1-1-2016

Exploiting the error correction mechanism in QR codes for secret sharing

Yang-Wai Chow
University of Wollongong, caseyc@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Guomin Yang
University of Wollongong, gyang@uow.edu.au

James Phillips
Auckland University of Technology, jim.phillips@monash.edu

Ilung Pranata
University of Newcastle

See next page for additional authors

Exploiting the error correction mechanism in QR codes for secret sharing

Abstract

This paper investigates a novel approach to secret sharing using QR codes. The proposed QR code secret sharing approach exploits the error correction mechanism inherent in the QR code structure, to distribute and encode information about a secret message in a number of shares. Each share in the scheme is constructed from a cover QR code, and each share itself is a valid QR code which can be scanned and decoded by a QR code reader. The secret message can be recovered by combining the information contained in the QR code shares. Since each share is a valid QR code, the proposed scheme has the advantage of reducing the likelihood of attracting the attention of potential attackers. In addition, the shares can be distributed via public channels without raising suspicion. Moreover, shares do not have to be stored or transmitted electronically, as QR codes can be distributed via printed media.

Keywords

secret, sharing, codes, exploiting, error, qr, mechanism, correction

Disciplines

Engineering | Science and Technology Studies

Publication Details

Chow, Y., Susilo, W., Yang, G., Phillips, J. G., Pranata, I. & Barmawi, A. (2016). Exploiting the error correction mechanism in QR codes for secret sharing. Lecture Notes in Computer Science, 9722 409-425. Melbourne, Australia Information Security and Privacy: 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part I

Authors

Yang-Wai Chow, Willy Susilo, Guomin Yang, James Phillips, Ilung Pranata, and Ari Moesriami Barmawi

Exploiting the Error Correction Mechanism in QR Codes for Secret Sharing

Yang-Wai Chow[1], Willy Susilo[1], Guomin Yang[1], James G. Phillips[2], Ilung Pranata[3] and Ari Moesriami Barmawi[4]

[1] Centre for Computer and Information Security Research, School of Computing and Information Technology, University of Wollongong, Australia
{caseyc, wsusilo, gyang}@uow.edu.au,
[2] Department of Psychology, Auckland University of Technology, New Zealand
jphillip@aut.ac.nz,
[3] School of Design, Communication and Information Technology, University of Newcastle, Australia
ilung.pranata@newcastle.edu.au,
[4] School of Computing, Telkom University, Indonesia
mbarmawi@melsa.net.id

Abstract. This paper investigates a novel approach to secret sharing using QR codes. The proposed QR code secret sharing approach exploits the error correction mechanism inherent in the QR code structure, to distribute and encode information about a secret message in a number of shares. Each share in the scheme is constructed from a cover QR code, and each share itself is a valid QR code which can be scanned and decoded by a QR code reader. The secret message can be recovered by combining the information contained in the QR code shares. Since each share is a valid QR code, the proposed scheme has the advantage of reducing the likelihood of attracting the attention of potential attackers. In addition, the shares can be distributed via public channels without raising suspicion. Moreover, shares do not have to be stored or transmitted electronically, as QR codes can be distributed via printed media.

Keywords: Error correction, QR code, secret sharing

1 Introduction

A secret sharing scheme refers to a method by which a dealer encodes a secret into a number of shares and distributes these shares to a group of participants. Individually, the shares reveal no information about the secret. The secret can only be reconstructed when information from an authorized number of shares is combined [2]. By splitting and encoding a secret into a number of shares, secret sharing overcomes the problem of storing a secret in a single information-carrier, which can easily be lost or damaged [19, 21]. Secret sharing schemes are important tools that have found many applications in cryptography and distributed computing [2]. In addition, secret sharing is also regarded as a mechanism that

can be used to transfer secret information via public channels in cryptography [25]. This paper investigates a novel approach to secret sharing by distributing and encoding a secret into a set of Quick Response (QR) codes.

The concept of secret sharing was first introduced independently by Blakley [3] and Shamir [18]. The secret sharing schemes that they proposed are known as k-out-of-n, or (k, n), threshold schemes. In a (k, n) secret sharing scheme, a secret is divided into n shares, where $n>1$, and k, or more, shares are required to reconstruct the secret. Even complete knowledge of any $k-1$, or fewer, shares will reveal no information about the secret. Since its inception, many different secret sharing schemes have been proposed [1, 2, 8, 7, 16, 17, 19–22, 25]. Some schemes require complex numerical computation, whilst others require little or no computation [21]. Of the varying approaches, image sharing can be seen as a subset of the general secret sharing problem as the secret is a concealed image [22]. One of the popular ways of secret image sharing is known as visual cryptography [17]. Visual cryptography is a method of encoding and distributing a binary image into a number of shares, each to be printed on separate transparencies. When the qualified number of shares are stacked together, the human visual system can recover the secret image without the need for any computation.

Each share in visual cryptography looks like a random pattern of pixels. In an extension to visual cryptography, known as extended visual cryptography, each share is encoded using a meaningful cover image [1]. This means that when viewed individually, a meaningful, albeit noisy, image is visible on each share. The advantage of encoding the secret image into shares containing 'innocent-looking' meaningful cover images is that it reduces the likelihood of attracting the attention of attackers [20].

