Dynamic QR codes for Ticketing Systems

José Miguel Oliveira
jose.f.oliveira@tecnico.ulisboa.pt
Instituto Superior Técnico
Portugal

ABSTRACT

Mobile phones are invaluable systems of our society. With the increasing growth in use of mobile phones, emerging technologies are constantly being introduced and the development of technologies, such as barcodes, contributed to the rapid widespread of mobile phones over various systems, including ticketing systems. Mobile ticketing is a subset of ticketing systems and it provides a simple alternative to the use of physical tickets. Mobile ticketing makes use of mobile tickets which are digital tickets in smartphones and they have the same functionalities as their predecessors such as tickets in smartcards.

By taking advantage of these technologies, it is proposed a ticketing system which makes use of tickets as QR codes. The proposed system contains a mobile application which can be used to purchase and present tickets, in the form of QR codes. Tickets are purchased from the backend server and validated in ticket validators. Ticket validators perform the offline validation of tickets and therefore, in instances where ticket validators do not have access to the internet, they do not lose functionality and can continue performing the correct validation of tickets.

1 INTRODUCTION

Nowadays mobile phones are invaluable systems of our society. A rapid development of the internet in recent years has allowed mobile phones to be used more than any other form of device and, the development of technologies in mobile phones has contributed to the growth of use and reachability of barcodes such as QR codes, since they can be processed not only by barcode readers but also mobile phones.

Barcodes are methods that allow for the access to services. Since barcodes provide a uni-directional method of communication, very efficiently and free of charge, barcodes are used in the most varied of applications. Taking this into account, by being able to facilitate a payment in order to obtain a service, the intention is to extend this method to public transport systems, which are one of the many applications of ticketing systems. Mobile ticketing is a subset of ticketing systems that aims to provide an easy alternative to the most currently used ticketing systems which, as seen in the public transport industry, require a lot of physical interaction. Mobile ticketing systems provide the ability of using tickets within the mobile phone application, with a view to process the obtaining and validation of tickets. When trying to display a ticket for the validation process, the mobile application creates a mobile ticket, in the form of a QR code.

In the public transport industry, the validation of tickets is done offline since data connections are often unreliable and in order to prevent frustration when internet is not available. Besides that, the security of tickets can be achieved using bi-directional communication such as the case of smartcards which does not work for static barcodes since barcodes do not receive information.

However, there is a problem with the usage of static QR codes for the offline validation of QR codes. Since QR codes do not receive information, having a static QR code leads to the problem of not updating the contents of the QR code and thus, the security behind tickets could be compromised. Cloning attacks are attacks that work by stealing information such as, in this system, the stealing of QR codes. In order to prevent against these attacks, the mobile application makes use of technologies such as one-time passwords. With this, the mobile application implements a system with dynamic QR codes, which protects this system against the most commonly used attacks against mobile ticketing.

Applications of this system include systems which are based on tickets such as public transports, with a view to providing the ability to ride public transports, by using a mobile phone, and without the need of a public transport card.

One of the main objectives with the implementation of this system is the prevention of fraud. It is mandatory that the system is able to accept tickets that are valid and does not accept expired tickets or tickets that have already been used. Any other situations must be rejected as the system must not lead to a monetary loss neither for clients nor the companies that provide and manage the ticketing systems. Another main objective with the implementation of this system is availability. Users should be able to use the system and perform ticket validation without the need for having an active internet connection, in the event that a certain location has limited or there is a failure with the network access. Since situations like these could lead to frustration for the users of the ticketing system, the mobile application should be able to display tickets without internet connection, even if only for a brief period of time.

This paper is organized into five sections. Section 2 introduces the work related with mobile ticketing and applications on public transports. Section 3 presents the proposed methodology, including use cases of the system and technologies used for the prevention of fraud. Section 4 contains the obtained results and discussion over these.

2 BACKGROUND

There have been numerous advancements in the many forms of payment and unsurprisingly, the trend towards electronic cashless commerce is growing worldwide. Ticketing systems contain payment methods in the form of tickets and there have been implemented systems which take advantage of several technologies such as NFC, RFID, BLE and QR codes.

Ticketing systems are systems that are comprised of exchange of tickets in order to process payments to services. Industries that have a large amount of users, using the system at the same instant, often

take advantage of ticketing systems as the use of tickets facilitates the usability of the service and verification of the users who are using and have paid for the service. With this, ticketing systems are most commonly seen in the public transport industry.

Public transports make use of tickets that are purchased and validated in order to be able to use a desired method of transportation. These tickets are physical tickets emitted by an office worker or ticket vending machines, usually located at the entrance of a public transport and each ticket has attributes related to the service provided by the public transport. Even though traditional tickets are still widely used worldwide, these types of tickets are falling in decay as their use is the least convenient among other types of tickets(Mezghani) such as tickets in smart cards or mobile ticketing.

Belani et al. (Škarica et al., 2009) propose two different approaches for a mobile ticketing system. One implementation, which is an online validation system, takes advantage of MMS technology in order to send mobile tickets to clients, and clients validate the mobile ticket which is in the form of a data-matrix, in a 2D Image Scanner. Alternatively, it is proposed an offline validation system, by using QR codes. A ticket, which contains user information and other details, is processed in order to generate a hash value, from the concatenation of the ticket identifier and the digital signature of the ticket identifier. Since there is not an established connection between the mobile phone and the validation system, the content of the ticket is encrypted using asymmetric-key cryptography. The resultant hash value is stored inside a QR code which, after being read by a web camera, is validated in the validation system.

