# Ticketing system with a secure QR Code and a new payment system

João Carreira
joao.duarte.carreira@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

September 2020

**Abstract**

Public transport is a very important part of the public service, especially in large cities, where smart card technology is widely used in ticket systems, as it guarantees great security of the data contained but allows users to transmit the card to other people and must be recharged in machines suitable for this purpose. Currently, there is a greater diversity of transport, with shared bicycles appearing for example, guaranteeing a greater supply of transport for the population although there is a lack of interconnection between transport. These disadvantages could be minimized using secure QR Codes as a new ticket system and the blockchain to enable greater integration of transport, creating a new payment system. Thus, two solutions were designed and developed. One that allows the ticket to be presented in the form of a QR Code via the smartphone (a generalized technology) and guarantees the security of the data encoded in the QR Codes through an encryption algorithm and the generation of keys by the DUKPT algorithm, together with a periodic update of the data, ensures strong security and makes it difficult for users to pass the ticket. The other, a new payment system using blockchain technology, which allows creating a greater interconnection between transports, with a payment method through purchased credits (vouchers) that can be used on any public transport. Finally, with this work, libraries were designed and implemented for the company Card4B Systems, S.A, with the objective of a later integration in the systems present in public transport.

**Keywords:** Public transports, Ticket system, QR Code, Security, DUKPT, Payment system, Blockchain

## 1. Introduction

In recent years there has been an increase in the offer of public transport services in large cities, such as Lisbon, which has seen an increase in different modes of transport such as shared bicycles. The most recent services have adopted a ticketing system through the use of smartphones while older means, such as trains or buses, use smart cards and paper tickets.

In the case of paper tickets, widely used for ticket purchases on the transport, it is required that a payment be made and only after the ticket is issued, spending time and paper. When the ticket is represented by the smart card, despite the fact that it contains strong security that is able to prevent counterfeiting and its alteration as well as a quick validation, it is required that they be reloaded in specific machines, spread over the different stops, or on the transport, in which in both situations requires a waste of time either for the user or for transportation that fails to meet the pre-established schedule. In addition, the card can be easily forgotten or even lost, leaving the user without access to the transport service. For the service provider, there is no way to guarantee that the card is always used only by the cardholder, as it is only verified when there are inspections. Payment in the different transport services applications is through the ATM card, in which the user has to expose the data and it can be seen as insecure from the user's point of view, or through purchased credits that can only be used on that service.

To overcome the disadvantages of current systems, an alternative may be the use of smartphones to present tickets, being a very widespread technology in the Portuguese population in which about 75.1% [1] has one. Thus, there is no longer a need to purchase tickets on machines or in transport, and can be purchased using the mobile applications of the service the user intend to use, with the ticket shown on the smartphone screen, in the form of QR Code, with strong data security and quick validation by the transport system. The smartphone also helps in controlling the use of the ticket, since

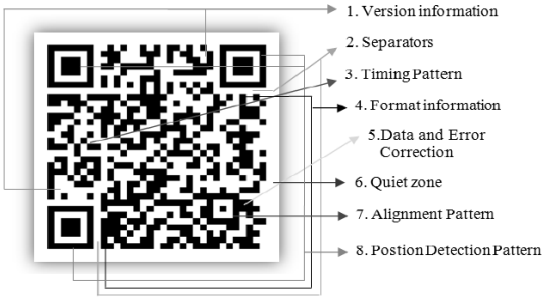



**Figure 1:** Representation of the qr code structure[5].

a facial identification can be performed through the camera incorporated in it. With different applications for each service, in order to have a greater interconnection between them and a payment method that can be presented in all services without constantly exposing the user's bank card, a new system is developed with Blockchain technology that allows to make ticket payments through purchased credits that can be used on any service that belongs to the network.

## 2. Background

Two technologies are used in the development of the systems, QR Code and Blockchain, in order to be used in the ticket systems and in the payment systems, respectively. Security algorithms are also studied to ensure that the security of the systems remains or improves compared to current systems.

