2-Factor Authentication with 2D Barcodes

L. Hebbes and C. Chan

WMN Research Group, Kingston University, Kingston upon Thames, UK
e-mail: l.hebbes@kingston.ac.uk, k1058149@kingston.ac.uk

Abstract

An improved user-friendly secure authentication scheme using 2D barcodes is proposed in this paper. The proposed scheme uses a known mobile device as a second factor to decode an encrypted message transferred as a 2D barcode and read via a camera on the mobile device. A one-time passcode is produced from this for authentication. Elliptic curve public-key codes are used on a PKI-SIM card to encrypt a challenge as well as additional temporal data. The proposed scheme improves the security and usability of multi-factor authentication and transaction verification without requiring increased investment in hardware or user training.

Keywords

2-factor authentication, QR codes, 2D barcodes, CHAP, ECC, PKI-SIM

1. Introduction

Virtually all secure systems require restriction of use to authenticated identities only, commonly through a username and password combination. This method is well understood by users and requires no additional hardware, software or training beyond the most basic authentication server and asking a user to register a password initially. The use of passwords, however, has many flaws and is considered by most as a weak authentication mechanism.

Authentication mechanisms are not keeping up with the usage and security requirements of online services. The CERT/CC (Computer Emergency Response Team / Coordination Center), a federally funded organization based at Carnegie Mellon University, has estimated that 80% of network security problems are caused by bad passwords. The problem is that the average user can only remember a maximum of about 7 passwords and these are usually of low complexity (Yan *et al.*, 2004). By low complexity it is meant that they are based on dictionary words that can be guessed or predicted with relatively little effort or are sufficiently short that a brute force attack can be performed very quickly. The human brain is very good at seeing and remembering patterns, but it is notoriously bad at seeing or remembering random information. As the most secure passwords are those that are random, system developers are inevitably fighting a losing battle.

To solve the problems with passwords, Multi-Factor authentication and one-time passcodes are sometimes used. Rather than relying solely on the single factor of what the user knows, e.g. password, it introduces either what the user is, e.g. biometrics, or what the user has, e.g. a token (O'Gorman, 2004). Further, with one-time passcodes any spyware or phishing scam that records entered passcodes is of no value as the next login passcode will be different. This is much more secure, but is still not without its flaws. Two-factor, One-Time Password (OTP) authentication is still susceptible to Man-in-the-Middle (MITM) attacks by just intercepting the OTP and submitting it to the legitimate authentication server. In addition, once the user has logged in to the system, the session can be hijacked and any additional transactions performed without the user's knowledge (Schneier, 2005).

Tokens and biometrics are also not a viable option for e-commerce websites due to the cost and delay caused by delivery of the token. Additionally, the user will have to carry the token with them and may require further software installed on each computer. The near ubiquity of the mobile phone, and the propensity for the user to carry their mobile wherever they go, mean that this is a good device to select for a second factor or token. Solutions using mobile phones are becoming more common from the sending of OTPs via SMS to software OTP tokens to run on devices such as RSA's SecureID software token (RSA, 2011). These, however, suffer the same problems as above and are not immune to being cracked or cloned, with tools able to mimic them freely available for download (OXID, 2011).

Here an architecture is proposed to provide 2-factor authentication for websites by using an encrypted 2D barcode to transfer an encrypted challenge, which is used to produce a one-time passcode on a known, registered mobile device. Simply displaying an image on the screen and asking a user to select something from it, or having additional challenge questions, does not constitute 2-factor authentication, as it still relies solely on what the user knows. Therefore, in this solution, a registered mobile device that has a unique code and a private key on it must be present.

Tanaka *et al.* (2007) showed that users rate using QR code-based authentication schemes as being almost as easy as fixed passwords to use and nearly twice as easy as OTP tokens. The perceived security is also higher than OTP tokens. When asked to rate 'unforgettableness', users rated the mobile phone based, QR code system as being nearly twice as memorable as either fixed passwords or OTP tokens, which can be left behind. Overall user responses were positive in all areas, other than lower speed of authentication, than using fixed passwords.