3 METHODOLOGY

The main goal of this project is to develop a mobile ticketing system that allows for the offline validation of tickets, in the form of QR codes, by taking advantage of the technologies available in mobile phones. With using technologies such as one-time passwords, this system contains a method to regularly update QR codes, known as the dynamic generation of QR codes, that are to be presented to a ticket validator.

In the case of this system, ticket validators are not guaranteed to have full internet availability and therefore, the implementation of online communication with near offline ticket validators would be infeasible, since ticket validators would not be able to send ticket validations, in real time, to a backend server which would then be responsible for notifying the mobile application that the user presented a certain ticket to a ticket validator.

Despite working in a near offline state, ticket validators should contact the backend server regularly in order to obtain new information and send the validations of tickets, which they recorded. One of the most innovative points of this system is that, even though ticket validators perform the validation of tickets offline, they are able to remain usable for various days in case they face problems contacting the backend server. Taking into account that in a real environment such as public transports, these devices require very low maintenance, failures in communication between ticket validators and the backend servers, if not taken in consideration, could be detrimental to the correct functioning of this system.

The proposed ticketing system implements a check-in only approach so that users are only required to confirm their entrance to the service, by validating their tickets at the entrance of the public transport. At the exit of the service, users can check-out of the system by exiting the service, whenever the user considers the service finished.

3.1 Use Cases

The proposed solution is a mobile ticketing system based on the dynamic generation of QR codes through the use of a mobile application, ticket validators and backend servers. The process of using this system can be divided in two use cases: the first use case is related to the process of acquiring tickets and the second use case is related to the process of validation of tickets.

3.1.1 Ticket Acquisition. The figure 1 contains a diagram with the comprehensive representation of the method of purchasing tickets. To start off, users need to authenticate in the application with an individual user account. Hence, when using the application for the first time, users are required to create a new user account. After creating their account, users are able to sign in the application and use the system.

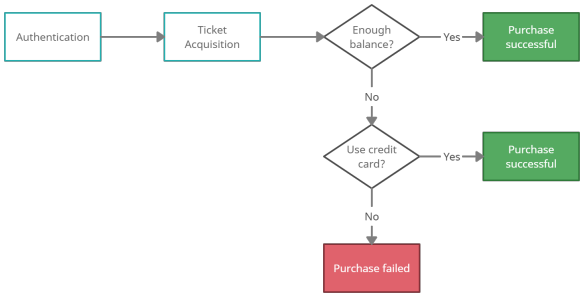

Figure 1: Ticket Acquisition Diagram

Following the authentication step, users can proceed to acquire tickets. After requesting the list of available tickets from the ticketing server, the mobile application displays the catalog of products for users to choose.

3.1.2 Ticket Validation. In the mobile application, users can select the ticket which they intend to use and the mobile application generates a dynamic QR code, as it can be seen in the figure 2. This QR code can then be presented to a ticket validator, in order to gain access to a specific service.

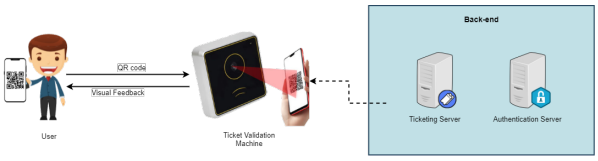

Figure 2: Ticket Validation Diagram

After scanning the QR code, ticket validators deliver a response to the user, in the form of visual feedback. To complete the process of validating tickets, ticket validators record a ticket validation,

which is going to be sent to ticketing server, at a later time. The ticketing and authentication servers are located in the backend and therefore, users does not have contact with either server.

As aforementioned, in order to exit the system, users can simply leave the service and the journey is considered complete.

3.2 Security and Fraud Prevention

With the development of an innovative ticketing system, which makes use of the offline validation of tickets, it is of utmost importance that this system can prevent attackers from committing fraud. Some users with malicious intents might attempt to attack this system, by exploiting its vulnerabilities, to gain advantage such as making use of the ticketing system whilst not paying for tickets.

Therefore, the problems which were needed to be taken into consideration when developing this ticketing system: the verification of authenticity of users and the prevention against cloning and double spending attacks.

Starting off with cloning and double spending attacks, attackers attempt to commit fraud by using tickets from other users and therefore, avoid paying for tickets. It is of utmost importance that this system can be resilient to these attacks, in order to prove its reliability.

The problem of the authenticity verification lies in the fact that the communication between the mobile application and ticket validators is done offline. As previously stated, a user presents a QR code to a ticket validator which performs the validation of ticket and emits visual feedback to the user. In order to assert authenticity and to prevent tickets from being either forged or used by other entities, this system implements two security methods in every QR code, with the use of digital signatures and one-time passwords.

3.2.1 Digital Signatures. The use of digital signatures allows for the prevention against the forgery of tickets and prevent any modifications in the contents of tickets. Each ticket contains a signature field which is verified by ticket validators, when validating tickets, and tickets whose signature does not match the signature in the ticket are rejected. In addition, every signature is accompanied with an expiration date. By setting an expiration date of the signature, in the ticket, users are required to request for a renewal of the signature regularly, and this method was implemented with a view to assert the authenticity of users on a regular basis.

