QRP: An improved secure authentication method using QR codes

David Pintor Maestre
Universitat Oberta de Catalunya
08018, Barcelona, Spain
dpintor@uoc.edu

June 8, 2012

Abstract

It seems quite obvious that any online service that aims to be secure nowadays should seriously consider implementing a strong authentication method. This paper presents the design and implementation of QRP, an open source, proof-of-concept authentication system that uses a two-factor authentication by combining a password and a camera-equipped mobile phone, acting as an authentication token. QRP is extremely secure as all the sensitive information stored and transmitted is encrypted, but it is also an easy to use and cost-efficient solution. QRP is portable and can be used securely in untrusted computers. Finally, QRP is able to successfully authenticate even when the phone is offline.

Keywords: QR code, two-factor authentication, pincode, open source.

1 Introduction

In a modern world where we are able to do almost everything on-line (banking, shopping, communicating, storing and sharing personal information...), it is nowadays a critical matter to be able to access these services in the most secured manner. Indeed, as viruses and cracking methods become more complex and powerful by the day, the available security techniques must improve as well, allowing users to protect their data and communications with the maximum confidence.

Some research has been undertaken on the subject of secure authentication: Rif-Pous [5], Young Sil [6], Kuan-Chieh [7], Mukhopadhyay [8] and Starnberger [9] present multiple approaches to this issue, the last four using QR codes and One-Time Passwords as an advisable manner to authenticate. Also, various authors have raised the issue of securely paying online, which is absolutely relevant

for our work as well, as this is achieved in a very similar way than for securely authenticating. Lee et al. [3] and Maritz [4] describe in their papers different methods to securely effectuate payments.

All these approaches consider security as an extremely important point, and for this reason, most of them ([4],[5],[6],[7],[8] and [9]) use a two-factor authentication method in the design of their solution. A two-factor authentication method is composed by two of the three authentication verification methods below:

- Something the user knows, such a password or a pincode.
- Something the user has, such a token or a smart card.
- Something the user is, such a fingerprint or a retinal scan.

It seems obvious that the higher the factor, the most secure will be the authentication method. Even though single-factor authentication is still widely used, even for critical services such as electronic banking, this doesn't seem to be a good approach as per the obvious dangers involved. Using multiple authentication methods from the same verification method is not a good approach either: using a password plus the ID number or date of birth (or even the three of them) is still a single-factor authentication method and thus, not secure enough for critical transactions.

The aim of this paper is to develop an authentication method using a two-factor authentication: a trusted device (a mobile phone) that will read a QR code and that will act as a token, and a password known by the user.

This paper is structured as follows: first, I will give a brief introduction about what are QR codes. Then, I will detail the requirements of the proposed system. Then, I will describe in detail how the system works. In the following section I will explain the computational costs and security considerations and finally, I will describe my conclusions about the system.

2 QR codes

In 2002, Clarke et al. were probably one of the first to suggest the usage of camera-based devices as an alternative, more secured authentication method for critical transactions, such as banking operations, and most particularly when connecting from untrusted computers [1]. While QR codes are not explicitly mentioned in that paper, their authors were somehow visionaries: the amount of camera-quipped smartphones around us is increasing so rapidly that mobile-based authentication might become a popular method to authenticate in a short time.

QR codes (Quick Response codes) were introduced in 1994 by Denso-Wave [2], a Japanese company subsidiary of Toyota. Initially, these codes where conceived as a quick way to keep track of vehicle parts, being nowadays extremely popular in Asian countries like Japan, South Korea, China or Taiwan and becoming more and more popular in western countries by the day.

QR codes are two-dimensional bar codes, so they can be read from any direction in 360; can store up to 4,296 alphanumeric characters (or up to 1,817 Japanese *kanjis*), which is much more than the 20 digits that a traditional bar code can store; have a great deal of resistance to damage, being readable even if they are partially damaged; and they are easy and quick to read with a camera-based device. Its versatility has made them quite popular among some industries, particularly in the advertisement world, where these are today widely used as a way to quickly store an URL by scanning it with a camera-equipped mobile device.

On the other hand, QR codes are unfortunately only understood by machines and not by human beings. This means that scanning QR codes may entail some security issues: the user doesn't really know what is behind the QR code, so she might be scanning malicious code without being aware.