### 2.1. Quick Response Code

QR Code [2] is a matrix code, with a structure represented in figure 1, developed in 1994 in Japan by Denso Wave in order to assist manufacturing processes. This code overcomes the limitations of conventional barcodes. A major limitation is the ability to encode characters, which with the QR Code has increased, making it possible to encode up to 7089 numeric characters, 4296 alphanumeric characters, 1817 kenji characters and 2953 bytes [3]. In total, there are about 40 versions of the QR Code that differ according to the number and type of characters they manage to code, along with the level of correction. In addition to its great ability to encode characters, about 23648 bits of mixed data, it has a very fast and effective reading ability, regardless of the reading angle [4]. It is also resistant to distortion and damage, guaranteeing a reading even if the code is damaged, up to a certain percentage according to the level of correction used when creating it.

The first code developed represents the most used two-dimensional code and is called QR Code model 2 [6], an improved version of QR Code model 1, the original. Denso Wave has also developed the following four codes:

- Micro QR Code [7]: code smaller than the traditional, with only a position detection pattern, the traditional uses three, although it has a lower character encoding capacity.
- iQR Code [8]: this code allows to encode the same number of characters in a code with a size 30% smaller than the traditional one, and can even have smaller and larger sizes, managing to encode 80% more data. It can also adopt a rectangular shape, allowing its use on cylindrical surfaces for example.
- Frame QR Code [9]: with this code it is now possible to place an image in the center of the code.
- Secure QR Code [10]: this code makes it possible to encode two types of data, public and private, in which the private data is encrypted and can only be read through specific readers.

Of the different types of QR Codes presented, only the QR Code model 2 and the micro QR Code are standardized.

However, QR Code is a code that allows you to encode any type of information without being able to guarantee its authenticity, integrity, confidentiality and non-replication. Attacks can be made through them, such as Phishing in which a fake website url is encoded to extort information from those who read it. In the QR Code all encoded data can be read by any reader, the data can be changed and a new QR Code can be created easily without the reader being able to detect it and it is easily replicated.

### 2.2. Blockchain

Blockchain is a decentralized and distributed ledger [11] that functions as a database that allows to keep a permanent record of transactions, in which individual transactions and blocks are grouped. Blocks are structures that allow to store a list of transactions and are always linked to the previous block through a cryptographic hash, and transactions are created and made between network neighbors and allow you to change the blockchain. Among the network's stakeholders there is a pre-established consensus for there to be a consensus between entities that are mutually suspicious.

A blockchain network can be of two different types [12] and can have two different types of players, writers and readers. A writer writes the status in the database, helping to grow the blockchain and intervening in the consensus process, unlike the reader who does not help to grow the blockchain, only participates in the process of creating the transaction, with the

functions of reading, analyzing and audit the blockchain.

- Permissionless blockchain: this type of blockchain allows any player to join or leave the network at any time, either as a reader or writer, with no entity that manages the enrollment of new players.
- Permissioned blockchain: in this type of blockchain there is a central entity that decides and assigns the right to the stakeholders to participate in the blockchain's reading and writing operations.

The technology is open source, and can be used by anyone to implement systems using it. It also allows for transparency between network stakeholders and each network block is independently verified through consensus models that provide rules for block validation[13].

In a network such as blockchain in which the actors do not trust each other, it is necessary to create a consensus among all so that it is possible to ensure that the ledger is consistent among all. From several existing consensus protocols , the following four are presented [13]:

- Proof of Work (PoW): this protocol is used in the Bitcoin network and to validate a transaction it is necessary to demonstrate work in order to prove that who validates the transaction does not intend to attack the network. This work involves substantial calculations, needing to calculate the value of the block hash that is lower than the target. Upon reaching the value, the block is transmitted to the network for the rest to confirm the value that, in case it is accepted, the block is added to the blockchain.
- Proof of Stake (PoS): in this algorithm there are validators that are nodes that validate the next block, chosen at random but, with a probability of being chosen depending on the amount of coins they have, the more coins the more likely to be chosen. In the case of validating fraudulent transactions, the validator lose coins, thus motivating to do the job correctly.
- Pratical Byzantine Fault Tolerance (PBFT)[14]: this algorithm tolerates Byzantine failures, malicious nodes to propagate incorrect information in the network, up to a maximum of 1/3 of malicious nodes. At each round a primary node is selected to be responsible for ordering the transaction. Initially, the client sends the operation request to the primary node and the latter sends a request to the remaining nodes, leaving them with the decision to accept or not. Each node responds to the request received by sending its decision to the other nodes and, if 2/3 accepts, the commit process begins, in which the nodes send the commit request to the rest and, if a node receives 2 / 3 of orders accepted, execute the instructions for the requested operation.
- Delegated Proof of Stake (DPoS): in DPoS an indirect democracy policy is used in which nodes called "delegates" are chosen as an authority representative who acts on behalf of other nodes, generating and validating the blocks. This process validate transactions faster.

