Quick Response Code Secure: A Cryptographically Secure Anti-Phishing Tool for QR Code Attacks*

Vasileios Mavroeidis[1] and Mathew Nicho[2]

[1] University of Oslo, Norway
vasileim@ifi.uio.no
[2] Robert Gordon University, Scotland
m.nicho1@rgu.ac.uk

Abstract. The two-dimensional quick response (QR) codes can be misleading due to the difficulty in differentiating a genuine QR code from a malicious one. Since the vulnerability is practically part of their design, scanning a malicious QR code can direct the user to cloned malicious sites resulting in revealing sensitive information. In order to evaluate the vulnerabilities and propose subsequent countermeasures, we demonstrate this type of attack through a simulated experiment where a malicious QR code directs a user to a phishing site. For our experiment, we cloned Google's web page providing access to their email service (Gmail). Since the URL is masqueraded into the QR code, the unsuspecting user who opens the URL is directed to the malicious site. Our results proved that hackers could easily leverage QR codes into phishing attack vectors targeted at smartphone users, even bypassing web browsers' safe browsing feature. In addition, the second part of our paper presents adequate countermeasures and introduces QRCS (Quick Response Code Secure). QRCS is a universal efficient and effective solution focusing exclusively on the authenticity of the originator and consequently the integrity of QR code by using digital signatures.

Keywords: quick response (QR) codes, 2D codes, smartphone security, mobile phishing attacks, cryptography, digital signatures

1 Introduction

Quick response (QR) code has become one of the more popular two-dimensional barcodes due to its inherent data capacity and higher damage resistance [1]. With smartphone security and privacy becoming a major concern [2], a hijacked QR code can be a dangerous attack vector for smartphone users. With millions of QR codes displayed by companies in public places its not difficult for

* Proceedings of the 7th International Conference on Mathematical Methods, Models and Architectures for Computer Networks Security (MMM-ACNS 2017), Warsaw, Poland, August 2017. DOI: 10.1007/978-3-319-65127-9_25

a malicious author to replace or modify them. The rapid pace of smartphone adoption and usage [3] has not only enhanced QR code popularity and usage over a much wider range of applications [4], but also introduced newer QR code attack vectors for malicious users [5]; thus, posing a serious threat to unsuspecting smartphone users. With mobile phone usage crossing the two billion mark in 2016, outnumbering personal computers [6], this threat assumes greater significance. Compared to the infamous 376 bytes Slammer worm that destroyed millions of computers in the year 2003 [4], the maximum binary data that a QR code can hold is roughly 2.9KB which can be a threat vector for malicious payloads. QR codes can be used in several different types of attacks such as social engineering and automated processes attacks. Automated processes attacks are executed by exploiting SQL injection vulnerabilities, command injection, as well as Cross-Site Scripting (XSS) attacks [7].

In this article, we ethically simulate a QR code phishing attack to demonstrate the attack methodology and the bypass method that can be employed by hackers to counter safe browsing. In addition, we propose Quick Response Code Secure (QRCS), a secure framework that makes use of digitally signed QR codes to verify their genuineness.

Our paper is structured as follows. In Section 2, we explain smart phone vulnerabilities with respect to QR codes. Section 3 provides an overview of QR codes and how it has been leveraged as an attack vector. Thereafter, Section 4 details the simulated attack using malicious QR codes, which demonstrates how it can be leveraged as a phishing attack vector including the identified vulnerable points. In Section 5, we present our QRCS solution which details the components and the subsequent process used to authenticate genuine QR codes by users. Section 6 provides a conclusion alongwith future directions.

2 Smartphone Vulnerability

Phishing is a difficult unresolved problem [8]. Since phishing attacks can be made more convincing on smartphones than on a desktop browser, users can expect to see more phishing attacks on mobile malware in the future [8]. Attackers create different kinds of malware to exploit devices for financial gain, utilization of resources, information and data theft, search engine optimization, spam messages, access private networks, or even damaging devices for amusement. Generally, such malware are attached to popular legitimate applications or in new applications having some functionality to trick the user.