3 System requirements

A prototype called QRP has been implemented as a proof of concept, meeting the following requirements:

(a) Security: On the server side, our application is hosted in an Apache web server. As the communications with the web server need to be secure, a 2048-bit secure key and a self-signed SSL certificate have been generated to encrypt all communications, using https and therefore, ensuring that all data sent and received between client and server passes through a secure channel (SSL/TLS).

The application is developed in PHP5 and a Postgres9 database is used in the back-end to store some user and transaction information. The database has been configured to be only accessible internally.

At a OS level, a Fedora 14 distribution has been chosen as it is a reliable and secure operating system. The *iptables* firewall has been configured in order to accept only connexions on ports 80/tcp and 443/tcp. The *Apache* web server has been also configured to automatically redirect any http connexion requests to https, so there is no chance of unsecured communication.

On the mobile side, an Android application called QRP Scanner has been developed to authenticate on the server after reading and processing a QR code.

In order to access the application, a correct password needs to be entered by the user. The password is stored encrypted in the internal storage of the phone, which is not accessible directly by the user or another application (unless if the device has been previously rooted). The application is compatible with Android 2.2 onwards and uses an external application (ZXing) to read the QR codes, so installing this is a requirement before using QRP Scanner.

(b) Cost efficiency: Only non proprietary software is used to implement this prototype (Linux, Apache, PHP, Postgres, ZXing, Java) and no other non-free services, such as SMS and similar, are used for this project.

(c) Ease of use: Some attention has been put as well in the design of both web/phone applications, implementing a neat, user-friendly and consistent interface in both ends. A one-step authentication method can be achieved when the user's phone is online, improving the system presented by Mukhopadhyay [8], which uses a two-step authentication. In our system, the user is automatically recognised when the communication between the server and the mobile device is taking place.

(d) Phone online/offline: A registered user will be able to authenticate using her phone, even when this is not connected to the network, and so, unable to reach the server. To achieve this, the phone will generate a pincode that will be shown to the user and that will need to be input, along with her username, in the website. A more detailed explanation of this process will be given in the next section. Our system improves the work presented by Kuan-Chieh and Wei-Hsun [7], offering to the user the ability to authenticate even when the phone is not web-enabled.

(e) Portability: The schema is secure even when authenticating from an untrusted computer, thus improving the system presented by Lee [6]. The user does not need to keep a certificate nor do they need to type anything in the computer, even when entering a pincode if authenticating offline, as a randomly-ordered numeric keyboard will be presented to the user in the screen.

4 Authentication system

Once the requirements to implement this prototype have been described, we need to explain how the actual authentication will take place. For this, we need to understand first how the registration process would work. Then, the authentication method will be described.

(a) Registration: This part is not implemented as the paper is only intended to present an authentication method. The following steps are a suggestion on how to complete the registration process:

- The user would go into the registration section in the QRP web application and would submit her username, password and IMEI number[1] of the phone she intends to use to authenticate.
- After validating the data entered (correct IMEI, password complex enough, etc.), the server would store this information on the database.
- Next, the server would generate a private and public pair of keys unique to the user, that would be stored on the server.
- After this, the user would proceed to download and install the application on her phone.
- The first time the mobile application is run, the user will need to enter her username and password (the IMEI can be verified by the mobile application) and the credentials (user/password) would be validated against the database through a https request to the application server.
- If successful, three files would be imported and stored in the user's phone internal storage: the server's public key, the user's private key and a user data file, containing the user's encrypted credentials. The server's public key will be used to decrypt the credentials file. The user's private key will be used to authenticate in the server.

(b) Authentication: Due to the need to store transactions and users information, we will have the following two tables in the database:

- Table transactions

Column	**Type**	**Modifiers**	
imei	bigint		*User's IMEI*
rn	integer	not null	*Random number*
ts	integer	not null	*Timestamp*

Table 1: Table Transactions

- Table users

Column	**Type**	**Modifiers**	
username	character varying(20)	not null	*Username*
password	character varying(20)		*Password*
imei	bigint	not null	*User's IMEI*

Table 2: Table Users

[1]The International Mobile Equipment Identity or IMEI is a number, usually unique, to identify GSM, WCDMA, and iDEN mobile phones, as well as some satellite phones. It is usually found printed inside the battery compartment of the phone. It can also be displayed on the screen of the phone by entering *#06# into the keypad on most phones. Source: http://en.wikipedia.org/wiki/International_Mobile_Equipment_Identity

When the authentication page in the web application is opened, a QR code containing a random number *rn* between 1 and 999999999 is shown. In the background, a row is created in the *transactions* table, containing the current timestamp *ts* and the random number *rn*.