While the advantage of these visual secret sharing schemes is that decryption can be performed by the human visual system without any computation, these approaches suffer from a number of problems, including pixel expansion, contrast issues, share misalignment and the visual quality of the reconstructed image [4, 8, 16, 26]. Secret image sharing techniques have also been proposed to share and hide a secret image by distributing and embedding the information required to reconstruct the secret image in a number of digital images, each of which is a meaningful cover image [19]. However, these techniques hide information in digital images. Therefore, decryption must be performed on the digital image shares, which necessitates that the shares must be distributed via electronic means.

Our Contribution. This paper introduces a novel approach to secret sharing by distributing and encoding a secret message into a number of QR code shares. The proposed approach exploits QR code error correction redundancy, which is an inherent feature of the QR code structure. The advantage of this approach is that each share is a meaningful QR code, which individually does not reveal the secret message. The secret message can be recovered by combining the information contained in the QR code shares. Since each QR code share can be scanned and decoded by any standard QR code reader, this means that the shares can be

distributed via public channels without raising suspicion. In addition, since QR codes are meant to be scanned by a QR code reader, the shares do not have to be stored or transmitted electronically and can be distributed via printed media. Furthermore, the shares can be constructed using any artistic QR code method as long as it can be scanned and read. Therefore, each QR code share can be constructed using a different artistic QR code scheme in order to increase the secret sharing subterfuge by reducing the likelihood of attracting the attention of potential attackers.

2 Related Work

The QR code is a two-dimensional code that was invented by the company Denso Wave [10]. In recent years, QR codes have seen widespread adoption due to its convenience and ease of use, as any smartphone equipped with a camera and QR code reader can retrieve the information encoded within a QR code. The application of the QR code in the area of information security has previously been proposed for a number of different purposes. For example, QR codes have been used for authenticating visual cryptography shares [23], e-voting authentication [11] and for digital watermarking [14].

QR codes have also been used in the area of data hiding and steganography. Among the work conducted in this area, Wu et al. [24] proposed a data embedding approach for hiding a QR code in a digital image. Their purpose was to camouflage the appearance of a QR code in an image so as not to degrade the visual quality of the picture. Huang et al. [12] developed a reversible data hiding approach for images with QR codes. The purpose of their method was to avoid a QR code from degrading the quality of the image or concealing information contained in the image. In their approach, using reversible data hiding a portion of an image is hidden in the rest of the image. This portion is replaced with a QR code. Once the QR code has been scanned, it will be removed from the image and the original image will be restored using the data that was previously hidden in the rest of the image. A nested image steganography scheme was proposed by Chen and Wang [6] using QR codes. In their approach, two types of secret data, in the form of text (lossless) and image (lossy), are embedded into a cover image. The text portion of the secret data is embedded using a QR code. A similar approach was also reported in Chung et al. [9].

Unlike approaches that first convert a secret into a QR code before embedding it into a cover image, Lin et al. [15] introduced a scheme for concealing secret data in a cover QR code. Their approach capitalizes on the error correction property of a QR code in order to conceal secret data. The amount of secret data that can be concealed depends on the QR code version and error correction level that is used. Bui et al. [5] also investigated the problem of hiding secret information in a QR code. They argue that previous approaches to embedding secret messages in QR codes use bit embedding which is vulnerable to modification attacks. Consequently, they propose a method of using Reed-Solomon code and list decoding to hide a secret message in a QR code.

3 Background

The International Organization for Standardization (ISO) has established a standard for the QR code (ISO/IEC18004) [13]. This section outlines the basic QR code structure and error correction feature as defined by the ISO standard.[1]

3.1 The QR Code Structure

A QR code symbol is constructed as a two-dimensional array of light and dark squares, referred to as modules. There are forty sizes of QR code symbol versions ranging from version 1 to version 40. Each QR code symbol version is comprised of a different number of modules, and as such different QR code versions give rise to different data capacities. Version 1 is made up of 21×21 modules, and each successive version increases by 4 additional modules per side, up to version 40 which is made up of 177×177 modules. The appropriate version to use depends on the amount and the type of data (alphanumeric, binary, Kanji or a combination of these) to be encoded as well as the error correction level. The error correction level will be described in Section 3.2 to follow.

The QR code structure consists of encoding regions and function patterns [13]. An example of the encoding regions and function patterns for a QR code version 7 symbol is illustrated in Fig. 1. As can be seen from the figure, a QR code symbol is surrounded by an empty region known as the quiet zone, which should have the same reflectance value as the light modules. Function patterns do not encode data, they consist of the finder patterns, separators, timing patterns and alignment patterns. There are three identical finder patterns located at each corner, except the bottom right corner, of the symbol. These are used by a QR code reader to recognize the QR code and to determine the rotational orientation of the symbol. The separators are one module wide, and are constructed from light modules to separate the finder patterns from the encoding region. Timing patterns are alternating light and dark modules used to determine module coordinates in the symbol. Alignment patterns allow QR code readers to compensate for image distortion. Different QR code versions have a different number of alignment patterns.