2. Matrix (2D) Barcodes

A matrix code, also known as a 2D barcode, is a two-dimensional way of representing information that can be scanned and read into a computer reliably and quickly. They are a logical progression from the standard linear barcode that appears on practically every product package on the market currently. These matrix barcodes are essentially a whole series of linear barcodes stacked on top of each other. Comparing the linear barcode, which is one dimensional, with the matrix barcode, there is the capability to store more data whilst maintaining the speed and reliably.

There are several different types of 2D-barcode that have been developed for different purposes. The most common ones are DataMatrix, PDF417, Ezcode, Semacode, and QR Code. The Quick Response (QR) Code was created by the Japanese Corporation Denso-Wave in 1994, an example of which can be seen in Figure 1 below. These matrix codes were initially used for tracking parts in vehicle manufacturing, but they have been repurposed in recent years. They are now used both in commercial tracking applications and convenience-oriented applications specially aimed at mobile phone users. QR Codes are already very popular in Japan for advertising and information purposes. They are frequently found in places such as magazines, business cards, advertising posters and signs.

Figure 1: Example QR Code encoding the title of this paper

Due to the nature of these codes, they are able to contain almost any information such as URLs, names, addresses, telephone numbers, e-mail addresses, etc. These codes then allow the information to be decoded at high speed. In order to read the QR Code from a camera phone, the mobile phone has to have reader software installed on it. This software is programmed to identify different types of information and to launch the appropriate application on the mobile to deal with it, for example the browser to redirect to a URL. Most Japanese and Nokia Mobile phones have this software pre-installed and it is widely available for other platforms (Rohs, 2004).

The QR Code has a maximum data capacity of 7,089 characters for numeric, 4,296 characters for alphanumeric and 2,953 bytes for binary data. The image itself will change size depending on the amount of data stored in it, and has several control points in it to locate and align the image. It is also possible to merge images in the QR Codes, such as logos, characters or even photos. It is, likewise, possible to colour the codes as long as there is still high contrast between the two colours used.

3. Challenge-Handshake Authentication Protocol

The Challenge-Handshake Authentication Protocol (CHAP) is used to authenticate a user or network host to an authenticating entity and is used in the Point-to-Point Protocol. CHAP uses a three-way handshake verified by a shared secret, which is not sent over the network. Instead a random challenge is sent and the handshake is calculated at both ends based on the shared secret. It is this response that constitutes the authentication and is a one-time passcode, as the challenge will change every time. It is for this reason that a similar CHAP authentication is used in the proposed system to produce one-time passcodes for authentication.

The three-way handshake is depicted in Figure 2 below. The first step is for the client to send the User ID in plaintext to the CHAP server. On receiving this, the server will generate a random challenge string. This challenge will be sent in plaintext to the client and will be combined with the known password on the server and the hash taken. At the same time, the client will receive the challenge, combine it with the password and take the hash of the result. This resulting hash is then sent back to the server as the response. If this response matches the hash computed by the server then the client is authenticated.

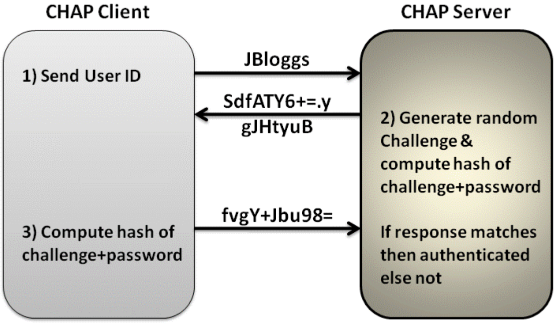

CHAP Client
CHAP Server
1) Send User ID
JBloggs
SdfATY6+=.y
gJHtyuB
2) Generate random Challenge & compute hash of challenge+password
fvgY+Jbu98=
3) Compute hash of challenge+password
If response matches then authenticated else not

Figure 2: CHAP 3-way handshake

Once the authentication is complete, the server will acknowledge the login and the client will have access. However, at random intervals after this, the process will be repeated and further challenges made. The client will have to provide a correct response each time to maintain access.

4. Existing QR Code Authentication Schemes

The idea of using QR Codes for user-friendly authentication is not new – several schemes have been proposed over the past few years. However, they have shortcomings or particular design goals that make them unsuitable for web-based authentication. Some of these proposed systems will be discussed here by way of introduction to the proposed system and design criteria.