When the user opens the mobile application, she will need to input the password first. It will be verified against the user's encrypted file containing the credentials and if successful, the scanning application will run. The user can now proceed to scan the code from the web application screen.

The contents of the QR code will be captured and sent back to our mobile application. Our mobile application will then generate a string containing the captured random number and the IMEI of the phone, that will be encrypted using our private key. Next, the mobile application will check the state of the phone and decide whether we are going to authenticate in online or offline mode.

Case 1: Online mode authentication: If the phone detects an active Internet connexion, the steps below are followed (refer to Fig.1):

- The encrypted string plus the username are sent to the web server via POST through a secured channel (https). This means that the IMEI and random number are encrypted twice, and the username once.
- The server decrypts the string using the user public key and verifies that a row exists in the *transactions* table with our random number, updating the row with the IMEI of the user.
- The server checks then that the IMEI is correct and assigned to an user as per the *users* table.
- In case of success, the transaction row will be deleted and the user authenticated.
- A PHP session is created for the user, being destroyed when the user logs off or when the browser is closed.

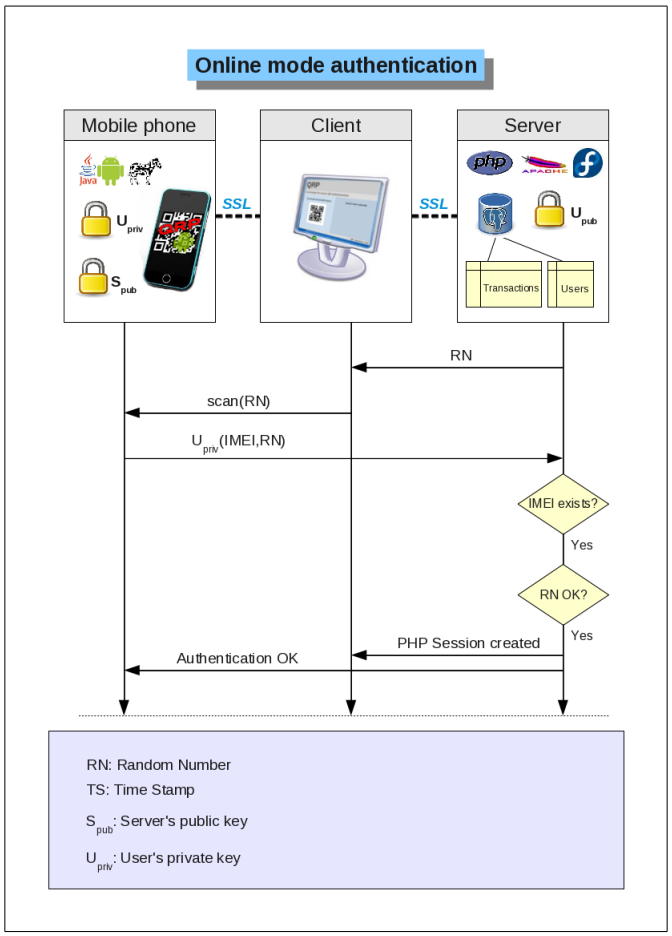

Figure 1: Online authentication diagram

Case 2: Offline mode authentication: If the phone detects that the Internet cannot be accessed, the steps below are followed (refer to Fig. 2):

- Using an internal algorithm, a unique six-digit number is derived from the encrypted string. This number is the pincode that the user will need to input in the authentication screen within the web application, along with her username. The pincode is entered through a screen keyboard, in order to avoid keyloggers.
- The server receives the username and pincode, recreates the pincode using the user's private key, the random number shown and the user's IMEI, the last stored in the *users* table.
- The timestamp is also checked, rejecting the authentication if the random number was generated more than 5 minutes ago.

- If the pincode matches, the transaction row is deleted and the user authenticated.
- Once again, a PHP session is created for the user.