Mobile users put mobile devices in their everyday life using them for several reasons like making phone calls, sending text messages and emails, online transactions, accessing social networks, accessing corporate data, portable storage, saving information (notes) and many more. These make mobile devices a valuable asset to target for malicious attacks. Moreover, in US only 14% of the mobile users have antivirus installed while 34% of the users dont use any form of technical protection, let alone the four-digit screen lock PIN [9]. In addition, the lack of security in legitimate mobile applications makes them vulnerable to traditional attacks like SQL injection, cross site scripting (XSS), and man in the middle attacks (MITM).

Several smartphone antivirus applications do not handle malware adequately due to the limitations imposed by Android's security system [10]. Hence, not all antivirus applications are effective at preventing malware and spyware from infecting an Android phone [11]. Furthermore, most of the antivirus software for smartphone devices are signature-based; thus, making it impossible to protect the devices from zero day attacks and sophisticated (complex) polymorphic malware. The subsequent sections explain QR codes and how it can be leveraged by phishers, the simulated attack we conducted, the vulnerabilities we detected, subsequent countermeasures, and our proposed solution to secure QR codes.

3 QR Codes as Attack Vector

QR codes, developed by a subsidiary of Toyota named Denso Wave in Japan in 1994, consist of black square dots arranged in a square pattern (matrix code) on a white background. It can store 7,089 numeric characters or 4,296 alphanumeric characters, 2,953 bytes of binary, and 1,817 Japanese Kanji/Kana symbols [12]. QR code has six desirable features, namely high capacity encoding of data, small printout size, Chinese/Japanese (kanji and kana) capability, dirt and damage resistance, readability from any direction in 360°, and a structure append feature [13].

3.1 Leveraging QR Codes for Malicious Purpose

QR codes are being increasingly used as an attack vector facilitating phishing attacks and redirecting users to malicious websites that host malware [5]. Even though people might fall for QR code leveraged phishing attacks there are other possible weaknesses that QR codes can possibly exploit [7]. Specifically, QR codes can be used for SQL injection and command injection attacks in automated readers-programs that extract information from QR-codes. Moreover, depending upon the type of data recognized and the nature of the application, the decoding of QR codes can result in a phone number being automatically dialed, a short text message being sent, a web page corresponding to the decoded URL displayed in a mobile browser, or a different application executed [14]. In addition, the Unstructured Supplementary Service Data (USSD) codes encoded in 2D barcodes can be used to wipe a phone, execute other system functions, generate premium rate SMS messages, trigger vulnerabilities in the reader software, the operating system, or a remote site, such as SQL injections [15].

3.2 Leveraging QR Codes for Phishing

The most prevalent attacks employing QR codes are phishing and drive by downloads [3] that trick users to disclose confidential information. QR codes enable users to open web pages via scanning, without typing the URL. Furthermore, to improve usability some browsers in mobile devices hide the URL and even if the browser shows the URL, due to limitations in smartphone screen size users won't be able to notice it clearly; thus, making QR codes a very attractive vector for phishers. In addition, some attackers use URL shortening mechanisms to hide the true URL for tricking naive users. Regarding the possibility of exploited sites, attackers can also direct users to malicious sites that can initiate 'drive by downloads' based on the fingerprint of the device. As the fingerprint information is included in the HTTP header when a user requests a site (GET Request) [16], the exploit kit retrieves the information about the device and sends the relevant malicious code.