Seeing-Is-Believing (SiB) is a system used for verification and transfer of authenticated public-keys and authentication proposed by McCune *et al.* (2009). The system "utilises 2D barcodes and camera-phones to implement a visual channel for authentication and demonstrative identification of devices." The system can be used to obtain keys for wireless access points or to share keys directly between two mobile phones, for example.

The idea behind this is that a user can identify a device or user by physical contact with them. The QR code then becomes a simple way to exchange data once identity has been established. For the scenarios that they suggest, their system works well. However, this does not work across the internet for web-based authentication, where physical proximity is not possible. Simply substituting a new QR code in transit would be sufficient to render this system insecure.

4.1. Out-of-Band Channel

Other QR code authentication systems have been proposed for web-based authentication. One such example is proposed by Mizuno *et al.* (2005). The system uses the mobile phone as a token in the authentication process by displaying a QR code in the user's web browser passing a session-ID and nonce back and forth between the authentication server and the mobile device. The weakness of this system stems from relying on the cellular network to be secure. If the link between the mobile and the authentication servers can be intercepted then the security fails. The mobile phone is identified solely by the ability to send and receive.

An improved system was proposed by Tanaka *et al.* (2007), which sends the unique ID of the mobile device, the user's PIN and the network service provider username to the authentication server in order to lock the authentication down to an individual, unique mobile phone. However, the token, *T*, displayed in the QR code is sent in the clear and used, concatenated with the other data, in the return channel, as follows:

$$T \,||\, h(T \,||\, ID \,||\, PIN \,||\, username) \,||\, username$$

where $h(M)$ is a hash function and || denotes concatenation. *T* and *username* would be known to an attacker and it is possible to find the *ID* as well. In this scenario, if the *ID* were to become known then a simple brute-force attack of the 10,000 possible PIN numbers is feasible. Also, in the case of a collision in *T*, then a simple replay attack is possible.

Again, this system relies on the security of the cellular network assuming that it will be secure. This is not a valid assumption. Whilst UMTS security may currently be proof against practical attacks, GSM is not and frequently phones will have to drop back to GSM. Indeed, it is possible to jam the signals of other services and only leave GSM open. In addition to this, although out-of-band channels can help security, they also hurt usability and availability. If users are out of network coverage or in an environment where the use of wireless transmission is prohibited, then this authentication mechanism fails. It could also put the financial burden on the end user due to data transfer requirements over the cellular network.

4.2. In-band Communication

Schemes have been proposed that only use a single communication channel. The system proposed by (Ion & Dragovic, 2008), highlights a method that could be used to verify transactions from a Point-of-Sale (PoS) device in a shop environment. The PoS device is responsible for encoding the QR code with the data sent from the card, which is as follows:

$$A \,||\, N \,||\, h_K(A \,||\, N)$$

where A is the transaction amount, N is a nonce and h_K is a key-based hash function. The response is the hashed nonce concatenated with the previous hash value. An attacker would have access to these values. The security of this system is, once again, relying on a shared secret alone. Nonce collisions, although unlikely, may cause a problem especially if the amount of the transaction is the same. The nonce is 20-bits and the amount 16-bits in this solution. The response is simply truncated for ease for the user to 20-bits and relies on the system to control the maximum number of attempts at authenticating. Also, the problem is that they only propose a 56-bit hash. This makes collisions possible.

A better solution is proposed in (Starnberger *et al.*, 2009), which follows a similar pattern, but the nonce and amount are not sent in clear. Equally, an explicit accept or reject code are concatenated with the A and N before hashing. The resulting hash is also simply truncated before being sent back to the authentication server. Problems can occur if the truncated hashes for accepted and rejected messages are the same, so an additional *count* parameter is added and incremented until the two messages are different. This can put extra load on the user's device.

Both these systems rely on a shared secret, which is open to attack. They can also both suffer from collisions. If the mobile phone is to be used as an authentication token in a full 2-factor scenario, then the authentication must be locked to a single mobile device. Neither of these solutions identifies the mobile device and any other could be substituted along with the shared secret. Also, no authentication of the user is performed in these systems. Therefore, if an attacker were to gain physical access to the mobile phone, then they would be free to authenticate as the user.