3.2 Encoding and Error Correction

The encoding region consists of data codewords and error correction codewords, format information and version information (version information is only used in QR code version 7 and above). Message data is encoded as a bit stream which is divided into a sequence of codewords. All codewords are 8-bits in length. The codewords are divided into a number of error correction blocks, based on the QR code version and error correction level, and an appropriate number of error correction codewords are generated for each block. Error correction allows

[1] For a comprehensive description of the QR code structure and error correction mechanism, please refer to the ISO standard (ISO/IEC18004) [13].

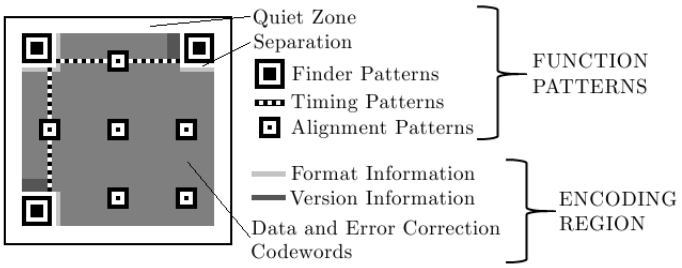

Fig. 1. QR code version 7 structure.

correct decoding of the message in the event that part of the symbol is dirty or damaged. This error correction feature has also been exploited to embed art or other information in QR code symbols.

The QR code employs Reed-Solomon error control coding for error detection and correction [13]. There are four error correction levels that the user can select. Each level provides a different error correction capacity (L $\sim$ 7%, M $\sim$ 15%, Q $\sim$ 25% and H $\sim$ 30%). Higher error correction levels improve the recovery capacity, but also increases the amount of data to be encoded. This means that if the same message were to be encoded using a higher error correction level, a larger QR code version may be required. The number of data codewords, error correction blocks and error correction codewords depend on the QR code version and error correction level. Table 1 shows these characteristics for QR code version 4 and 5. In the table, the error correction codewords for each block is given as (c, k, r), where c is the total number of codewords, k is the number of data codewords and r is the error correction capacity. Note that some QR code versions have blocks with different (c, k, r) values for certain error correction levels. For example, it can be seen in Table 1 that QR code version 5 with an error correction level of Q has a total of 4 error correction blocks. The (c, k, r) values for the first 2 blocks are (33, 15, 9) while the values for the next 2 blocks are (34, 16, 9).

The codewords from the blocks are encoded in an interleaved manner, with the error correction codewords appended to the end of the data codeword sequence. This is done to minimize the possibility that localized damage will cause the QR code to become undecodable. Fig. 2 shows the data codeword and error correction codeword arrangement for QR code version 4, with an error correction level of H. After encoding the codewords, a data mask is applied to the encoding region. There are a total of eight data mask patterns. The purpose of the data mask is to balance the light and dark modules, as well as to minimize the occurrence of undesirable patterns that may potentially confuse a QR code reader. Note that the data mask is not applied to the function patterns.

The format information is a 15-bit sequence consisting of 5 data bits and 10 error correction bits. Two copies of this information are encoded in a QR code symbol, as can be seen in Fig. 1. Of the 5 data bits, the first 2-bits indicate the error correction level and the next 3-bits indicate the data mask pattern used in

Table 1. Error correction characteristics for QR code version 4 and 5 [13].

Version	Total codewords	Error correction level	Number of blocks	Error correction codewords per block[2] (c, k, r)
4	100	L	1	(100, 80, 10)
		M	2	(50, 32, 9)
		Q	2	(50, 24, 13)
		H	4	(25, 9, 8)
5	134	L	1	(134, 108, 13)
		M	2	(67, 43, 12)
		Q	2	(33, 15, 9)
			2	(34, 16, 9)
		H	2	(33, 11, 11)
			2	(34, 12, 11)

the QR code symbol. Error correction for the format information is performed using the Bose-Chaudhuri-Hocquenghem (15, 5) code, which allows for an error of 3-bits, or less, to be corrected. The 15-bit long data bit string is XORed with a specific mask to ensure that the format information bit sequence does not contain all zeros for any combination of data mask pattern and error correction level.

Version information is only contained in QR code version 7 and above. It consists of an 18-bit sequence consisting of 6 data bits and 12 Golay error correction bits. The (18, 6) Golay code allows up to 3-bit errors to be corrected. Its purpose is to convey version information to a QR code reader, for example, for QR code version 7 the 6-bit data string is 000111. QR code version 7 and above contain two copies of this information, as depicted in Fig. 1.

4 Proposed QR Code Secret Sharing (QRCSS) Scheme

The error correction mechanism in the QR code makes it possible to manipulate some of the codewords, while still maintaining a QR code symbol that can be correctly decoded. The approach proposed in this paper is an (n, n) secret sharing scheme which will be referred to as QR Code Secret Sharing (QRCSS). The idea behind QRCSS is to exploit the error correction redundancy in the QR code structure, in order to take a QR code containing a secret message and to distribute and encode its information into n QR code shares. Each share is a valid and decodable QR code in themselves, which are each constructed from a meaningful cover QR code. The secret message can be recovered by first XORing the light and dark modules contained in the encoding region of the n QR code shares and adding the function patterns. This will produce a resulting QR code that when decoded will reveal the secret message. In the proposed scheme, the cover