QR codes have been misused as attack vectors by social engineers via encoded malicious links that enable phishing sites to execute fraudulent codes [17]. Kharraz, Kirda, Robertson, Balzarotti and Francillon conducted an empirical analysis across 14 million web pages to discover the extent to which QR codes are leveraged by attackers in the wild. Their results showed that QR codes are being abused by attackers to distribute malware or direct to phishing sites on the public web [3]. The results not only revealed the malicious use of QR codes, but also identified 145 malicious QR codes out of 94,770 QR codes. However, this experiment was limited to QR codes found only on the Web and not in public places. To demonstrate that QR codes can be used for conducting phishing attacks, Vidas, Owusu, Wang, Zeng, Cranor and Christin [18] deployed QR code posters across 139 different location where they found that most users (75%) scanned the QR code out of curiosity or for fun, with very few scanned to solicit more information within the context surrounding the QR code. The results indicate that most users who scan a QR code will subsequently visit the related URL, even if the domain is unfamiliar and uses shortened style URL. A similar experiment was conducted where the researcher placed QR coded stickers in high traffic areas around a target town. When users scanned the QR code, they were redirected to a WordPress site that informed them about the experiment,and the dangers that QR codes can hide. [19].

However, our research not only simulates leveraging QR codes for phishing, but also proposes a cryptographic methodology to counter the attack vector.

4 Experiment

In this part, we demonstrate and analyse how QR codes can be used in phishing attacks making them a potent attack vector. Particularly, we start by creating a malicious QR code and a phishing Gmail page, followed by the demonstration of the attack.Finally, we bypass Google safe browsing to keep the malicious QR codes alive even in case they are flagged malicious.

To support our experiment, we captured a photo of a QR code at a bus stop in a city in United Kingdom (Figure 1) that we use in a way to simulate a phishing attack. We can clearly see that the bus stop has an NFC chip, and a QR code attached for commuters to check the time table electronically. In this regard, attackers do not need any special-purpose tools to launch a phishing attack neither sophisticated methods to trick unaware commuters to scan the malicious QR codes.

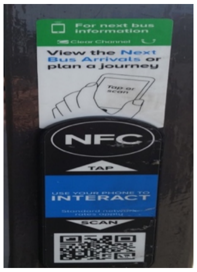

For next bus information
View the Next Bus Arrivals or plan a journey
Tap or scan
NFC
TAP
USE YOUR PHONE TO INTERACT
SCAN

Fig. 1. QR code at a bus stop in UK

4.1 Create a QR Code

The first step in our experiment was to create a QR code with a malicious URL attached directing the user to a phishing site.

The requirement of a successful phishing attack is a domain name similar to the original website for deceiving users including a link management platform to hide (masquerade) our domain name. The latter is a common method to deceive users in case a domain name is not similar to the original to keep the malicious QR codes operational. In our simulation we used Bitly, a link-management platform service allowing users to shorten a URL making it more attractive and practical. Bitly uses the following format; http://bit.ly/.... Attackers normally use such services as a masquerading method to trick people to visit the malicious site. Since the shortened link replaces the original, users are unable to find out the destination without first visiting the website.

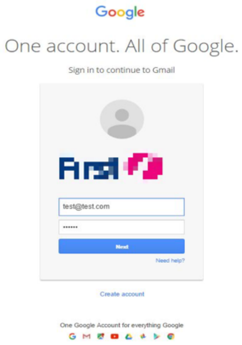
Furthermore, some QR code readers allow users to see the human readable QR code content (e.g. URL) before performing the action, while other code readers redirect without this intermediate action. We have to make clear that this is not a flaw of the mobile device but a limitation of the application software itself. To demonstrate the attack we cloned and modified Google's Gmail page.

4.2 Create a Phishing Gmail Page

In this scenario, we copied the official Gmail page people use to access their email accounts. Furthermore, we deployed a web server to host our phishing website and a database to store the credentials retrieved. In addition, we added a descriptive logo of the public transport service company in the city where the experiment was conducted (Figure 2 - with name of the bus company blurred for anonymity).

4.3 Launch the Attack

We simulated the attack on ourselves (unlike a normal QR code phishing attack scenario conducted at a public place). The browser will open and direct the user to the phishing site. The unsuspecting user adds the requested credentials and when taps the subsequent button will be redirected to the official Gmail page without noticing anything suspicious. The simulated credentials are stored into our created database.



Google

One account. All of Google.