3.2.2 One-time Passwords. Each QR code contains a one-time password, namely time-based one-time passwords. Time-based one-time passwords are used to create a QR code that can only be used once and for a very limited duration. Therefore, the use of a one-time password contributes to the prevention against cloning attacks.

In this system, cloning attacks are attacks where the attacker is able to steal the ticket from another user, by taking a photo of the QR code. With the use of one-time passwords, this system can prevent attackers from using the QR code after it has already been used and even if not used, since one-time passwords have a very limited period of validity, the one-time password may expire before the attacker is able to use the QR code. In the former scenario, the ticket validator rejects the ticket since it contains a history of ticket validations to prevent the use of the same QR code more than once.

The use of one-time passwords led to the implementation of a system with the dynamic generation of QR codes in the mobile application. This is done to avoid displaying invalid QR codes and, upon the expiration of the OTP value inside the QR code, the mobile application automatically updates the QR code.

3.2.3 Validation History. Since ticket validators perform the offline validation of tickets, they are required to maintain a history of the validation of tickets so that they can verify the usage of tickets in multiple instances, and reject tickets that are used more than once.

Using public transports as an example, the implementation of the validation history may differ depending on the vehicle. In case there is a single validator such as at the entrance of buses, the ticket validator is responsible for the validation of every ticket since all passengers must present their tickets to that ticket validator. In case there are multiple ticket validators such as at the entrance of metro stations, the station must contain a physical centralized server, which ticket validators are connected to, in order to share the validation of tickets across all ticket validators. This server may operate offline and its purpose is to keep the centralized registry of ticket validations and allow other ticket validators, located inside the metro station, to reject QR codes which were scanned in a different ticket validator.

3.2.4 Asymmetric Keys. This system uses asymmetric keys due to the fact that QR codes contain sensitive information and, to provide resilience against reverse engineering attacks to the QR code. The mobile application receives a public key from the ticketing server and uses it to cipher the ticket and OTP value, and its result is the message stored inside QR codes. A ticket validator receives the corresponding private key, also from the ticketing server, and can use it to decipher the QR code and validate its contents.

3.2.5 Types of Tickets. Due to the fact that the validation of tickets is performed offline and that QR codes cannot receive information, ticket validators are not able to notify the mobile application that a certain ticket has already been used. Besides that, the mobile application as well as the rest of the components of this system, implies a zero-trust policy where this system does not allow users to execute specific privileged actions and thus, not giving users the ability to commit fraud against this system.

Taking this into account, the types of tickets used by this system are time-based tickets which are tickets that can be used multiple times until the ticket becomes expired. Through the synchronization of clocks and the aforementioned security methods, this system can ensure that tickets are only usable for the specified time.

3.3 Offline use

Since the main goal of the proposed system is that the system can also be used offline, there are certain aspects of this system that were modified in order to support the validation of tickets without internet connection. The two most prominent scenarios where these aspects are important for the offline use of this system are the synchronization of clocks and the rotation of the encryption key.

3.3.1 Time Synchronization. The synchronization of clocks to the time server requires active internet connection. However, in order to

be able to achieve synchronicity in times within the entities of this system, when offline, the mobile application and ticket validators use the Network Time Protocol (NTP). With this, these entities can calculate the offset of their internal time clock to the time clock in the time server. The offset value is calculated to obtain the time difference from these entities to the time server, so that the synchronization of clocks can be achieved.

This value is then stored in memory and when there is the lack of internet connection, the mobile application and ticket validators use the offset to become synchronized with the system. Ticket validators work offline for large periods of time and thus, it is required to make use of an offset value for the correct validation of tickets. Furthermore, in the mobile application, the offset value is essential in instances where the the mobile application fails to connect to the internet, to obtain the synchronized time.

Without the calculation of the offset value, the mobile application would be required to be connected to the internet, at all times, which goes against one of the objectives of this system. The mobile application should be able to generate QR codes, without internet connection, even if only for a short period of time.

3.3.2 Key Rotation. Encryption keys are changed regularly to prevent brute-force attacks. As previously mentioned, since ticket validators require very low maintenance, in the event that ticket validators fail to obtain a new private key, used to decipher the contents of the QR code, they cannot properly validate tickets.

With this, instead of receiving a single private key, each ticket validator receives a batch of private keys from the ticketing server. The ticketing server is responsible for the management of the batch of keys and for securely distributing the batch of keys to ticket validators.

3.4 System Architecture

In the figure below 3, there is a representation of the system architecture which provides a comprehensive diagram of the relationship between every component of the system.

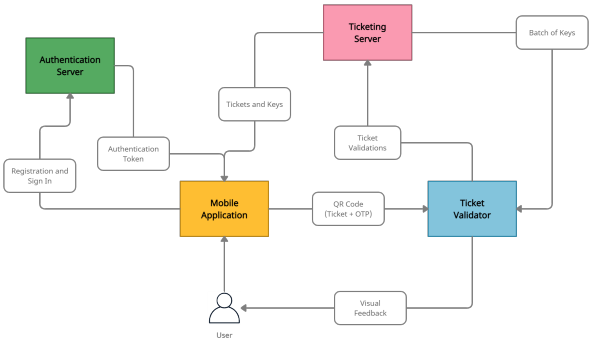

Figure 3: Architecture

The mobile application is the only user side entity which allows the user to interact with the system. It contains a login system as well as a dynamic QR code generation system which allows the user to present the QR code to a ticket validator.