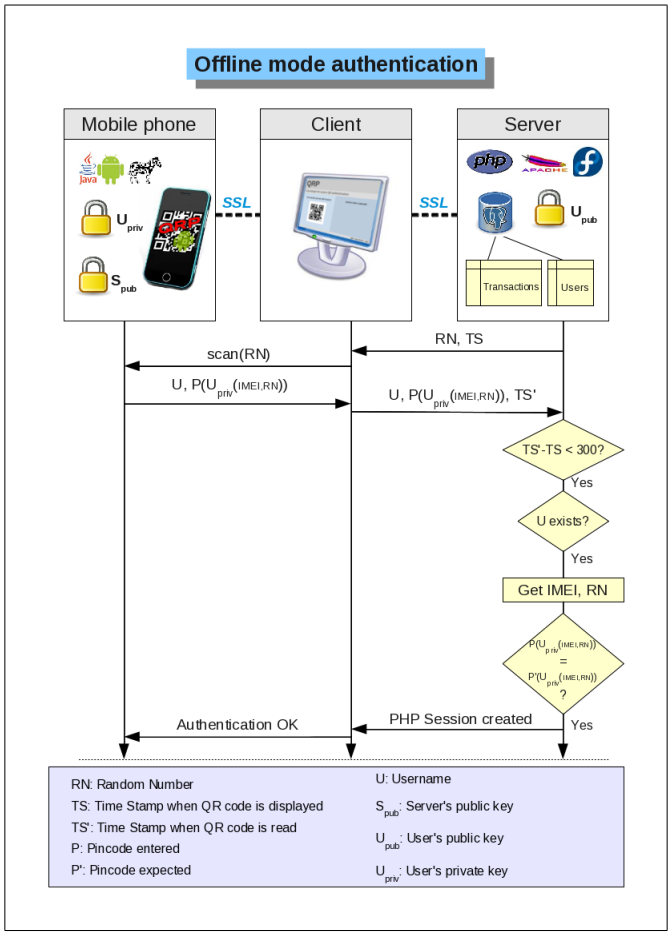

Figure 2: Offline authentication diagram

5 Computational costs and security considerations

In terms of performance, the web application has been tested in an average performance laptop (Intel Pentium T4300 2.1 Ghz processor, 4MB RAM) and the mobile application in a average-high performance mobile phone (1.2GHz Dual Core processor, 837MB RAM). No performance issues have been detected with this configuration. The client configuration is not relevant and just requires

Javascript enabled.

In terms of security, and following the described strategy to achieve authentication, the system would be robust and secure, and very hard for a malicious user to get illegitimate access:

- A man-in-the-middle attack would never succeed as all the communications between the phone and server are always encrypted.
- Repeating or copying a username's pincode would be of no use as it expires as soon as the user authenticates.
- Getting access to a phone with malicious intentions would mean that the person would need to know the password to access the mobile application, as the credentials file is not easily accessible and it is encrypted. Only in the eventuality that the "thief" would know how to access the phone internal storage and how to decrypt the credentials file with the server's public key, she would then be able to open the application and unlawfully authenticate. This set of conditions is obviously very unlikely and the owner would probably have had time by then to deactivate her account or to generate another personal key pair, becoming the one in the misplaced phone automatically invalid.
- A phishing attack on the mobile phone, by replacing the application by another with the same usage, would mean that the password could be recovered. However, without the certificates it would still not be possible to authenticate.
- A phishing attack via web would also be unsuccessful, as the only possible data that a malicious user could collect is the username and pincode (and only authenticating in offline mode). Unless if he's able to reproduce the exact QR code with the same random number and within the same 5 minutes, this could not be reproduced.

Although reliable and secure, the system presented could still be improved in some ways:

- In order to access to the mobile application we need to input a personal password. This might be inconvenient for some people as per the small size of a keyboard in the phone. Some possible solutions would be to use a numeric keyboard or to use pattern authentication1. Additionally, the system could even give the user a choice between different authentication methods to use.
- When using offline authentication, the pincode is generated using the user's private key. In this system, the server must have a copy of the user's private key in order to generate the same pincode. Ideally, we would like to avoid the server having all user's private keys, as if the server gets

hijacked, the security of the users could be compromised. However, there is no easy way to generate an identical pincode in both ends without the same private key. Using the server's private key would not be advisable either as a registered user could easily supplant anyone else's identity. A possible solution might be managing two different user keys, but this would increase the complexity of the system and it is not so clear if it would be worth it.