The blockchain allows digital transactions of assets, financial or not, in which a transaction comprises two phases, the signature phase and the validation phase. The signature phase[15] comprises the signature with the customer's private key and is addressed with the recipient's public key, ensuring authenticity, integrity and non-repudiation. In the validation phase[16], the transaction is represented by a block that is spread between the nodes. The customer who receives the asset, checks with the issuer's public key that the subscription is correct. When broadcasting the block, each node must verify that the transaction complies with the rules previously programmed in each blockchain client, only retransmitting if it is valid.

In order to incorporate more extensive instructions in the blockchain transactions there is a Smart Contract [17], a self-executing contract with the terms of the agreement between the stakeholders. The cycle of the smart contract[18] consists of four phases: **creation** in which the parties accept the objectives of the contract and this is coded and submitted to the ledger, **freezing** in which transactions are frozen for those who created the contract in order for us to ensure that the conditions for the execution of the contract are met, **execution** where it is saved in the ledger and read by the participating nodes, resulting in a group of new transactions, and **finalization** in which the transactions from the previous phase are saved in the ledger.

#### 2.3. Security

Security is a key point of the project and it is possible to guarantee characteristics such as confidentiality, authenticity and integrity to the project through existing algorithms.

To guarantee confidentiality, there are

cryptographic algorithms to make plain text into unreadable text, in order to ensure that only information is read by authorized entities. In this context, only symmetric key encryption algorithms are taken into account as they are faster than asymmetric key algorithms and the system must guarantee speed.

There are several algorithms although just one will be presented, the AES algorithm. Advanced Encryption Standard[19] is block encryption algorithm in which one block of 128 bits is encrypted at a time. The number of rounds to obtain the ciphertext is determined by the key size which can be 128 bits (10 rounds), 192 bits (12 rounds) and 256 bits (14 rounds). Each round consists of SubBytes, Shift Rows, Mix Columns and AddRound Key operations.

Any algorithm needs an encryption key and there are algorithms that manages the keys in a way that does not always use the same one, which would weaken the algorithm after several uses. One of the algorithms is called Derived Unique Key Per Transaction (DUKPT)[20] which allows each transaction to be encrypted with a different key. This algorithm uses 4 keys, the base derivation key (BDK), the initial pin encryption key (IPEK), the future keys and the key serial number (KSN), a key formed by the Key Serial ID (BDK ID), by the Device ID and a transaction counter. BDK is the master key and is the key that initiates the process, in which the IPEK is created and injected into the devices together with the KSN, being different for each device. On the device, future keys are created from the IPEK that are used to encrypt, being chosen a key for each transaction through specific operations. When decrypting, the device only needs to have the BDK and KSN corresponding to the key used in the encryption in order to be able to calculate the key used since the KSN informs which BDK was used, the source device and the transaction.

In security, to ensure data integrity, hash cryptographic algorithms are used to obtain a hash, preventing the entry value from being obtained. This hash value makes it possible to guarantee that, for example, the data stored in a cloud remains unchanged, comparing the initially calculated hash and the one that it was subsequently saved safely with the hash calculated when verifying the data, which, if they are the same, guarantees that the data has not changed.

Hashes are created using hash functions, with two families, Message Digest (MD) and Secure Hash Algorithms (SHA). The MD family is based on simple operations on 32-bit words, MD5 being the fastest algorithm but, as in the whole family, there is the possibility of collisions. In the case of the SHA family[21], collisions are already more difficult to happen in functions SHA-2 and SHA-3, whereas in functions SHA-0 and SHA-1 collisions have already been found.