As stated above, ticket validators are set up outside the application and operate offline to ensure there are no failures in the communication between ticket validators and the mobile application, and that internet availability is not a limiting factor for the validation of tickets.

Besides that, the mobile application and ticket validators interact with the backend servers, which are the authentication and ticketing servers. The authentication server is used on authentication in the mobile application and ticket validators, and the ticketing server is responsible for the purchase of tickets and the rotation of keys.

The architecture of the system is divided in three phases which are the login system, generation of the QR code and the validation of the QR code.

3.4.1 Login and Registration. The login system found in the mobile application allows users to register or authenticate in the system, through the authentication on the authentication server. The authentication server is an OpenID Connect server which emits an authentication token which consists of an access token that is going to be used on the ticketing server, for instance, to authorize the user when trying to purchase a new ticket.

At the moment of login, the mobile application is required to be connected to the internet and therefore, the mobile application obtains a public key from the ticketing server. This key is used to encrypt the contents of QR codes and the rotation of this key is done on a regular basis, to prevent brute-force attacks against the key.

The ticketing server is responsible for the generation of these encryption keys and, in order to allow ticket validators to read the contents of QR codes which are ciphered with the public key sent to the mobile application, the ticketing server sends the corresponding private key to ticket validators. With this, ticket validators are the only entities able to decrypt and thus, validate the QR codes generated in the mobile application.

3.4.2 QR code Generation. Upon entering their credentials and signing in the mobile application, users are presented with the home page which contains tickets that the user has purchased. On the first run of the mobile application, the home page does not contain any tickets and, in order to obtain tickets, users are required to purchase tickets from the ticketing server. Upon purchasing a new ticket, the mobile application redirects the user to the home page.

By clicking on the ticket, the mobile application displays the purchased ticket in the form of a QR code, so that the user can present the QR code to the ticket validator.

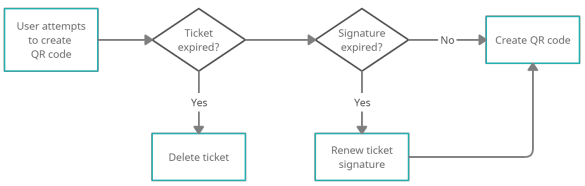

Figure 4: QR code Generation

From the figure 4, the process of generating a QR code starts by ensuring that the ticket does not contain expired parameters such as the expiration date and the expiration date of the signature. Upon the expiration of the signature in ticket, the mobile application is able to request for a renewal of the ticket signature from the ticketing server. Eventually, the mobile application proceeds to generate a time-based one-time password (TOTP) based on the ticket and a timestamp of the current time. The timestamp is obtained from a time server, to ensure that there is the synchronization of clocks across the system. Finally, the mobile application displays a QR code which consists of the encryption of the ticket and one-time password.

3.4.3 QR code Validation. The ticketing server is responsible for the emission of the batch of private keys to ticket validators, and this process is required to be a secure process to avoid the stealing of the batch of keys and prevent fraud. Users that get access to the batch of keys, are able to gain control over the system and it is of utmost importance that the batch of keys cannot be accessed by any entities, with the exception of ticket validators. Taking this into account, the ticketing server would supposedly send the batch of keys through a secure network and requests for the batch of keys can only be accepted for entities which are able to obtain the access token of a ticket validator.

As previously mentioned, in cases where ticket validators fail to obtain a new batch of keys, they are prepared to switch to the next private key in the batch of keys, which is synchronized with the corresponding public key, sent to the mobile application. The batch of keys is securely stored in the ticket validator and it is not shared with any other entities.

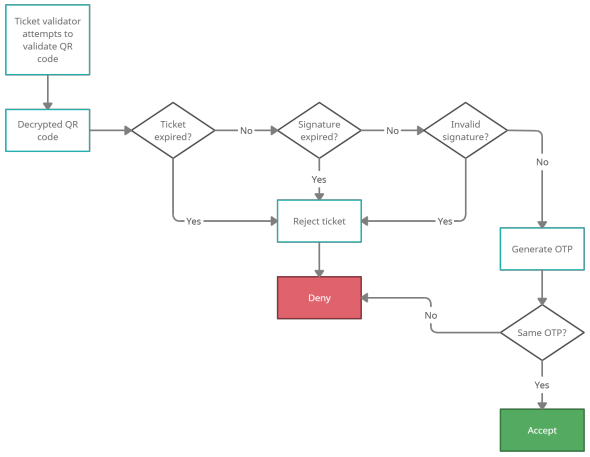

Figure 5: QR code Validation

The process of ticket validation starts by having the ticket validator decrypt the encrypted QR code, using the private key that was received from the ticketing server. Finally, ticket validators provide visual feedback to the user based on the validation of tickets and the result of the comparison of the OTP in the QR code and the OTP calculated by the ticket validator. After this, the ticket validator records a ticket transaction which is a registry with the information of a ticket along with the timestamp of the validation time.

The recording of tickets is done with a view to later sending these ticket transactions to the ticketing server. Since ticket validators do not have constant access to the internet, ticket transactions are stored in the memory of the ticket validator, and ticket validators attempt to send these tickets transactions, on a regular basis, to the ticketing server. The primary purpose of sending ticket transactions to the ticketing server is to allow for a centralized registry of the transactions made in the system. With this, the system allows for revenue sharing and it is able to gain insights over how users make use of the system and allow for improvements. In addition, the ticketing server contains an anti-fraud system which, by analyzing ticket transactions, this system can detect suspicious user activity and take preventive action.

