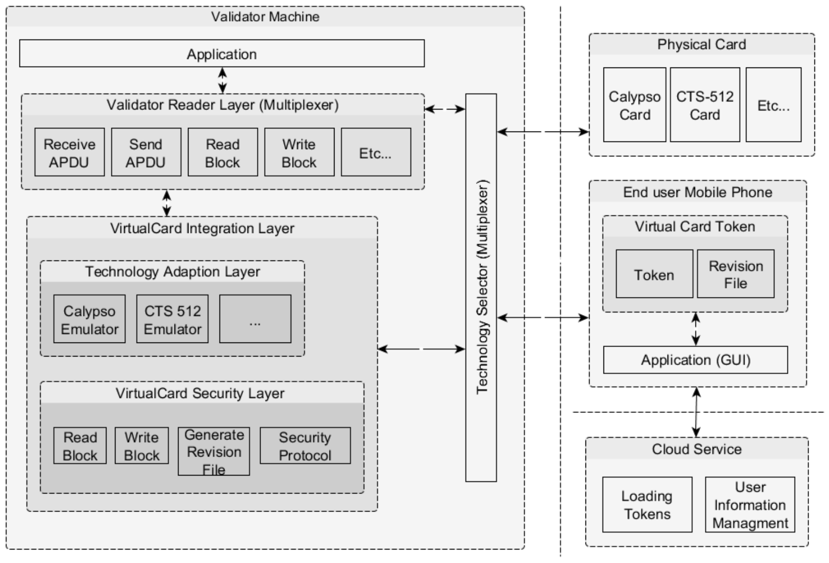


Fig. 3. Detailed overview of the scenario architecture with the VirtualCard Integration Layer.

4.2.2 Mutual authentication sub-protocol After the registration protocol, a mutual authentication is executed each time a consumer uses the CS. This protocol starts by establishing a TLS channel between the user's mobile phone and the CS. The user inputs his password on the mobile phone and submits it to the CS, along with the device ID and the user ID assigned by the CS in the registration protocol. On the CS side, the system applies a PBKDF2 to the user password, transforms it to the derived key, and compares it with the one stored in the database. If the key matches, the authentication is successfully completed. The main objective of this protocol is to validate the user identity through a mobile device. It must be noted that, given the modular nature of the proposed architecture, the mutual authentication protocol may be strengthened with additional authentication factors, for instance, non invasive authentication mechanism like Touchalytics [7] could be added without much visible changes.

4.2.3 Loading Tokens Loading tokens into the user mobile device starts by the user selecting on the application the transportation service he intends to use. Later the device requests a token for it and the user is required to authenticate himself (section 4.2.2).

Upon receiving the request, the CS loads the user VirtualCard and verify if it has enough balance to create the token requested and if the user is not blacklisted. If the verification checks, the token is generated, and two messages are generated and sent to the user's device, the $TokenDecryptMessage$ message and the $EncryptedTokenMessage$, containing the token encrypted and a necessary message for the decryption of the token. The $TokenDecryptMessage$ can only be decrypted by all the validators and by the CS, therefore it cannot be changed in transit or in the users device.

4.2.4 Redeem Tokens Redeeming a token requires the user to authenticate himself (section 4.2.2). After the connection is established the user device sends the $TokenDecryptMessage$ to the Validator, that decrypts, creates the $UserCheckMessage$, containing a random challenge, and sends it to the user device to verify if the user is legit.

When the user device receives the message, decrypts it and with the information gained in the message, decrypts the $EncryptedTokenMessage$. Finally the mobile device sends the $TokenMessage$, to the Validator, containing the encrypted token and a response to the previous challenge.

After receiving this message the Validator decrypts the token, checks the token integrity, then decrypts the challenge, and compares it with the one created previously. If the execution times are within an acceptable range, and the user challenge is correct, the Validator analyses the token validity period. After these verifications the Validator emulates a specific card technology. Upon emulation the Validator creates an updated version of the Revision File and sends it to the user mobile device.

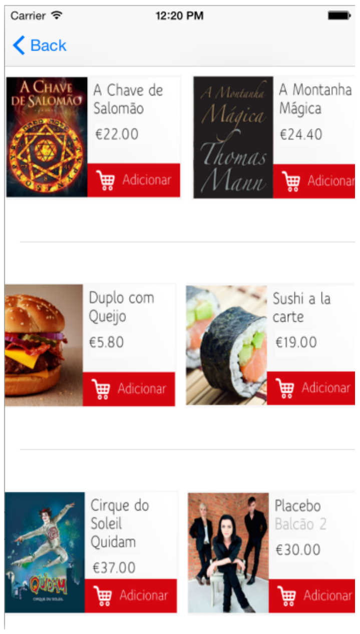
5 Implementation

In terms of implementation this thesis focused on augmenting the mutual authentication protocol (section 4.2.2) and Touchalytics meet our criteria for a solution that doesn't affect usability too much, or if possible at all. A solution that identifies the user in an invisible way, and its resistant to shoulder-surfing or smudge attacks. And doesn't need extra hardware in order to work.