Sign in to continue to Gmail

test@test.com

••••••

Fig. 2. Phishing Gmail Page

4.4 Bypassing Google Safe Browsing

Google has web crawlers searching for malicious sites making the job of attackers difficult, or not that difficult. Two days later, our page was flagged as deceptive by "Google safe browsing" feature. We hereby demonstrate how with only one line of code, even this countermeasure can be bypassed.

The bypass method that can be employed by hackers is to change the name of the phishing page. However, in reality, it is not convenient because the attacker would have to change the URL in the malicious QR code. The warning pop-up window alerts that "This web page at (URL) has been reported as deceptive and has been blocked based on your security preferences" which actually means that Google flagged only the actual phishing page (file) as malicious, and not the domain name. This practice helps web developers to identify the compromised page; otherwise, the domain name would be flagged malicious. Worthy to note is that it is not considered best practice to blacklist domain names because malicious domain names today may be benign after a period of time. In reality the phishing page should be the first page someone sees when directed to the site. A malicious actor can automatically redirect a victim from a blank main page to the malicious phishing page with one line of PHP. Therefore, we use the main page (for example index.php) only to redirect users to a phishing subpage. The victim will still see it as the main page because of the following PHP code (header(”location: test.azurewebsites.net/ test.php”)) which redirects the victim automatically to the phishing page without Google’s warning message popping up anymore.

5 Secure QR Code Solution

This section of the paper deals with the security of QR codes where our universal QR Code Secure tool is presented which ensures that users get redirected only to the intended sites. Since QRCS authenticates the originator by using digital signatures our solution is a safe way to certify that the QR code a user scan is genuine and not tampered.

5.1 Related Work-Security of QR Codes

Chuang, Hu and Ko [20] proposed a technique that improves data security in data transmission during QR code communication. The secret sharing technique divides the ”secret” data (QR code) into shadows (several QR codes) that are distributed to ”n” participants where some parts or all of them are needed to reconstruct the message. However, this secret sharing technique for QR codes focuses primarily on confidentiality of data rather than prevent phishing.

Gao, Kulkarni, Ranavat, Chang and Mei [21] proposed a 2D barcode-based mobile payment system, which uses QR codes to conduct secure and reliable payment transactions using mobile devices. While this solution secures QR code-based transactions it doesnt audit the integrity of the initial QR code the user scans.

In addition, Narayanan [22] describes several non-technical solutions to raise awareness of the security of QR codes that should not be overlooked. Interesting is his suggestion of using descriptive features, such as logos into the QR codes that would raise the difficulty to design similar QR codes, as well as the use distinctive colors. Our opinion regarding the former is that this solution should be avoided as hackers or even everyday users can use online tools to add logos or change the colors of the traditional QR codes easily. As a result, a QR code with distinctive properties can be used to easier trick people believing that this code is coming from a legitimate source. Regarding the use of colors, Krombholz, Frhwirt, Kieseberg, Kapsalis, Huber and Weippl [17] believe that the more complex the color theme of a QR code is , the harder is to replicate it. Additionally, they proposed the extension of these complex color themes to the whole advertising campaign hosting the QR code. Futhermore, they point out the relevance of digital signatures to verify the originator of a QR code.

Two encryption schemes for QR codes have been proposed that make use of symmetric and asymmetric cryptography respectively [23]. For symmetric QR codes the use of a shared secret key between the reader and the writer is used with AES as the recommended cryptographic algorithm. The asymmetric solution makes use of the RSA algorithm to encrypt the symmetric key that can be appended next to the message. It is worth noting that Peng et al. [24] proposed the use of digital signatures for verification of the source before any action is performed. Their idea to use a number of bits to identify the signers public key is included in QRCS. However, the solution occupies a considerable amount of space in a QR code that makes it appropriate only for specific purposes. In this regard, we emphasize that the RSA algorithm may not be an efficient algorithm that can be used in mobile devices. Hence, to minimize the computational overhead, we suggest to use only small public keys (exponent e).

Accordingly, we differentiate our research by creating a universal solution that can be used for the integrity and authenticity of any QR code for information or transaction purposes.