4 RESULTS & DISCUSSION

Following the implementation of the system, the various components that make up the system were subject to testing, namely measuring times as to obtain performance metrics over important services provided by each component such as the time to create a QR code or the time to validate QR codes. This section contains the metrics collected through the use of the system and contains a deliberation over the results obtained.

Notice that these values were obtained in a simulation environment and not in a real public transport vehicle or station. Only by testing in a real environment, with the interference of vehicles or temporary losses of internet connection, could the results obtained have been more precise. In a scenario where there are no slowdowns in internet connection, the results obtained are approximate to the real-time operation of this system.

The simulation scenario used for the evaluation of this system was a scenario similar to the scenario found in the ticket validation use case figure 2. As stated above, the user of this system uses the mobile application on a physical device, namely a mobile phone. A ticket validator is located in front of the user so that the ticket validator is able to scan a QR code with the ticket, shown by the user. After reading the QR code, the ticket validator emits visual feedback based on the validation of the ticket. The back-end servers are located in the backoffice and, in order to simulate the use of back-end servers, these servers were located in a separate room, different from the one where the user was showing tickets to the ticket validator.

As previously mentioned, there are two different version of the mobile application: one for Android devices and the other version for iOS devices. For the Android version, the devices used for testing were OnePlus 7T Pro and Samsung S7 Edge whereas for the iOS version, the mobile phone device was iPhone 7 Plus. In addition, ticket validators were created on a computing platform consisting of an Intel i7-3520M (3.6GHz) with 16GB of RAM. Furthermore, the back-end servers were created on a computing platform consisting of an Intel i5-6200U (2.8GHz) with 8GB of RAM.

In order to obtain the test results, each component tested records the timestamp of the time prior to executing the service as well as the timestamp of the time after the execution, and creates a log file

with these times and the difference between both times, and stores these metrics.

4.1 Performance Results

This section contains the results from the various tests made to the mobile application and ticket validator. Besides containing a detailed discussion of the obtained results, this section also provides insights on situations where these values can be different from real-world results.

4.1.1 Creation of QR codes. Starting off with the ticket validation use case 3.1.2, a user has already purchased a ticket and intends to display the ticket to a ticket validator.

By evaluating the performance of the mobile application, in different mobile phones, we can gain insights over time amount of time a mobile phone device takes to cipher the message and evaluate whether or not is it appropriate for the dynamic generation of QR codes.

Since older mobile phones may lack the processing capabilities of newer mobile phones, we obtained results across multiple mobile phones with different operating systems, and even with different OS versions, so that we can evaluate the difference in performance and study the impact of using mobile phones with different capabilities, on this system. These results are also taken into account to ensure that users are less likely to be limited to using newer devices, when using this system.

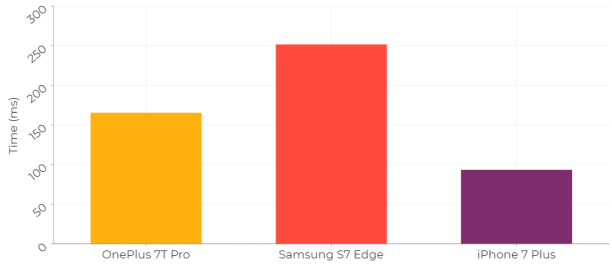

Figure 6: Creation of QR codes

The figure above 6 contains the results of the average time to perform the creation of QR codes, over 150 runs. As stated above, in order to obtain these results, the mobile application obtains the timestamp before creating a QR code and calculates the difference to the timestamp obtained when the QR code is set to display on screen. Besides that, the mobile application calculates the difference of these times and stores it in a file.

In a real environment, the results obtained would be similar to the results in the figure, since the mobile application was installed in everyday devices and thus, they are not specific to this test case.

From the results obtained, it is possible to infer that the time taken in the creation of QR codes is acceptable and also suitable for the dynamic approach to QR codes, since users are not required to wait for the process of generating QR codes as it occurs almost instantly after selecting the desired ticket to present as a QR code. Additionally, older devices such as Samsung S7 Edge, do not fall behind in comparison with newer models such as the OnePlus 7T Pro.

4.1.2 Validation of QR codes. Following the creation of QR codes in the mobile phone application, a ticket validator must ensure the correct validation of tickets, through the scan of QR codes. This section presents the evaluation over the validation of QR codes and the comparison between the usage of two different hybrid encryption mechanisms, and how these results impact validation machines.

The process of validation of QR codes starts with the decryption of the message in the QR code and, since this could be the most intensive step in the validation of QR codes, this test case is also focused on various methods of decryption of the message. Furthermore, the focus on the validation of tickets is done to ensure that ticket validators do not take up much time validating tickets and execute the validation of tickets within the time limit of 1 second.

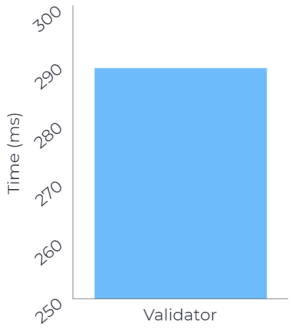

Figure 7: Performance results from the validation of QR codes bar chart

The figure 7 contains the average time that the ticket validator took, when executing 175 runs in a computing platform, to validate each QR code that was presented to the reader. Since it was not possible to test and obtain performance results from a ticket validator machine that is used in a real environment, these results do not represent the results that would be obtained in a real environment, since ticket validator machines have much smaller processing capabilities, in comparison with the computer used to obtain these results.