[2] c = total number of codewords, k = number of data codewords, r = error correction capacity

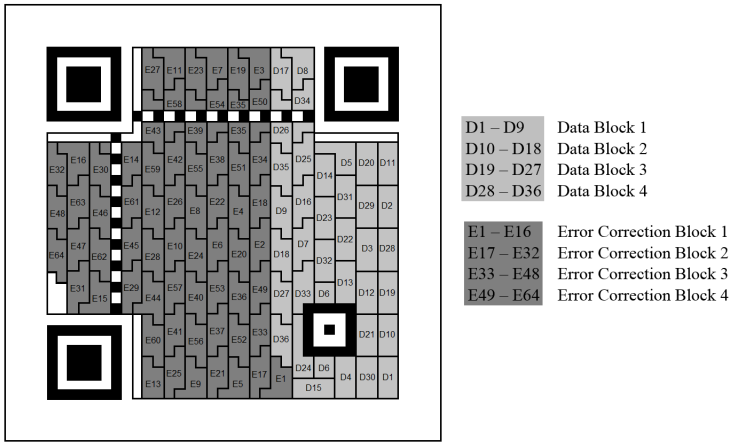

Fig. 2. Data and error correction codeword arrangement for QR code version 4 with error correction level H.

QR codes and the secret QR code must be of the same QR code version with the same error correction level. This is so that all the codewords and codeword blocks are at exactly the same locations across all the QR code symbols.

4.1 Number of Shares

An indication of the minimum number of QR code shares, n, required to share a secret QR code can be determined using the approximate recovery capacity previously discussed in Section 3.2. For example, to share a secret QR code using the error correction level H (i.e. 30%), the minimum number of QR code shares required is 3. This is because up to ∼30%[3] of the codewords per block in each QR code share can be manipulated, which allows up to $\sim 30\% \times 3 \approx 90\%$ of the codewords in the secret QR code to be distributed across the QR code shares. This means that the error in the reconstructed secret QR code is at least ∼10%. As long as the error in the reconstructed secret QR code is no more than its error correction capacity, which is ∼30%, the secret can be recovered. It can easily be seen that it is not possible to share the secret using only 2 QR code shares, as the reconstructed secret QR code will contain a minimum of ∼40% error, which overwhelms the error correction capacity and hence cannot be decoded correctly. This also means that the greater the number of shares used to share a secret, the smaller the amount of error that has to be introduced in each QR code share. An indication of the minimum number of shares required for the proposed QRCSS scheme for the respective error correction level is as follows: L ($n \geq 14$), M ($n \geq 6$), Q ($n \geq 4$), and H ($n \geq 3$).

[3] In practice, it is not advisable to introduce this much error in the QR code shares, because any damage or dirt may make the QR code symbol undecodable.

4.2 Codewords

The QR code error correction levels of L $\sim$ 7%, M $\sim$ 15%, Q $\sim$ 25% and H $\sim$ 30%, are merely approximate values. The exact error correction capacity depends on the QR code characteristics, which varies between the different QR code versions and error correction levels, as can be seen from the characteristics for QR code version 4 and 5 that were previously shown in Table 1.

The error correction capacity per block, r, for a QR code represents the maximum number of codewords that can be in error per block. In other words, a QR code cannot be decoded if more than r codewords per block contain errors. This means that to split the secret QR code across the QR code shares, the maximum number of codewords that can be changed across the QR code shares per block is $r \times n$. However, since c is the total number of codewords per block, there is no point changing more codewords than c, even in the case where $r \times n$ is greater than c. Hence, the maximum number of codewords that may be changed should be the smaller of the two values. This will be referred to as m, where $m = min(r \times n, c)$.

In order for a QR code to be correctly decoded, there must be a minimum number of codewords per block that cannot contain any errors. This value will be referred to as l, where $l = c - r$. This also means that in the reconstructed secret QR code, each block must have l, or more, correct codewords in order to decode the reconstructed QR code and to reveal the secret message.

The proposed QRCSS scheme requires that $m > l$ for each block. Otherwise, it will not be possible to split information required to reconstruct the secret QR code across the shares, while maintaining valid QR code shares. This also means that the number of codewords that have to be altered per block, t, can lie anywhere between l and m. Nevertheless, in practice it is best to evenly distribute the error among the shares and the reconstructed secret. Given that l is the minimum number of codewords per block that have to be changed across the shares, to evenly distribute the error among the shares, at least $\frac{l}{n}$ codewords per block have to be changed for each share. However, using this value would mean that the reconstructed secret has absolutely no error tolerance. As such, in the proposed QRCSS scheme, the error correction capacity for each block of the reconstructed secret was taken as the ratio between the total error correction capacity of the shares and minimum number of codewords that had to be changed, i.e. $\frac{r \times n}{l}$ rounded to the nearest integer. Hence, to balance between the error correction capacity of the reconstructed secret and the number of errors that have to be introduced in the shares, $t = min(l + round(\frac{r \times n}{l}), m)$, since t ranges between l and m. In general, the larger the value of n, the larger the error correction capacity of the reconstructed secret, for cases where $t < m$. If $t = m$, the error correction capacity of the reconstructed secret QR code is equal to the error correction capacity of the original secret QR code, i.e. r.