- Another eventual drawback that we might might want to avoid would be having a single point of failure in the server. This should not be really an issue, as in a real-life system the server should be normally configured as a high-availability cluster with a few redundant nodes.

6 Final conclusions

Nowadays, a relevant amount of people living in a developed country would have a smart-phone able to take pictures and scan QR codes.

The suggested authentication approach is therefore a real possibility in the real world as it makes it more secure than the average authentication method, as it is based in a two-factor authentication method and not in the usual user-name and password approach. The fact that the user does not need to carry any additional device (as she would carry the phone anyway) makes it even easier and more comfortable, and the risks of misplacing the "token" will be reduced: seeing the cost of an average smart-phone and the sensitive information stored on them (contacts, pictures...) we can be sure that most users would look after it.

At the same time, security has become an extremely important matter in our digital society and therefore, two-factor authentication methods should be seriously considered by services that store sensitive information, as they offer a much higher level of security that is definitely worth a few seconds of the user's authentication time.

QR-based authentication offers a very secure and fast authentication method that must be considered to securely and easily authenticate. Indeed, some blue chip companies seem to be testing it already[2].

[2] After Charlie Osborne's article in ZDNet: "Google's QR code log in experiment concluded", http://www.zdnet.com/blog/igeneration/googles-qr-code-log-in-experiment-concluded/14679

References

[1] Clarke, Dwaine; Gassend, Blaise; Kotwal, Thomas; Burnside, Matt; van Dijk, Marten: *"The Untrusted Computer Problem and Camera-Based Authentication"*. Lecture Notes in Computer Science, 2002, Volume 2414, Pervasive Computing, Pages 114-124, Jan.2002.

[2] Denso-wave: http://www.denso-wave.com/qrcode/index-e.html

[3] Lee, Jaesik; Cho, Chang-Hyun; Jun, Moon-Seog: *"Secure quick response-payment(QR-Pay) system using mobile device"*. Advanced Communication Technology (ICACT), 2011 13th International Conference, Feb. 2011.

[4] Maritz, Adrian: *"Secure Payments Using Mobile Device"*. Proyect "iSpaza", University of Cape Town (South Africa).

[5] Rifà-Pous, Helena : *"A Secure Mobile-Based Authentication System for e-Banking"*. On the Move to Meaningful Internet Systems: OTM 2009. Confederated International Conferences, CoopIS, DOA, IS, and ODBASE, Nov. 2009.

[6] Young Sil Lee; Nack Hyun Kim; Hyotaek Lim; HeungKuk Jo; Hoon Jae Lee: *"Online Banking Authentication System using Mobile-OTP with QR-code"*. Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference, Nov.-Dec. 2010.

[7] Kuan-Chieh Liao; Wei-Hsun Lee: *"A Novel User Authentication Scheme Based on QR-Code"*. Journal of Networks, Vol 5, No 8 (2010), 937-941, Aug. 2010.

[8] Mukhopadhyay, Syamantak; Argles, David: *"An Anti-Phishing mechanism for Single Sign-On based on QR-Code"*. The First International Workshop on Application of Artificial Intelligence for Email Management (AAIEM 2011) in London (UK), June 2011.

[9] Starnberger, G.; Froihofer, L.; Goeschka, K.M.: *"QR-TAN: Secure Mobile Transaction Authentication"*. Availability, Reliability and Security, 2009. ARES '09. International Conference, Mar. 2009.

[10] Felt, Adrienne Porter; Wagner, David: *"Phishing on Mobile Devices"*. Workshop on Web 2.0 Security and Privacy (W2SP), 2011.

[11] DeFigueiredo, Dimitri: *"The Case for Mobile Two-Factor Authentication"*. Security & Privacy, IEEE, Sept.-Oct. 2011.

[12] Shamir, Adi: *"How to Share a Secret"*. Magazine Communications of the ACM CACM Homepage archive Volume 22 Issue 11, Nov. 1979.

[13] Jun-Chou Chuang; Yu-Chen Hu; Hsien-Ju Ko: *"A Novel Secret Sharing Technique Using QR Code"*. International Journal of Image Processing (IJIP), Dec. 2010.

[14] Rouillard, José: *"Contextual QR codes"*. Computing in the Global Information Technology, 2008. ICCGI '08. The Third International Multi-Conference, July-Aug. 2008.