5.2 Secure QR Code Solution (QRCS)

Our proposed solution adopts the traditional server-client architecture using digital signatures. The server side controls and authenticates the entities that wish to access our platform to create QR codes under their company's profile (can be verified by the users that scan the codes), while the client is an application available (like the traditional QR readers) for public use. Our proposed solution makes use of the popular hash functions and digital signatures to provide integrity and authenticity to the QR codes.

Hash functions: QRCS makes use of the cryptographically secure hash function SHA-2 (SHA-3 can be used too) to generate a digest message of 256 bit, which is currently resistant to brute force attacks. A hash function or better a hash algorithm is a mechanism that can be fed with arbitrary length input to generate a fixed length output. This property is not only significant for the use of digital signatures if the plaintext is large enough, but also overcomes storage limitations in the QR code.

Digital signatures: Our solution makes use of digital signatures, which is a way to prove that a certain entity generated the message we observe, such as the plaintext into the QR code. The cryptographic primitive is achieved through the public key cryptography which makes use of the difficulty to factorize large prime numbers or the discrete logarithm problem. Since smartphones have limited resources for doing heavy calculations we propose the ECDSA (elliptic curve digital signature algorithm) for signing the hash functions. The entity that generated the plaintext and the hash will digitally sign the hash by making use of a private key and an ephemeral key (the ephemeral key is different every time we want to sign a new QR code). The solution we propose does the aforementioned automatically to generate the QR code. Consequently, a user that scans the code needs a reader specialized in decoding, decrypting, and verifying the signature.

Server-Side Platform On the server side we propose a platform that can host our certificate authority (to generate digital certificates) and the registration system. Since companies, organizations, or even small businesses can apply for an approved profile on our platform, to illustrate the process we use a fictional company named ABC. Company ABC can get access to our platform by applying and submitting the requested documents that actually prove their claimed identity. The approved entity will get authorized access, a unique digital certificate, and can generate digitally signed QR codes.

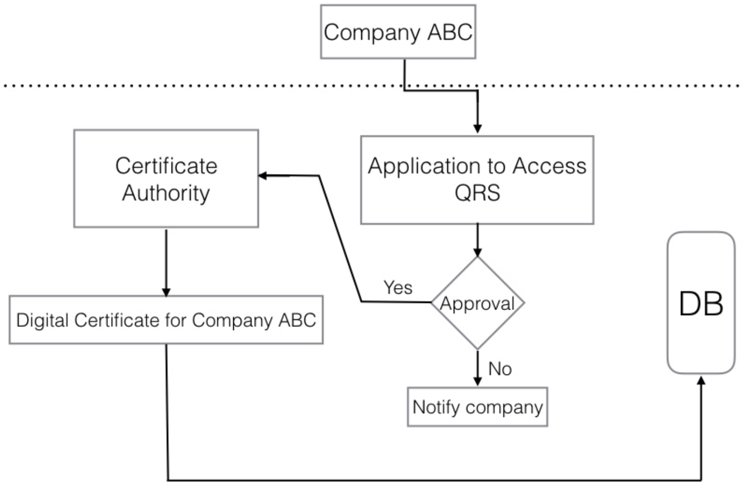


Fig. 3. Server-Side Platform (QRCS)

In summary, a company ABC has to include only the plaintext needed to be encoded and QRCS will calculate the hash function and digitally sign it (encryption with the private key and a unique ephemeral key). In addition, we make use of Peng et al. [24] concept of using a number of bits to uniquely identify the signer's public key. As a result, when a code is scanned, the application identifies the public key that is used to verify whether the QR code is genuine or not.

Client-Side Application Once we have a QR code created by QRCS it contains the plaintext, the unique number which specifies the public key of the signer, and the digitally signed hash. Figure 4 presents the client-side QRCS.

The application can access the public keys of the entities online stored into our database, or they can be downloaded in the application for offline access in case a network connection is not available. When a code is scanned the application will check with the use of the public key and the signature if the verification condition is satisfied. If it is satisfied, then the application will perform the intended action; otherwise, the QR code is classified as non-verified. In case of a non-verified QR code, the application will alert the user and the QR code will be blocked from action. In addition, the user is able to submit a report for the non-genuine QR code with details.