Let e be the maximum number of codewords in a block to change per share, or in other words the maximum number of errors that will be introduced to a block per share in order to distribute the secret. To evenly distribute the error across the shares, the value of e should be $\frac{t}{n}$ rounded up to the nearest integer,

i.e. $e = round\_up(\frac{t}{n})$. This means that the larger the value of n, the smaller the amount of error per share e. So while the original cover QR code error capacity per block is r, the resulting error capacity per block for each QR code share in this QRCSS scheme is $r - e$. For the reconstructed secret QR code, since t codewords per block were altered in the shares, $c - t$ codewords per block will be in error when reconstructed. So the error correction capacity per block in the reconstructed secret QR code is $r - (c - t)$.

4.3 Format and Version

The format information for a QR code contains 15-bits and allows for an error of 3-bits, or less. The first 2-bits in the bit sequence represent the error correction level, which in the proposed QRCSS scheme must be the same for all the shares and the secret. Hence, these 2-bits do not have to be altered as only the next 13-bits may vary between the individual shares and the secret. The format information allows for an error of up to 3-bits in each share to be corrected. In addition, the reconstructed secret itself allows for a 3-bit format information error to be corrected. As such, to evenly distribute the error across the shares, the maximum number of format information bits to change per share f, should be $\frac{13}{n+1}$ rounded up to the nearest integer, i.e. if $n = 3$ then $f = 3$, otherwise $f = round\_up(\frac{13}{n+1})$.

QR code version 7 and above have additional version information. However since the proposed QRCSS scheme uses QR codes of the same version, the version information modules are the same for all QR code shares. As such, the unaltered version information simply has to be added to the reconstructed secret QR code.

In the proposed scheme, the choice of QR code version used to share a secret has a direct bearing on its security. The security of the QRCSS scheme is governed by a security parameter λ. The size of λ depends on the data capacity of a QR code. The more data a QR code can contain, the larger λ will be. Hence, the QR code version has to be taken into consideration when determining security of the scheme. This is because the larger the QR code version, the higher its data capacity.

4.4 Algorithm

The proposed (n, n) QRCSS scheme can formally be described as follows. The algorithm takes one secret QR code, S, and n cover QR codes, C_1, C_2, ..., C_n, as input and outputs n QR code shares, S_1, S_2, ..., S_n. The minimum required value of n for each error correction level was previously discussed in Section 4.1. Each QR code share S_1 to S_n is a valid QR code that when decoded by a QR code reader will produce the same information as their respective cover QR codes C_1 to C_n. Pseudocode for the proposed QRCSS scheme is presented in Algorithm 1. In should be noted that the input QR codes can be generated using any standard QR code generator.

To recover the secret, XOR the modules in the encoding regions of all the shares $S_1 \oplus S_2 \oplus ... \oplus S_n$ and add the function patterns to produce a reconstructed

QR code, S^r. When decoded by a QR code reader, S^r will produce the same message as S.

5 Analysis and Discussion

A secret sharing scheme can be evaluated based on a number of properties, including its security, reconstruction precision, computation complexity and storage requirement [21]. This section presents an analysis and discussion of the QRCSS scheme based on these properties.

A program implementing the proposed (n, n) QRCSS scheme, using the pseudocode shown in Algorithm 1, was developed. An example of the results produced by the QRCSS scheme is depicted in Fig. 3 and Fig. 4[4]. The example shown in these figures is for the case where $n = 3$, using QR code version 4 with error correction level H. Fig. 3(a)-(c) show the three cover QR codes, C_1, C_2 and C_3, while Fig. 4(a) shows the original QR code containing a secret message, S, which is to be encoded into the shares. Fig. 3(d)-(f) show the QR code shares resulting from the proposed QRCSS scheme, i.e. S_1, S_2 and S_3, which were obtained from their respective covers. Note that these shares are valid QR codes that can be read by any standard QR code reader. Fig. 4(b) in turn shows the reconstructed QR code, S^r, which was obtained by XORing the modules in the encoding regions of all the shares, $S_1 \oplus S_2 \oplus S_3$, and appending the function patterns. The secret message can be recovered by decoding S^r using a QR code reader.

A description of the characteristics for the example results shown in Fig. 3 and Fig. 4 is as follows. The codewords in QR codes of version 4 and error correction level H are divided into 4 error correction blocks, with the characteristics of $c = 25$, $k = 9$ and $r = 8$ for all blocks (refer to Table 1). Using the proposed algorithm, $l = 17$ and $m = 24$, while $t = 20$ and $e = 7$ for each block. As such, a maximum of 7 codewords per block were altered in S_1, S_2 and S_3 (i.e. a total of 20 codewords were changed per block, so two of the shares had 7 codewords altered while the remaining share had 6 codewords altered). This means that each block in S_1, S_2 and S_3 has an error correction capacity of at least 1 and each block in S^r has an error correction capacity of 3. Hence, even if there are some errors introduced to S^r, due to damage or other reasons, the secret message can still be recovered.