By taking a look at the results in the figure, the ticket validator performs the validation of tickets with an average time of 289.4ms. Despite the fact that ticket validator machines in public transports have smaller processing capabilities than the machine used to obtain these results, the implementation of this system with the former ticket validators should not be too time consuming and cause users to find the process of validating tickets, inconvenient.

Comparison between ECIES with RSA-AES. In addition, it was tested the use of alternative hybrid encryption methods such as RSA-AES and these results were compared with the results obtained from using ECIES. In the figure 8, it is presented a graph with the average time across 75 runs, of the process of decryption of the message in the QR code. This is solely for the process of decryption of the message, and these times do not include the time from other methods for validating the ticket as well as emitting visual feedback to the user.

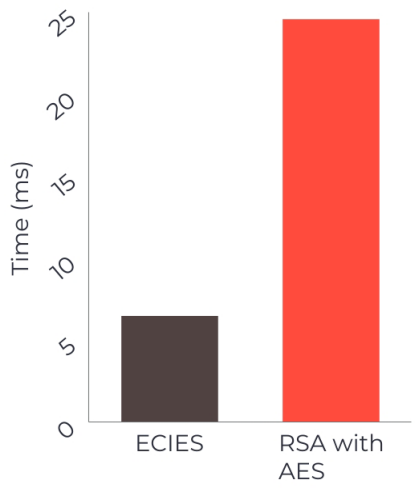

Figure 8: Comparison between RSA+AES and ECIES

From the figure, we can see that the implementation of ECIES finishes the process of decryption much quicker, with an average time of deciphering of less than 7ms whereas the implementation of RSA with AES has an average time of over 24ms. Despite the fact that using RSA with AES is not as fast as using ECIES, these results indicate that either approach would be suitable for the current security levels, since the validation times are not very significant when used in the current prototype of the ticket validator. However, in a real environment, due to the fact that ticket validator machines do not have the same processing capabilities as the computing machine used to represent the ticket validator, the impact of using RSA with AES instead of ECIES could become more noticeable and therefore, the use of ECIES would be more suitable for the validation of tickets.

The process of hybrid encryption using ECIES consists of using ECC with symmetric encryption, namely EC keys generated from the curve NIST P-256 combined with AES-128 symmetric keys. In order to be able to perform a fair comparison, the implementation of RSA with AES provides similar security levels to the ones found in ECIES. When using a combination of RSA with AES, the approach used RSA 3072-bit keys with AES-128 since RSA 3072-bit keys have the same level of security as ECC P-256 keys.

In the event that this system required higher security levels such as using EC keys from NIST P-512 for 256-bits of security, instead of having 128-bits of security from using ECIES, with EC keys from NIST P-256, the average time of using the implementation with RSA with AES could rise exponentially and cause a major impact in the overall performance of ticket validators (Gayoso Martínez et al., 2015).

4.2 Battery usage

Nowadays, it is important that mobile phones are able to remain usable for extensive periods of time, namely for the duration of a whole day. This section details the battery consumption over time of the mobile phones and an explanation of how these results could diverge from usages in a real environment.

Since the generation of QR codes in the mobile application is dynamic, the mobile application creates and displays a new QR code every 60 seconds and as such, it is considered to be the most expensive action in regards to battery usage. For the current use case of the application, users would use the mobile application for a few minutes and close the application as the main purpose of the mobile application is allow users to present a ticket from the mobile application to the ticket validator.

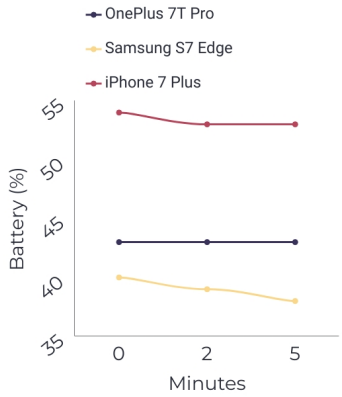

Figure 9: Battery test results

The figure 9 contains a graph with the variation of battery over time, in different mobile phone devices, that were tested using the mobile application, performing the dynamic generation of QR codes, in order to put stress over the mobile phone and maximize its battery consumption. These results were taken using a refresh rate of 60Hz and without the use of the power saving mode across all devices.

From the results in the figure, presenting the QR code over a short period of time such as 2 minutes, may not even affect the battery percentage as seen in the OnePlus 7T Pro. Despite this, the mobile application does not lead to draining huge percentages of battery resources, with an average of less than 1%. Therefore, these values are acceptable for real environment usage since these values do not cause almost any impact to the battery percentage of the mobile phone.

An exception to the rule is the Samsung S7 Edge which has a battery with years of extensive use and charges, and since the health of its battery is really low, the results on the Samsung S7 Edge may diverge from other devices of the same model.

Alternatively, it would be possible to test the battery consumption of the mobile application in case it was left open, for instance, on the home page, for a long period of time such as 1-2 hours. It would provide insights on the different processes that run in the background of the mobile application impact the battery performance of the mobile phone.