This option is available only for registered users who have proved their identity with a OTP (one time password) sent to their submitted mobile number when they first registered. For example, if the QR code is available on the Internet, then the user can submit the URL targeting the non-verified code. The user has also the option to specify the geographic location of the non-genuine code as well as the organization that the QR code claims to originate and are completely optional. These options are available only after a code has been flagged as non-verified. Furthermore, a user can use the application without registering.

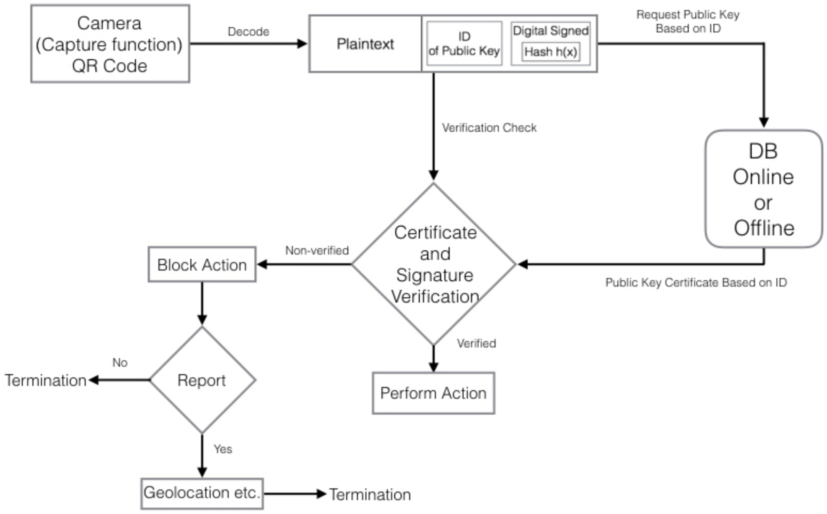


Fig. 4. Client Software (QRCS)

6 Conclusion

In this paper, we went through the process commonly deployed by hackers to leverage QR codes for phishing attacks and subsequently proposed a client-server based solution using digital signatures to authenticate benign QR codes from malicious ones. Considering the rate at which QR codes are being deployed by the industry targeting the mobile phone users, we believe that hackers are increasingly leveraging QR codes as attack vectors putting companies and users at risk. In this regard, our simulated attack using a compromised QR code revealed vulnerabilities that can lead to unintentional disclosure of sensitive personal information. Subsequently, we conceptually demonstrated our client-server-based QRCS cryptographic solution using digital signatures that can successfully thwart malicious QR code attacks at the initial scanning phase itself, by digitally verifying the genuine QR code. Security analysis of our model shows that apart from preventing users from redirection to malicious sites by the malicious QR codes, our model is also effective against man in the middle and replay attacks as well. This model is straightforward to implement as it requires less modification from a QR code deployment perspective.

Our future work focuses on making QRCS convenient to use for both companies and everyday users by separating the application in security levels needed by the interested party. Unlike companies, everyday users normally wont go through a formal checking process. In this regard, our future work targets the lower trust level zone in the application layer where user access is provided through a unique email, username, password, and mobile number.