Fig. 3(g)-(i) show the respective difference images between the cover QR codes and the resulting QR code shares, while Fig. 4(c) show the differences between the secret QR code and the reconstructed secret QR code. In the difference images, gray represent no change, while white represents a change in the reflectance value from a dark module in the cover QR code to a light module in the QR code share, and conversely black represents a change from a light module in the cover to a dark module in the resulting share. The difference images

[4] Please note that all QR codes presented in this paper are valid QR codes that can be decoded by a QR code reader.

Algorithm 1 QRCSS pseudocode

function GenerateShares()
Input: A secret QR code, S, and n cover QR codes, C_1, C_2, ..., C_n

/* Create n shares and copy contents of each cover to the respective share */
for i from 1 to n **do**
Create S_i
$S_i \leftarrow C_i$
end for

/* Data and error correction codewords */
for each QR code block **do** /* Some QR code versions have 2 different block sizes */
Calculate l and m
if $m \leq l$ **then**
Quit /* Scheme requires $m > l$, otherwise input is not valid */
else
Determine t and e
for i from 1 to t **do**
Randomly select a codeword, w, within the block that has not previously been changed
Select a random share, S_x, where $1 \leq x \leq n$, that has not exceeded the value of e

for each module j in w **do** /* where $1 \leq j \leq 8$ (a codeword contains 8 modules) */
if $S_1^j \oplus S_2^j \oplus ... \oplus S_n^j \neq S^j$ **then**
/* S_k^j denotes the j-th module of w in the k-th share */
/* S^j denotes the j-th module of w in the secret */
Flip the reflectance value of S_x^j
end if
end for
end for
end if
end for

/* Format information */
Determine f /* The maximum number of format information bits to change per share */

for j from 3 to 15 **do** /* For each module of the format information, except the first 2-bits */
if $S_1^j \oplus S_2^j \oplus ... \oplus S_n^j \neq S^j$ **then**
Select a random share, S_x, where $1 \leq x \leq n$, that has not exceeded the value of f
Flip the reflectance value of S_x^j
end if
end for

Output S_1, S_2, ..., S_n
end function

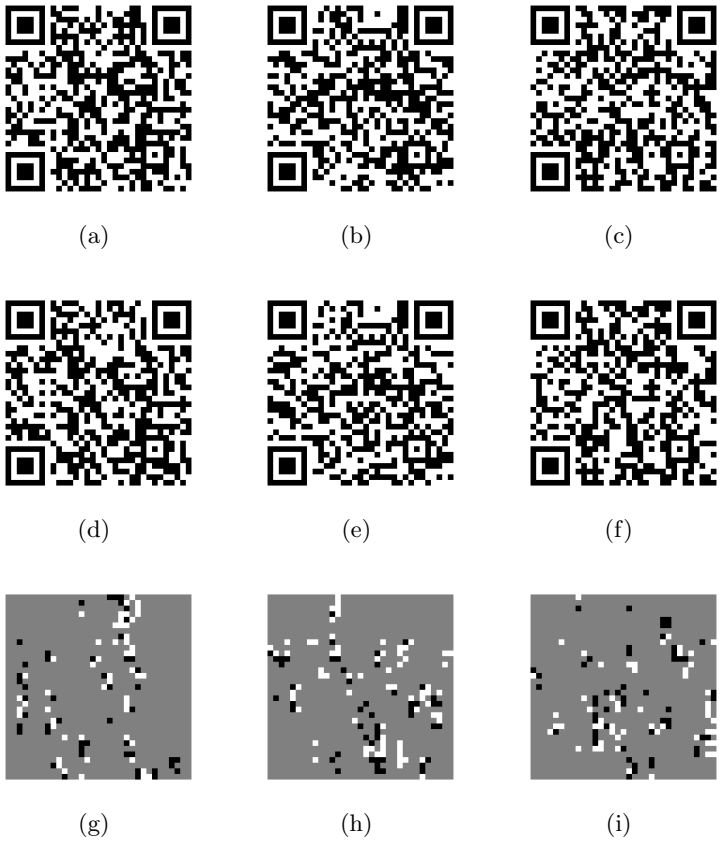

Fig. 3. Example of a (3, 3) QRCSS; (a) C_1, (b) C_2, (c) C_3, (d) S_1, (e) S_2, (f) S_3, (g) Difference image between C_1 and S_1, (h) Difference image between C_2 and S_2, (i) Difference image between C_3 and S_3.

Fig. 4. Secret and reconstructed secret for the example shown in Fig 3. (a) Original QR code containing the secret message, S. (b) Reconstructed secret QR code, S^r. (c) Difference image between S and S^r.

indicate the error that was deliberately embedded into the QR codes for the purpose of sharing the secret.

5.1 Properties

From a computational complexity point of view, the QRCSS scheme is not computationally complex, as encoding and decoding in the scheme is based on the simple Boolean XOR operation. In addition, despite the deliberate error incorporated in the shares by the proposed scheme, the scheme is still able handle a small amount of error when decoding the reconstructed secret QR code. The greater the number of shares used to distribute the secret, the smaller the amount of error that has to be encoded into each of the QR code shares. This will also increase the recoverability of the secret, because the reconstructed secret QR code will contain less errors.

Since the secret is encoded as a QR code, the size of the secret message is governed by the QR code data capacity. The size of of each share in the proposed scheme is the same as the size of the secret QR code. One of the advantages of the QRCSS scheme is that the shares do not have to be stored in digital form, as they can be distributed on printed media.