5. Proposed System

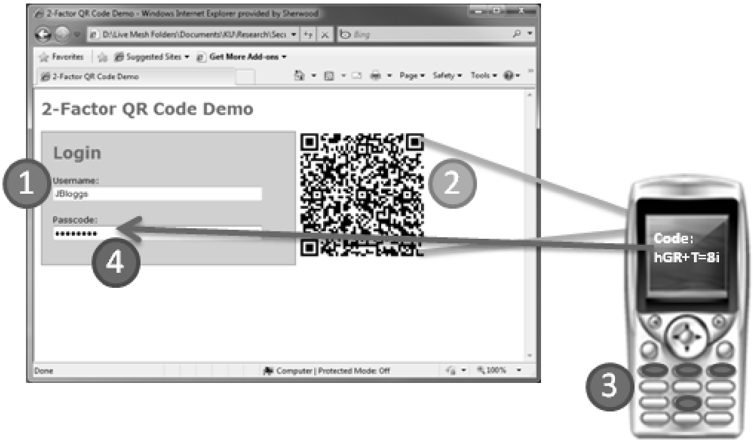
The proposed system uses a similar method to CHAP authentication to produce one-time passcodes for web-based authentication. A registered mobile device is used to read the encrypted challenge from the web page via a QR code; the response is calculated on the mobile and typed into the web page by the user. In addition to the challenge, however, there is temporal data embedded in the QR code, such as a timestamp, to foil replay or delay attacks.

The mobile phone is a trusted device in the system along with the authentication server. However, all other devices and channels are considered to be insecure and are untrusted. The mobile device makes use of the new PKI-SIM to store a private key and perform Elliptic Curve Cryptography (ECC). A PKI-SIM is a SIM/USIM with an integrated public-key crypto-processor and storage space for private keys, similar to that of a smartcard. It provides strong PKI based authentication, encryption and digital signature services.

Some mobile phones have the capacity to hold two smartcards: one for the mobile network and one for mobile electronic identities. This could solve the problems of switching mobile network overseas and changing or upgrading the mobile phone. Whichever is chosen, it allows the user to protect the private keys with a tamperresistant smartcard, but not have to change private key each time they change their phone. They also have the benefit of offloading the cryptography from the device, to enable greater speed for the whole process.

5.1. The Authentication Process

The proposed system has a simple authentication process for the user, which is shown below in Figure 3 using a proof-of-concept implementation. The user enters their username into the web page. The web page then dynamically displays the QR code on the same page. The QR code is then captured in a custom application on the user's mobile device. This application will display certain relevant information, such as the current timestamp, user configured information and even payment transaction details if this is used for payment confirmation. Then the user will be asked for their PIN number, if they confirm the details are correct, and the application will generate an 8-character alpha-numeric response. The user will then type this response code into the web page as their one-time passcode to login.



2-Factor QR Code Demo

Login

Username: JBloggs

Passcode: ••••••••

Code: hGR+T=8i

2-Factor QR Code Demo - Windows Internet Explorer provided by Sherwood
D:\Live Mesh Folders\Documents\KU\Research\Sec
Bing
Favorites
Suggested Sites
Get More Add-ons
2-Factor QR Code Demo
Page
Safety
Tools
Done
Computer | Protected Mode: Off
100%
1
2
3
4

Figure 3: The Authentication Process: 1 – Enter Username; 2 – Read QR Code; 3 – Enter PIN; 4 – copy passcode into web page

This process is intended to be as simple as possible for the user to adapt to in place of a static password, but still have the added security of requiring a physical token in addition to their knowledge of a shared secret. The shared secret has been made easier for the user to compensate for the additional steps and complication. Although, to the user this process is simple, there is a lot of information exchanged. The actual format of the information in the QR code is critical to the system.

5.2. Registered Mobile Device

This system requires a trusted, registered mobile device as the second factor. There are a few important things that need to be considered before implementing this system, which include how to uniquely identify the mobile device, the private key and the user's PIN. The International Mobile Equipment Identity (IMEI) code is a 56-bit number unique to every mobile phone whether it is a GSM, WCDMA, iDEN or satellite phone. It can normally be displayed on the screen for the user to enter into the website registration by typing *#06# into the keypad. Originally, the IMEI number was used to identify valid devices and stop a stolen phone from connecting to the network. Although IMEIs can be read by code on smartphones and can also be changed, for this system this is not thought to be a problem as obtaining that number requires physical access or locally installed malware and it will be combined with other data, so is of little value on its own.