References

1. Lin, P.Y., Chen, Y.H.: High Payload Secret Hiding Technology for QR Codes. EURASIP Journal on Image and Video Processing (2017)
2. Zhou, Y., Jiang, X.: Dissecting Android Malware: Characterization and Evolution. In: Security and Privacy (SP), 2012 IEEE Symposium on, IEEE (2012) 95–109
3. Kharraz, A., Kirda, E., Robertson, W., Balzarotti, D., Francillon, A.: Optical Delusions: A Study of Malicious QR Codes in the Wild. In: Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on, IEEE (2014) 192–203
4. Sharma, V.: A Study of Malicious QR Codes. International Journal of Computational Intelligence and Information Security **3**(5) (2012) 21–26
5. Jain, A.K., Shanbhag, D.: Addressing Security and Privacy Risks in Mobile Applications. IT Professional **14**(5) (2012) 28–33
6. Chaffey, D.: Mobile Marketing Statistics Compilation. http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/
7. Kieseberg, P., Leithner, M., Mulazzani, M., Munroe, L., Schrittwieser, S., Sinha, M., Weippl, E.: QR Code Security. In: Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia, ACM (2010) 430–435
8. Felt, A.P., Finifter, M., Chin, E., Hanna, S., Wagner, D.: A Survey of Mobile Malware in the Wild. In: Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, ACM (2011) 3–14
9. Tapellini, D.: Smart Phone Thefts Rose to 3.1 Million in 2013 Industry Solution Falls Short, while Legislative Efforts to Curb Theft Continue. http://www.consumerreports.org/cro/news/2014/04/smart-phone-thefts-rose-to-3-1-million-last-year/index.htm
10. Fedler, R., Schütte, J., Kulicke, M.: On the Effectiveness of Malware Protection on Android. Fraunhofer AISEC **45** (2013)
11. Ramachandran, R., Oh, T., Stackpole, W.: Android Anti-Virus Analysis. In: Annual symposium on information assurance & secure knowledge management, Citeseer (2012) 35–40
12. Rouillard, J.: Contextual QR codes. In: Computing in the Global Information Technology, 2008. ICCGI'08. The Third International Multi-Conference on, IEEE (2008) 50–55
13. Chen, W.Y., Wang, J.W.: Nested Image Steganography Scheme Using QR-Barcode Technique. Optical Engineering **48**(5) (2009) 057004–057004

14. Liao, K.C., Lee, W.H.: A Novel User Authentication Scheme Based on QR-Code. JNW **5**(8) (2010) 937–941
15. Dabrowski, A., Krombholz, K., Ullrich, J., Weippl, E.R.: QR Inception: Barcode-in-Barcode Attacks. In: Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices, ACM (2014) 3–10
16. Penning, N., Hoffman, M., Nikolai, J., Wang, Y.: Mobile Malware Security Challeges and Cloud-Based Detection. In: Collaboration Technologies and Systems (CTS), 2014 International Conference on, IEEE (2014) 181–188
17. Krombholz, K., Frühwirt, P., Kieseberg, P., Kapsalis, I., Huber, M., Weippl, E.: QR Code Security: A Survey of Attacks and Challenges for Usable Security. In: International Conference on Human Aspects of Information Security, Privacy, and Trust, Springer (2014) 79–90
18. Vidas, T., Owusu, E., Wang, S., Zeng, C., Cranor, L.F., Christin, N.: QRishing: The Susceptibility of Smartphone Users to QR Code Phishing Attacks. In: International Conference on Financial Cryptography and Data Security, Springer (2013) 52–69
19. Deborah, M.: Security Expert Warns Smartphone Users of the Risks in Scanning Cybercoding. http://www.post-gazette.com/business/businessnews/2012/06/01/Security-expert-warns-smartphone-users-of-the-risks-in-scanning-cybercoding/stories/201206010228
20. Chuang, J.C., Hu, Y.C., Ko, H.J.: A Novel Secret Sharing Technique Using QR Code. International Journal of Image Processing (IJIP) **4**(5) (2010) 468–475
21. Gao, J., Kulkarni, V., Ranavat, H., Chang, L., Mei, H.: A 2D Barcode-Based Mobile Payment System. In: Multimedia and Ubiquitous Engineering, 2009. MUE'09. Third International Conference on, IEEE (2009) 320–329
22. Narayanan, A.S.: QR Codes and Security Solutions. International Journal of Computer Science and Telecommunications **3**(7) (2012) 69–71
23. Paar, C., Pelzl, J.: Understanding Cryptography: A Textbook for Students and Practitioners. Springer Science & Business Media (2009)
24. Peng, K., Sanabria, H., Wu, D., Zhu, C.: Security Overview of QR Codes. Student project in the MIT course 6.857,'14 (2014)