5.2 Security

Individual QR code shares by themselves do not reveal the secret message. However, since shares and the secret QR code use the same QR code version and error correction level, an adversary who can identify a QR code share will have this knowledge. In addition, given any QR code share, it is trivial to obtain its cover QR code. This can be done by simply decoding the information in the QR code share, then using this information to generate a QR code of the appropriate version and error correction level. This will produce the original cover QR code. As such, it is easy to find the differences between the original cover QR code and the modified QR code. This property also makes it possible to identify potential QR code shares. Based on the information about the differences between the original and the modified QR code, the value of n can be inferred. Nevertheless, it is not possible to reconstruct the secret with this information alone.

Brute Force Attack. In light of the fact that in QRCSS the secret message is encoded as a QR code, an adversary can adopt a brute force strategy akin to a dictionary attack. This is because the secret must be in the form of a valid QR code. Thus, if an attacker has information about the version and error correction level of the secret QR code, the attacker can go through all the possibilities of valid QR codes with the same version and error correction level. Let S' denote a valid, or in other words 'meaningful', QR code of the known version and error correction level, and $|S'|$ be the cardinality of all the valid QR codes with the same version and error correction level. The probability of success for this attack will be bounded by $\frac{1}{|S'|}$. The size of $|S'|$ is governed by the size of data that a

QR code can contain, which is determined by the specific QR code version used to encode the message. Hence, the larger the secret QR code, the larger $|S'|$ will be, which in turn lowers the success of an attack. Let d be the number of data codewords for a QR code of a given version and error correction level. Since each codeword contains 8 modules, $|S'| = 2^{8d}$.

Collusion Attack. It is conceivable that $n-1$ participants may collude and combine the information from their shares together in an attempt to find S.

Since $S_1 \oplus S_2 \oplus ... \oplus S_n = S^r$, in a collusion attack where $n-1$ participants combine their shares, they will have S_1, S_2, ..., S_{n-1}. Let S'_n denote a possible S_n, or in other words the attacker's 'guess' of S_n. Furthermore, let S'^r denote a valid QR code that when decoded will give the same information as S', and let $\mathbf{S'^r}$ represent the set of all valid QR codes created by modifying the modules of all QR codes in $\mathbf{S'}$. This is in line with the fact that the modules in the reconstructed secret QR code, S^r, are not the same as the secret QR code, S. Hence, even though when decoded S'^r and S' will give the same information, the modules in S'^r and S' are not the same. This means that the entries in $\mathbf{S'^r}$ must be created by going through all the possible modifications of all QR codes in $\mathbf{S'}$ that will produce valid QR codes. Therefore, the cardinality of $\mathbf{S'^r}$, $|S'^r|$, will be very much larger than $|S'|$.

$|S'^r|$ can be determined based on the specific QR code version and error correction level. Each block of a QR code has c codewords and an error capacity of r codewords. Each codeword contains 8 modules. This means that a block can correctly be decoded as long as there are r, or less, codewords in c that are in error. However, as previously shown in Table 1, some QR code versions have 2 different block characteristics, B_1 and B_2. Let c_1 and c_2 denote the number of codewords per block, and let r_1 and r_2 represent the error correction capacity per block, for blocks of type B_1 and B_2 respectively. In addition, let n_1 and n_2 represent the number of blocks of type B_1 and B_2 for that specific QR code. For a QR code of a given version and error correction level, $|S'^r|$ can be determined as follows:

$$|S'^r| = \left[[(\sum_{i=0}^{r_1} \binom{c_1}{i}).2^8]^{n_1} . [(\sum_{j=0}^{r_2} \binom{c_2}{j}).2^8]^{n_2} \right] . |S'|$$

A collusion attack may take the following form. By going through the entries in $\mathbf{S'^r}$ using $S_1 \oplus S_2 \oplus ... \oplus S_{n-1} \oplus S'^r = S'_n$, if S'_n is not a valid QR code, i.e. cannot be decoded, then S'^r can be removed from $\mathbf{S'^r}$. Thus, reducing the space of potential reconstructed secret QR codes, $|S'^r|$, to contain less possibilities. Nevertheless, it should be noted that this space will still be very large as $|S'^r|$ is very much larger than $|S'|$. Let $|S''|$ denote this reduced space. It is obvious that the value of $|S''|$ will increase for the cases of $n-2$, $n-3$, etc.

Security Parameter. The security underlying the QRCSS scheme is therefore governed by the security parameter $\lambda = min(|S'|, |S''|)$. Clearly, larger QR code versions will increase the size of λ, thereby increasing the security of the scheme.

Concealment. One of the primary advantages of this scheme stems from the fact that since each share is a meaningful QR code in itself, this will reduce the likelihood of attracting the attention of potential attackers. In addition, since the modules in QR codes do not have to be black and white squares, it would be advantageous if each QR code share were to be constructed using different artistic QR code schemes. The QRCSS scheme will work as long as the modules in each share can be scanned. The use of different artistic schemes will not only increase the secret sharing subterfuge by using meaningful innocent-looking QR code shares, but will also improve concealment as shares will appear to be unrelated.