In order to guarantee the authenticity and integrity of a message, a Message Authentication Code (MAC) is built, which is based on cryptographic dispersion functions. One of the existing types of MAC is the Hash-based Message Authentication Code (HMAC)[22] which involves a cryptographic hash function and a cryptographic key, according to formula 1, obtaining a final value through XOR operations that involve the key, message and constants and, through a hash function.

$$HMAC(K,m) = H((K \oplus opad)||H((K \oplus ipad)||m)) \quad (1)$$

**3. Methodology**

In order to maintain a quick reading and validation as well as strong security through the use of tickets in the form of smart cards, BilheteSeguro is designed, capable of creating a secure QR Code to be used as a ticket or as a form confirmation of a ticket purchase made through the PagamentoVoucher solution. The PagamentoVoucher is a new payment method capable of being used on any public transport, for example buses or shared bicycles, which is based on Blockchain technology and allows the exchange of credits between customers. In the case of the BilheteSeguro system, this proposes that the application has already integrated a login and a purchase system.

With the BilheteSeguro system it is intended to use the QR Code as a way to present the ticket through the smartphone, which must have a high level of security in order to protect the data contained therein, using the encryption algorithm together with DUKPT. To make the transmission of the ticket difficult, the QR Code is renewed every time and has a validity for each order, making it necessary to place an order whenever you need to get on public transport. In the validation, the application calculates which key is used in the encryption and validates the attributes encrypted in the QR Code.The system consists of a web service, a module for the application of the transport service used by the customer and a module for the application of ticket validation.

The web service aims to provide data to the customer's application whenever the customer needs to show a ticket and, for this, the application needs to place a ticket request to the service in which it sends information, such as the title identifier purchased, the customer identifier , an

image of the user's face and the title body purchased previously, accompanied by a signature created using the HMAC algorithm, using as a key the token received at login. After verifying the signature, the customer's identity and the validity of the ticket, a key is created using the DUKPT algorithm, the IPEK and, fields such as the time of issue, the time it expires, the update time, are defined, the ticket status, the time of the last validation and the location of the last validation, all sent in response together with a signature.

The module present in the client application is called BilheteSeguro_Lite and is responsible for the creation of the data with security that will be coded for QR Code. Before it is possible to place a ticket order with the service, the customer must have previously purchased a title and, whenever placing an order, the customer's face must be captured for later confirmation of identity. The module makes the ticket request to the service and, after receiving the response, starts the ticket creation process. The ticket consists of the identifier of the purchased ticket, the customer's identifier, the smartphone's IMEI to ensure that it is the customer himself, the time of the update period and, the time of issue, the time it expires, status and, time and location of the last validation received in the service response. With the key received in the response, new keys are created, starting the DUKPT process in which a new encryption key is generated for each QR Code. The service response also contains the update time (how often a new QR Code has to be generated), the time of the issuing period being the maximum time for which that QR Code is valid. This process is executed until the time of the order expires. In the end, the data to be encoded in QR Code is the encrypted ticket, the KSN of the encryption key and a signature calculated using the HMAC algorithm.

The ticket validation application already has the ability to read QR Codes, but misses the module that allows decrypting and validating QR Code tickets using the BilheteSeguro system. In this module, the authenticity and integrity of the data encoded in the QR Code is verified through the HMAC algorithm that calculates a new signature and compares it with the received one, and the key that was used through the KSN of the used key is obtained, which allows obtaining information on how to get to the key. This process begins with the calculation of the IPEK because the application must have the master key (BDK) and then calculates the future keys until it finds the key with a KSN equal to that received in the QR Code, a process in which only the keys needed to obtain the correct one are calculated. When the module is able to obtain the encryption key, it decrypts the ticket and performs a temporal validation to verify if the time period of the update period is greater than the time when the QR Code was detected and, if the time of detection is between the time of emission and the time of expiry. In case of confirmed validity, the application validates the remaining data received, title identifier, customer identifier and IMEI.

Since that in public transport it is usual to have inspection, the same module developed for the application of public transport is used in inspection applications. The whole process of decryption and temporal verification is the same, differing in the final step in which it has to validate the data referring to the last validation, the time and place, in order to identify whether the customer is valid on that transport.

The PagamentoVoucher system allows the existence of a payment method through the purchase of vouchers common to all transport services using the blockchain, developing the business logic programmed through chaincodes. To ensure that blockchain players are only authorized customers, it is necessary to use a private permissioned blockchain, with the Hyperledger Fabric project allowing this. This project has several APIs and one of them allows to create and register a certificate for each stakeholder who, whenever he/she wants to execute an action on the blockchain, must use the certificate and it must be verified by an entity. The Hyperledger Fabric then allows the transfer of assets, keeping the most current state of each asset in the world state and all the transactions made on them in the transaction log.