4.3 Security Issues

Besides obtaining performance metrics of the usability of the components of this system, specifically the mobile application and ticket validator, it is necessary to take into consideration that this system can be susceptible to attacks.

As previously mentioned, this system could be implemented in a public transports system and thus, some users with malicious intents could attempt to circumvent the security measures imposed for the correct functioning of this system, in order to avoid paying for tickets and commit fraud. In addition, other types of fraudulent behaviour may include the attempt of taking advantage of inexperienced users, with techniques such as social engineering or phishing.

This section details the possible attacks and methods that attackers can attempt to exploit the vulnerabilities of this system, and countermeasures implemented to protect this system against those attacks and prevent fraud. Taking this into account, the following attacks are the attacks that are most likely to be used against this system:

- Reverse Engineering
- Cloning
- Double Spending
- Phishing

4.3.1 Reverse Engineering. A reverse engineering attack consists of opening up an object to study its mechanisms. This system contains two methods to prevent reverse engineering attacks: code obfuscation of the application code and the encryption of the contents of QR codes.

Code obfuscation. In this system, attackers may attempt to reverse engineer the application code with a view to extracting valuable information on how the application works, and to do so, attackers have at their disposal several mechanisms such as IDA Pro, which allow to convert the binary application code, to source code. The intention behind dissecting this application could be to gain insights over the creation of QR codes such as which algorithms are being used or even, the methods of storing tickets, to facilitate access to tickets.

In order to protect against these types of reverse engineering attacks, the obfuscation of the application code reduces the possibility of execution of this attack since its purpose is to difficult the process of extraction of valuable information from the mobile application.

The Android version of the mobile application uses ProGuard which is a Java open-source cross-platform tool whereas the iOS version of the mobile application contains a simplistic approach to code obfuscation where sensitive data is stored encrypted, beforehand, and decrypted at the moment of use. This approach is not fully secure and only protects sensitive fields while ProGuard can also provide obfuscation to class names and methods. A possible implementation of code obfuscation in the iOS version of the mobile application, which would suit our needs better, is using iXGuard.

Encrypted QR codes. Alternatively, this system can be subject to reverse engineering attacks to the QR code. Since tickets are displayed in the form of QR codes, attackers may attempt to gain insights over the creation of tickets and find vulnerabilities in the methods of creating tickets.

Hence, in order to prevent this specific case, the mobile application implements the encryption of the contents of QR codes so that only authorized entities, namely ticket validators, can be able to read the QR code. In addition, the rotation of the encryption key is done on a regular basis, to prevent brute-force attacks against the key and consequently, make reverse engineering attacks more difficult to perform.

4.3.2 Cloning. Cloning attacks are attacks where an attacker steals information from another user, with the intention of using this information for their benefit and, in the case of this system, cloning attacks could be performed by copying or taking a photo of the QR code from another user. The window of attack is very short-lived since this attack could only be executed within the timestep or period of validity of the TOTP, which, as aforementioned, usually lasts less than one minute.

Despite this, cloning attacks can not be fully prevented with the use of one-time passwords due to the fact that attackers could still make multiple usages of the stolen QR code, as long as it had not yet expired. In order to prevent this, ticket validators are required to have a validation history so that the ticket validator can detect and reject QR codes that had been previously used. Time-based OTPs remain unchanged for the duration of the timestep and therefore, ticket validators are able to detect different uses of the same QR code.

As previously mentioned, ticket validators could be connected through a physical centralized server and therefore, these ticket validators would have access to validations which were performed by other ticket validators. With this, even though ticket validators operate in a near offline state, they would be able to reject tickets which had been previously accepted, in a different ticket validator.

4.3.3 Double Spending. A double spending attack is an attack which can be executed by reproducing the ticket and having two users, using the same ticket. To execute this attack, an attacker is required to obtain a ticket from another user, and both users could be cooperating to commit fraud.

There are multiple ways of executing double spending attacks such as mobile phone sharing, the sharing of QR codes over the internet and the extraction of tickets from the mobile application. This section also covers the reasoning behind the methods used to prevent these attacks.

Mobile phone sharing. To start off, one user could share their mobile phone with another user, after validating their QR code, and the other user can use the same ticket, as long as the mobile application has created a new QR code. This is due to the fact that ticket validators create a record of the ticket information and OTP value. Since this system makes use of time-based tickets, the owner of the ticket should be able to use the ticket more than once. Hence, ticket validators do not prohibit multiple uses of the same ticket but instead, they prohibit multiple uses of the same QR code.

For simplicity reasons, the current prototype of this system assumes that the sharing of the mobile phone would not be possible and therefore, the current implementation of this system does not prohibit against this specific scenario. A possible implementation

of this feature could make use of an anti-passback control system, which would prevent subsequent usages of the same ticket.

In addition, the ticketing server contains an anti-fraud system which detects fraudulent behaviour by analysing the ticket transactions sent from the ticket validators. Users that bypass the proposed anti-passback control mechanism, are going to be detected by the anti-fraud system which takes preventive action.

Sharing QR codes over the internet. Another case of double spending is the sharing of QR codes over the internet. For instance, one user could take a screenshot of a QR code, created in their mobile application, and send it to another user so that they would use the same QR code, in a different ticket validator. This scenario can only be possible due to the fact that ticket validators operate in a near offline state and do not receive information from other ticket validators, or a backend server, that a specific ticket has already been used.