Touchalytics implementation is divided into two sides:

– Server-side: Written in Node.js and responsible for exposing a web service via a JSON interface, handling the user connections, calculating all the biometric features, and storing the necessary data in a persistent way.
– Client-side: Mobile application for iOS, responsible for monitoring the user interaction, extracting the necessary raw data, position, time, pressure, area, and send it to the server-side in a JSON format.

Figure 4 and 5 exemplify our final iOS application capable of retrieving the swipes performed by the user and sending that information for further process on the server side.



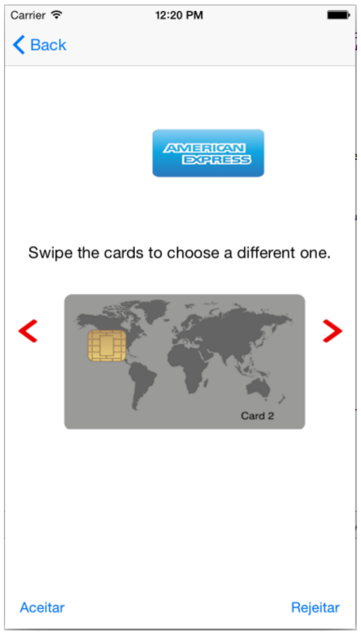


Fig. 4. Final application shopping screen.

Fig. 5. Final application card selection.

6 Evaluation & Results

The project evaluation was made in the terms of the following standard metrics, false positives (FP), false negatives (FN), true positives (TP), true negatives (TN), accuracy (1), and training delay.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \tag{1}$$

To obtain the values used in the evaluation metrics, we performed a series of tests:

- **Test 1:** The objective of this test is to simulate the legit user authentication by the system, which consisted in selecting a user along with 30 of his swipes, and save the other 10. Querying the database for the remaining 13 users and their swipes, calculate Touchalytics biometric features, train the SVM classifier and record the training time. After this the classifier passes from training mode to classification mode, and we can select the 10 swipes saved from the first user, feed it to the SVM classifier, and record the amount of TP, along with the FN.
- **Test 2:** The second test simulates a possible attacker trying to access the system by performing a certain number of swipes. The test begins by selecting a random user to be the attacker, and another user to be the victim. In

Training users	10
Training swipes	13
Classification swipes	8
Authentication Experiences	220
False negatives	15 (6.8%)
True positives	205 (93.2%)
Number of Attacks	220
True negatives	192 (87.3%)
False positives	28 (12.7%)
Accuracy	90.2%
Training delay	3311 (ms)

Table 1. Test results (10 users - 8 Classification Swipes).

similarity to the previous test, all the biometric features are calculated, and inserted into the classifier, in this case we leave the attacker out of the training phase. During the classification the swipes of the attacker are converted into the biometric features and inserted into the classifier. This tests enables the retrieval of TN and FP.

A key point that we miss on the tests performed is the fact that after some time people tend to change their their touch behavior, which affects the data collected. Taking this point into consideration we decided to divide the tests into two scenarios, a best case, where the classification and training data are collected on the same date, and a worst case, where the data is collected at different dates.

Table 1 depicts the results on a worst case scenario using a database with 10 users, 13 swipes per user during the training phase, and 8 swipes to classify a new user. The results show a high throughput in successful authentications (TP), 93.2%, and also a high success on preventing possible attacks (TN), 87.3%. In terms of accuracy the system achieved 90.2%.

7 Conclusions

In this thesis we presented a mobile ticketing solution using HCE technology combined with tokenization strategy and revision files to increase overall security in case of compromise. Additionally we designed and implemented a system capable of interacting with NFC4sure during the mutual-authentication protocol (section 4.2.2), with the objective of identifying and authenticate an user with a mobile device.

References

1. *MobiCom '13: Proceedings of the 19th Annual International Conference on Mobile Computing &#38; Networking*, New York, NY, USA, 2013. ACM. 533133.
2. Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies*, WOOT'10, pages 1–7, Berkeley, CA, USA, 2010. USENIX Association.
3. Calypso. Sam usage in a ticketing system. http://www.spirtech.com/phocadownload/category/13-sam-s1/5-sam-s1, 2014. Accessed: 2015-07-15.
4. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.
5. T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1):21–27, September 2006.
6. Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner, and Heinrich Hussmann. Touch me once and i know it's you!: Implicit authentication based on touch screen patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 987–996, New York, NY, USA, 2012. ACM.
7. Mario Frank, Eugene Ma Ralf Biedert, and Dawn Song Ivan Martinovic. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *CoRR*, abs/1207.6231, 2012.
8. Bytemark Inc. Method and system for distributing electronic tickets with visual display, granted patent us 8494967 b2, 2012.
9. Markus Jakobsson, Elaine Shi, Philippe Golle, and Richard Chow. Implicit authentication for mobile devices. In *Proceedings of the 4th USENIX Conference on Hot Topics in Security*, HotSec'09, pages 9–9, Berkeley, CA, USA, 2009. USENIX Association.
10. Vojislav Kecman. *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. MIT Press, Cambridge, MA, USA, 2001.
11. Colin Percival. Stronger key derivation via sequential memory-hard function, 2009.
12. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing - Section 16, "Support Vector Machines"*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.