There may be different players on the network, application customers who may be users of the service, entities that charge fees, entities that issue vouchers or different transport services. To identify which type of application customer, when creating the certificate, a specific attribute is inserted.

The assets of the PagamentoVoucher network can be of three types, Account, Voucher or Transfer. The Account is the asset that represents the virtual wallet of an application customer, where the number of credits purchased by the customer are stored and, to access, you need access credentials. This asset has as attributes the creation date, the number of credits, the mobile phone number and the area code, which was the last operation and the date, and a key composed of the area code and telephone number, with specific codes and numbers for entities that charge fees and transport services. In order for the Account to have credits, it is necessary to purchase Vouchers, which will be associated with a user using the key that contains the Account asset key. The

Voucher has the number of credits purchased, the entity that issued it, the identifier of the user who purchased it, the date of purchase, the expiration date and the status, which can be ”New”, ”In use”, ”Empty” or ”Out of date”. Whenever a transfer is made, the Transfer asset is created, allowing to store all transfers and data of each one as the transferred value, issuer, recipient, date, collecting entity, commission amount, key, which corresponds to the transfer identifier and, the status, which can be ”Sent”, ”Received” or ”Confirmed”.

Hyperledger Fabric, in addition to saving assets, also allows the storage of private data that is only accessible within an organization. Thus, the PagamentoVoucher allows the user to privately store the access data to the assets as well as their profile, and in order to access the data, the certificate used when they were created is necessary, thus protecting the data against access by other users.

For the user to make transactions on his assets, rules are created through three smart contracts, AccountContract, VoucherContract and TransferContract, one for each type of asset, thus defining the business model. AccountContract allows you to perform operations on the Account asset, in which it is possible create, change the status or consult it. After the asset is created, whenever the user wants to access it, they need to login to obtain a session key that will be verified in each operation. When the user want to make a transfer, needs to have credits and they are removed from the sender's assets and added to the receiver's assets, creating a Transfer asset with the respective data. When a transfer is made between two users of the application, a Voucher associated with the recipient is created to ensure that the number of credits in the Account corresponds to the sum of credits from all Vouchers. VoucherContract allows you to create, modify and consult Voucher-type assets. Finally, the TransferContract that creates, modifies and consults Transfer-type assets, tracks the transfer status so that, when a transfer is confirmed, it is not possible to confirm it again, thus preventing, for example, entering public transport twice by making only one transfer. Not all operations present in smart contracts are available to all customers of the application, there is a verification of the attribute present in the certificates that differentiates customers in order to ensure that that customer can perform the operation.

For the client application to be able to carry out transactions or perform queries according to the rules present in smart contracts, applications must exist. Each application can perform transactions or queries on the blockchain by invoking the different operations existing in smart contracts, establishing a gateway to a network node and using the user's certificate.

**Figure 2:** Demonstration of the validation of a ticket in the form of QR Code


**(a)** QR Code scan **(b)** Valid QR Code **(c)** Invalid QR Code

**4. Results & discussion**

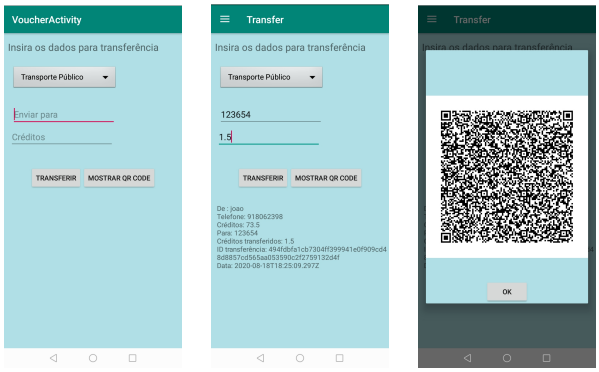
In order to test both systems in an environment that simulates a real scenario, two applications have been developed that allow testing and guaranteeing the correct functioning of the modules in the respective applications. In figures 2 and 3 are shown some screens of the two applications developed for tests, referring to the ticket validation application and the user application, respectively.

In figure 2, the validation of a ticket in the form of a QR Code is represented, in which the validation application detects the QR Code and the validation process begins, being concluded when a valid ticket, figure 2b, or invalid ticket, figure 2c, is obtained.