The proposed system uses public key cryptography to encrypt the data in the QR code. A private key is therefore needed to decrypt the message on the mobile device. The proposed system uses a PKI-SIM for this purpose as explained above. This would mean that the mobile device could be used for authentication even when disconnected from the mobile network, e.g. in flight mode, because it works like a small personal computer rather than just a phone.

The final aspect that needs considering is the user's PIN. The PIN is for this system only and has no connection to the phone's PIN. It is stored on the authentication server and nowhere else. It is not stored anywhere on the mobile phone or the SIM card and there is no local checking of the validity of the PIN, so if an attacker were to get hold of the user's phone, they would have no way to obtain the PIN from it. The user would not know if they have entered the correct PIN until their authentication attempt either succeeded or failed as it is only checked on the server. The exact data exchanged will be dealt with in the next section. In addition to this, future mobile devices may be available with biometrics onboard, allowing for the addition of biometric authentication to the mobile device.

5.3. QR Code Format

The general exchange follows a similar pattern to CHAP as described in Section 3 above. On receiving the username from the user, the server will generate a 256-bit random challenge, a 160-bit mask and a 4-bit random number (power). These will be concatenated with the current timestamp, a Time To Live (TTL) and information about the transaction requested or some user-defined data. This last part is significant, as it allows the user to check to make sure that an attacker hasn't injected their own payload. If this technique is used to verify a payment, for instance, the transaction amount and number of items would be added here. If, on the other hand, this were purely for authentication, the user's IP address, userAgent string and custom text could be inserted here. The custom text could work like Mastercard's Secure Code, whereby they ask for a sentence to be registered by the user.

If the correct user data is not displayed on the user's mobile device, then there is a problem and the user should abort the authentication attempt. Similarly, any replay attacks should be foiled by the timestamp and TTL, which can be set to have a very short lifespan of 60 seconds or less.

All the data contained within the QR code is encrypted using the public key of the user and signed by the authentication server, making a man-in-the-middle (MITM) attack significantly harder. The public key codes proposed are Elliptic Curve Cryptography (ECC) codes. The reasoning for this is their efficiency and small key sizes, which make them more suitable for mobile deployment than RSA. The following would therefore be contained in the QR code:

$$E_{ASPriv}(E_{UserPub}(N \,\|\, mask \,\|\, power \,\|\, timestamp \,\|\, TTL \,\|\, user\ data))$$

On the user's device, the QR code is recognised and decrypted, using the authentication server's public key, then the locally stored user's private key. The timestamp, TTL and user data are then checked and verified by the device and the user respectively. Upon verification the response to the challenge can be calculated.

To calculate the response, firstly the user-entered 4-digit PIN number is raised to the power of the 4-bit random number transferred in the QR code, i.e. $\mathrm{PIN}^{\mathrm{power}}$. This will then be expressed as a 200-bit binary number ($\log_2 (9999^{15}) \approx 200$). The 256-bit challenge, 200-bit expanded PIN and 56-bit IMEI number will then be concatenated together to produce a 512-bit pre-response number. The 160-bit SHA1 hash of this pre-response is then taken.

$$h(N \,\|\, PIN^{power} \,\|\, IMEI)$$

However, this 160-bit number is not used as the response, as it would be too long and complex for a user to type in. Instead, the 160-bit random mask created by the server is used to reduce the 160 bits down to 48 bits by simply having 48 1's randomly distributed over the 160 bit mask. These are the indicators to show which 48 bits from the hash value should be returned to the server. In turn these 48 bits are base-64 encoded to produce 8 alphanumeric characters on the mobile's display for the user to type into the web page as a one-time passcode. By using this random mask it reduces the chances of a challenge or power collision causing problems.

The PIN is a factor of the response value, with the challenge, and the IMEI number. Only the correct PIN will produce the correct response value. The same response value will also be calculated by the authentication server. If the response value matches, the user gains authentication to the service, otherwise the authentication fails. If there were a collision of the challenge then it could expose weaknesses and may lead to discovery of the PIN number. In this case, however, the PIN number is protected by not using it directly, but 'expanding' it.