In order to prevent users from sharing sensitive information, the mobile application implements security methods such as screenshot blocking. By blocking the possibility of taking a screenshot of the QR code, users are less likely to attempt the execution of this attack. The Android version of the mobile application contains the implementation of this method since it is supported natively by Android. However, as aforementioned, the iOS version of the mobile application does not implement methods to block screenshots since these are not supported natively by iOS.

Hence, the current prototype of the iOS version of the mobile application, contains a small window of attack which is equivalent to the period of validity of the OTP in the QR code. In addition, tickets contain specific parameters that can be used to limit the target audience of the ticket such as valid zones and valid operators. Tickets that are not presented to ticket validators that meet these requirements, are going to be rejected. With this, in the current prototype of iOS version of the mobile application, the combination of these factors does not fully prevent this specific case of double spending attacks. However, it makes the attack more difficult and reduces the possibility of successfully executing this attack.

Another possible implementation of screenshot blocking in the iOS version of the mobile application is through the use of ScreenShieldKit, which blocks sensitive content in screenshots.

Similarly to mobile phone sharing, users that bypass this security mechanism are going to be detected by the anti-fraud system which takes preventive action against fraudulent behaviour.

Ticket extraction. Users may attempt to forcefully move tickets from one application to another, in a different mobile phone. In case tickets were stored as plaintext, other applications could read the contents of tickets and generate new QR codes. Hence, the mobile application performs the storage of the encrypted contents of tickets, in the internal storage, of a mobile phone and therefore, these tickets can only be decrypted using the keystore from that specific device. Other devices are not able to read the contents of the ticket and thus, they cannot create new QR codes.

Besides that, in order to move tickets from one application to another, the attacker must bypass the security mechanisms imposed by the Android OS, in order to execute methods that require special access or root. For this case, the mobile application implements root checking and disables the ability of using the application in rooted devices or, in the case of iOS devices, jailbroken devices.

4.3.4 Phishing. Phishing attacks are attacks that can be performed by deceiving users so that they access websites or install applications, which are controlled by the attacker. The intent behind phishing is to gain personal information of the users of this system so that attackers can exploit this system and potentially commit fraud, through the use of personal information, from other users. For instance, in the event that users have installed a malicious fake application, there is high likelihood that their account credentials, tokens and tickets, may be stolen and used by attackers. Taking this into account, the application needs to provide a robust security mechanism that protects users against phishing.

Ticket validators send ticket validations to the ticketing server and, one of its purposes is to allow for the analysis of ticket validations and detect fraudulent behaviour. In the event that users share their tickets, ticket validations would allow for the detection of information sharing, between multiple users. Through analysis of ticket validations, there is the possibility that two or more users would use these tickets in situations where it would be doubtful that only one user would be using those tickets, and system administrators could take preventive action.

Besides that, there is the possibility of implementing multi-factor authentication in the mobile application. With multi-factor authentication, users would be required to authenticate through another method besides their account credentials such as using one-time passwords. In this case, users would be required to introduce the OTP value sent, by the authentication server, to the mobile phone of the owner of the account.

5 CONCLUSIONS

With the recent advances in mobile ticketing, several research groups have been trying to implement a ticketing system which makes use of barcodes such as QR codes, some with relevant results. However, after performing the analysis of these systems, we came to the conclusion that these implementations still did not meet all the required criteria and that there is room for improvement. For instance, one approach contained the implementation of a ticketing system that provided the offline validation of tickets, through the use of digital signatures. The approach to the system would be able to correctly validate tickets with a single validator but, in case of a distributed system that involves the offline validation of tickets over multiple validators, this system would not provide the reliable validation of tickets and prevent fraud.

To conclude, the proposed solution focuses on a ticketing system which provides the accurate validation of tickets by ticket validators and allows users to perform the generation of QR codes, within their mobile phone application. By providing an integration of a mobile tickets with one time passwords, the proposed solution is a mobile ticketing system, based on the dynamic generation of QR codes. Even though the user application and system aims to focus on the general concept of mobile ticketing, this system can be adapted to a specific field such as public transports. Some of the characteristics of system are innovative and involve the implementation of QR codes with one-time passwords in order to provide tickets that can be used for a mobile ticketing system. Our solution meets the

requirements and research objectives that were initially proposed, with an innovative approach and margin for improvements.

ACKNOWLEDGMENTS

I would like to thank my supervisors Prof. José Manuel Alves Marques, Eng. Joel Teixeira and Eng. Hugo Bicho for their knowledge, experience and support.

REFERENCES

M. Mezghani. Study on electronic ticketing in public transport. Final Report, European Metropolitan Transport Authorities. May 2008, http://www.emta.com/IMG/pdf/EMTA-Ticketing.pdf (date of access 25.04.2015). URL http://www.emta.com/IMG/pdf/EMTA-Ticketing.pdf.

Darian Škarica, Hrvoje Belani, and Sanja Illes. Implementation and evaluation of mobile ticket validation systems for value-added services. pages 260 – 264, 10 2009.

Network Time Protocol. URL https://www.techtarget.com/searchnetworking/definition/Network-Time-Protocol.

Víctor Gayoso Martínez, Luis Hernandez Encinas, and Araceli Queiruga-Dios. Security and practical considerations when implementing the elliptic curve integrated encryption scheme. *Cryptologia*, 39:1–26, 05 2015. doi: 10.1080/01611194.2014.988363.