A transfer made through the PagamentoVoucher system corresponds to making the purchase of a given ticket or ticket, as shown in figure 3. To make the transfer, the recipient's phone number must be entered as well as the amount of credits to be transferred, figure 3a. When the transfer is complete, the answer is obtained, figure 3b and, for the ticket validation application, confirm the transfer a ticket is created in the form of QR Code, figure 3c.

To calculate the validation time of a ticket in the form of a QR Code, 50 tests were performed and each test included the validation of 60 QR Codes in a row, containing different information and cryptographic key, the average times being represented in the graphs of the figure 4. In order to decrypt the QR Code data, the ticket validation application needs to calculate the encryption key used, and at most it needs to calculate 3 keys to find the key used. The results obtained are shown

**Figure 3:** Demonstration of a transfer on the PagamentoVoucher and QR Code of the payment for confirmation on public transport.




**(a)** Transfer operation **(b)** Transfer result **(c)** Payment QRCode

in table 1. The Android Studio emulator was used with the user application and a Huawei Y7 2019 smartphone with the ticket validation application installed, where the times were calculated from the detection of the ticket in the form of QR Code to the final result as in image 2b, if valid. The time was calculated using the TimingLogger class present on Android.

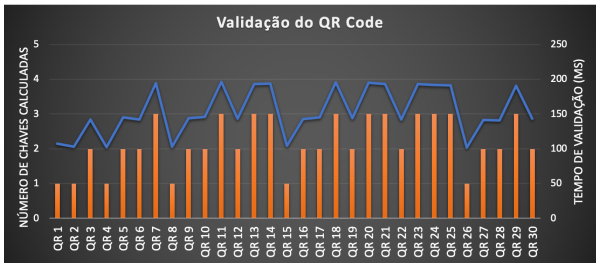
| Number of keys | 1 | 2 | 3 |
|---|---|---|---|
| Number of QR Codes per test | 7 | 21 | 32 |
| Average time (ms) | 102,94 | 141,89 | 191,35 |

**Table 1:** Average validation time.

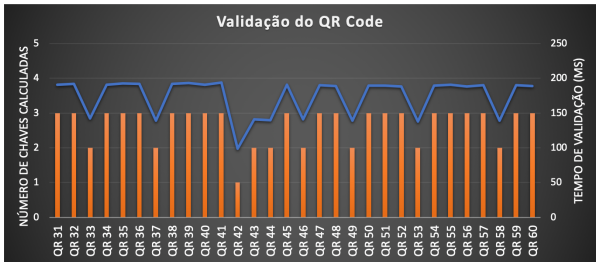
With the data in the table 1 we can see that the validation time is dependent on the number of keys to calculate until finding the encryption key, being faster when only one key is calculated. Even though it is slower to validate when calculating three keys, it continues to perform well, with an average below 200 milliseconds, a very positive result.

In order to verify the performance of the PagamentoVoucher system, transfer times and transfer confirmation times were calculated in the user application and in the ticket validation application, respectively. In the testing environment, a Huawei Y7 2019 smartphone was used with the user application, a Samsung smartphone with the ticket validation application and a laptop computer with the blockchain test network of the Hyperledger Fabric project to be run on an Ubuntu virtual machine. The times obtained are shown in table 2, with the time calculated in milliseconds and seconds.

**Figure 4:** Validation time of 60 tickets on the public transport device..




**(a)** First 30 QR Codes.




**(b)** Last 30 QR Codes.

| | Transfer time | Confirmation time |
|---|---|---|
| Average time (ms) | 5368 | 2837 |
| Average time (s) | 5.4 | 2.8 |

**Table 2:** Time tests of the PagamentoVoucher system in the credit transfer process.

## 5. Conclusions

The ticketing systems existing today have their advantages and disadvantages just like any system, already discussed in the introduction to this work. In order to evolve the systems, two technologies were identified that were explored, identifying the advantages and disadvantages of both the QR Code and the Blockchain technology, together with security algorithms, in the study done before moving to the architecture and development of any of the systems.

The BilheteSeguro system through a strong security algorithm, AES, is able to guarantee data encryption, aided by the DUKPT key management algorithm, in order to prevent the original data from being read or even falsified. With the QR Code it is possible to guarantee a way to show the ticket to the validation system, with the guarantee of a quick reading and without the need to have it in a certain position, just in the view of the device's camera. The entire validation process is able to guarantee fast validation with very positive results obtained, although it needs to be tested in a real environment.