6. Conclusions

A new authentication system is proposed in this paper that combines improved security over passwords with simplicity for the user and low cost for the deployment. It utilises a CHAP-style authentication process using a registered mobile device and a QR code to transfer the challenge from the web page to the mobile. It is considered low cost as it doesn't require the issuing of separate hardware devices as the user would only have to carry their mobile with them, which many users would carry anyway. The system is mobile as it could be use anywhere by the users, even without network connectivity from their mobile device. This authentication mechanism could be used in a wide variety of different remote services such as ecommerce, online banking, Intranet Applications, Virtual Private Networks (VPN), etc. It has a balance of security, ease and cost-efficiency.

By using public-key encryption, the challenge is protected from attackers attempting to authenticate. Similarly, the response has been engineered so that it does not provide any viable information to an attacker. The challenge will change for each authentication procedure, preventing replay attacks. The timestamp and TTL further improve this functionality and will reduce the window for attacking the authentication as well as making the mobile device aware of the time constraint so that it can stop the user from entering any data or attempting the authentication.

It has to be assumed that the attacker has access to the terminal in use for the authentication. To calculate the response, the attacker would need the private key from the PKI-SIM, the IMEI number of the phone and the PIN number of the user. The IMEI requires access to the phone, as does the private key. However, even with both of these, the PIN number is still required from the user. Users will be aware of MITM attacks and injection attacks due to securely sending the transaction or user data to the mobile via the encrypted QR code.

There are limitations to the security of such a system, but no system is absolutely secure. Several problems with other 2-factor schemes were highlighted in the paper. This scheme does overcome many of these to provide a more secure authentication scheme. Critically, it is considered to be significantly more secure than simple username/password combinations, but without the cost and deployment issues of other solutions whilst retaining the simple user experience. Usability is critical to the adoption of security by the users and is often reduced by security solutions. The scheme proposed does not require the user to carry an additional device specifically for authentication or engage in an overly complex interaction. Further work needs to be undertaken to ensure the usability and to fully evaluate it. However, initial very limited trials of the incomplete system have been positive.

7. References

Ion, I. & Dragovic, B. (2008), “Don’t trust POS terminals! Verify in-shop payments with your phone”, *SPMU'08 - Workshop on Security and Privacy Issues in Mobile Phone Use*, Sydney, Australia, May 19, 2008

McCune, J. M., Perrig, A. & Reiter, M. K. (2009), “Seeing-Is-Believing: using camera phones for human-verifiable authentication”, *International Journal of Security and Networks*, Vol. 4, Nos. 1/2, pp. 43-56, 2009

Mizuno, S., Yamada, K., Takahashi, K. (2005), “Authentication Using Multiple Communication Channels”, *DIM’05*, Fairfax, Virginia, USA, ACM 1-59593-232-1/05/0011, November 11, 2005

O'Gorman, L. (2003), "Comparing Passwords, Tokens, and Biometrics for User Authentication" *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021-2040, 2003

OXID (2011), "Caine & Abel", http://www.oxid.it/, (Accessed 25/3/11)

Rohs, M. (2004), "Real-world interaction with camera-phones" *2nd International Symposium on Ubiquitous Computing Systems (UCS 2004)*, Tokyo, Japan, 2004

RSA Security Inc (2011), "Software Authenticators", http://www.rsa.com/node.aspx?id=1313, (Accessed 25/3/11)

Schneier, B. (2005), "Two-factor authentication: too little, too late" *Communications of the ACM - Transforming China*, Volume 48 Issue 4, Page 136, ACM New York, NY, USA, ISSN: 0001-0782 EISSN: 1557-7317, April 2005

Starnberger, G., Froihofer, L. & Goeschka, K. M. (2009), "QR-TAN: Secure Mobile Transaction Authentication", *International Conference on Availability, Reliability and Security*, IEEE Computer Society, pp. 578-583, 2009

Tanaka, M., Teshigawara, Y. (2007), "A Method and Its Usability for User Authentication by Utilizing a Matrix Code Reader on Mobile Phones", *WISA 2006, Lecture Notes in Computer Science 4298*, pp. 225-236, Springer-Verlag, 2007

Yan, J., Blackwell, A., Anderson, R. and Grant, A. (2004), "Password memorability and security: Empirical results" *IEEE Security and Privacy*, vol. 2, no. 5, pp. 25-31, 2004