Regarding the PagamentoVoucher system, blockchain technology allows for a decentralized network, without a single point of failure, with the capacity to transfer assets between customers. Using the Hyperledger Fabric project, it is possible to create a permissionless network, with a good library and documentation. The developed system allows the creation of a virtual account to which only the owner has access, and only he/she is able to consult and execute operations on the asset. The great advantages of the system are the ability to be used in any transport service and the possibility of reducing ticket purchase times, since it can be done via smartphone anywhere and, when entering transport, it is only needed to to present proof of payment, in this the QR Code with the data.

In conclusion, the systems developed like all the others, have advantages and disadvantages, but in short two systems have been developed with the capacity to be integrated into the various transport services, with their added value for ticketing systems. In both systems, a greater number of tests in real situations is necessary to assess their behavior, as the devices installed in the different transports have different hardware that can influence the validation time and the size of the blockchain network and a greater number of requests received within a certain time interval can influence the calculated times. With the evolution of smartphones, the way of presenting the ticket can also evolve, as for example through NFC technology, with more and more smartphones with this technology.

**References**


[1] Daniel Almeida. 7 milhões de portugueses têm smartphone – marketeer, Aug 2018. (Accessed on 24/05/2019).

[2] Robin Ashford. Qr codes and academic libraries: Reaching mobile users. 2010.

[3] Jun-Chou Chuang, Yu-Chen Hu, and Hsien-Ju Ko. A novel secret sharing technique using qr code. *International Journal of Image Processing*, 4(5):468–475, 2010.

[4] Tan Jin Soon. Qr code. *Synthesis Journal*, 2008:59–78, 2008.

[5] Jumana Waleed, Huang Dong Jun, Sarah Saadoon, Saad Hameed, and Hiyam Hatem. An immune secret qr-code sharing based on a twofold zero-watermarking scheme. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):399–412, 2015.

[6] Qr code model 1 and model 2 — qrcode.com — denso wave. `https://www.qrcode.com/en/codes/model12.html`. (Accessed on 05/02/2019).

[7] Micro qr code — qrcode.com — denso wave. `https://www.qrcode.com/en/codes/microqr.html`. (Accessed on 05/02/2019).

[8] iqr code — qrcode.com — denso wave. `https://www.qrcode.com/en/codes/iqr.html`. (Accessed on 05/02/2019).

[9] Frame qr — qrcode.com — denso wave. `https://www.qrcode.com/en/codes/frameqr.html`. (Accessed on 05/25/2019).

[10] Sqrc® — qr code solutions — system solution — denso wave. `https://www.denso-wave.com/en/system/qr/product/sqrc.html`. (Accessed on 05/02/2019).

[11] J Michal, ALAN Cohn, and JARED R Butcher. Blockchain technology. *The Journal*, 1:7, 2018.

[12] Karl Wüst and Arthur Gervais. Do you need a blockchain? In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 45–54. IEEE, 2018.

[13] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 557–564. IEEE, 2017.

[14] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. A review on consensus algorithm of blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2567–2572. IEEE, 2017.

[15] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and smart contracts for the internet of things. *Ieee Access*, 4:2292–2303, 2016.

[16] Michael Crosby, Pradan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman, et al. Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2(6-10):71, 2016.

[17] Melanie Swan. *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.

[18] Christian Sillaber and Bernhard Waltl. Life cycle of smart contracts in blockchain ecosystems. *Datenschutz und Datensicherheit-DuD*, 41(8):497–500, 2017.

[19] Ako Muhamad Abdullah. Advanced encryption standard (aes) algorithm to encrypt and decrypt data. *Cryptography and Network Security*, 16, 2017.

[20] Kirsty Trainer. "key" to secure data - p2pe - derived unique key per transaction (dukpt). `https://www.foregenix.com/blog/p2pe-derived-unique-key-per-transaction-dukpt`, Nov 2015.

[21] Secure hash algorithms - wikipedia. `https://en.wikipedia.org/wiki/Secure_Hash_Algorithms`. (Accessed on 01/15/2020).

[22] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. Hmac: Keyed-hashing for message authentication, 1997.