6 Conclusion and Future Work

This paper presents a novel approach to secret sharing using QR codes. In QR code secret sharing, a QR code containing a secret message is distributed and encoded into a set of meaningful QR code shares. The proposed approach uses the error correction feature, which is an inherent part of the QR code structure, to distributed and hide information about the secret. Each share is a valid QR code which contains meaningful information when scanned individually. Hence, this reduces the likelihood of attracting the attention of potential attackers when distributed via public channels. When all shares are made available, the secret message can be recovered. Unlike a number of other secret sharing schemes where the shares are digital images, which have to be stored and transmitted electronically, the QR code shares in this approach can be distributed using printed media.

References

1. G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson. Extended capabilities for visual cryptography. *Theor. Comput. Sci.*, 250(1-2):143–161, Jan. 2001.
2. A. Beimel. Secret-sharing schemes: A survey. In *IWCC*, pages 11–46, 2011.
3. G. Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pages 313–317, 1979.
4. C. Blundo, P. D'Arco, A. D. Santis, and D. R. Stinson. Contrast optimal threshold visual cryptography schemes. *SIAM J. Discrete Math.*, 16(2):224–261, 2003.
5. T. V. Bui, N. K. Vu, T. T. Nguyen, I. Echizen, and T. D. Nguyen. Robust message hiding for qr code. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2014 Tenth International Conference on*, pages 520–523. IEEE, 2014.
6. W.-Y. Chen and J.-W. Wang. Nested image steganography scheme using qr-barcode technique. *Optical Engineering*, 48(5):057004–057004, 2009.
7. Y. Chow, W. Susilo, M. H. Au, and A. M. Barmawi. A visual one-time password authentication scheme using mobile devices. In *Information and Communications Security (ICICS'14)*, volume 8958 of *Lecture Notes in Computer Science*, pages 243–257. Springer, 2015.
8. Y.-W. Chow, W. Susilo, and D. S. Wong. Enhancing the perceived visual quality of a size invariant visual cryptography scheme. In *Information and Communications*

Security (ICICS'12), volume 7618 of *Lecture Notes in Computer Science*, pages 10–21. Springer, 2012.
9. C.-H. Chung, W.-Y. Chen, and C.-M. Tu. Image hidden technique using qr-barcode. In *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on*, pages 522–525. IEEE, 2009.
10. Denso Wave Incorporated. QRcode.com, http://www.qrcode.com/en/.
11. S. Falkner, P. Kieseberg, D. Simos, C. Traxler, and E. Weippl. E-voting authentication with qr-codes. In *Human Aspects of Information Security, Privacy, and Trust*, volume 8533 of *Lecture Notes in Computer Science*, pages 149–159. Springer, 2014.
12. H.-C. Huang, F.-C. Chang, and W.-C. Fang. Reversible data hiding with histogram-based difference expansion for qr code applications. *Consumer Electronics, IEEE Transactions on*, 57(2):779–787, 2011.
13. International Organization for Standardization. Information technology — automatic identification and data capture techniques — qr code 2005 bar code symbology specification. ISO/IEC 18004:2006, 2006.
14. H.-C. Lee, C.-R. Dong, and T.-M. Lin. Digital watermarking based on jnd model and qr code features. In *Advances in Intelligent Systems and Applications-Volume 2*, pages 141–148. Springer, 2013.
15. P.-Y. Lin, Y.-H. Chen, E. J.-L. Lu, and P.-J. Chen. Secret hiding mechanism using qr barcode. In *Signal-Image Technology & Internet-Based Systems (SITIS), 2013 International Conference on*, pages 22–25. IEEE, 2013.
16. F. Liu, T. Guo, C. K. Wu, and L. Qian. Improving the visual quality of size invariant visual cryptography scheme. *J. Visual Communication and Image Representation*, 23(2):331–342, 2012.
17. M. Naor and A. Shamir. Visual cryptography. In A. D. Santis, editor, *EUROCRYPT*, volume 950 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 1994.
18. A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
19. C.-C. Thien and J.-C. Lin. Secret image sharing. *Computers & Graphics*, 26(5):765–770, 2002.
20. D. Wang, F. Yi, and X. Li. On general construction for extended visual cryptography schemes. *Pattern Recognition*, 42(11):3071–3082, 2009.
21. D. Wang, L. Zhang, N. Ma, and X. Li. Two secret sharing schemes based on boolean operations. *Pattern Recognition*, 40(10):2776 – 2785, 2007.
22. J. Weir and W. Yan. A comprehensive study of visual cryptography. *T. Data Hiding and Multimedia Security*, 5:70–105, 2010.
23. J. Weir and W. Yan. Authenticating visual cryptography shares using 2d barcodes. In *IWDW*, volume 7128 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 2011.
24. W.-C. Wu, Z.-W. Lin, and W.-T. Wong. Application of qr-code steganography using data embedding technique. In *Information Technology Convergence*, pages 597–605. Springer, 2013.
25. W. Yan, J. Wier, and M. S. Kankanhalli. Image secret sharing. In S. Cimato and C.-N. Yang, editors, *Visual Cryptography and Secret Image Sharing*, pages 381–402. CRC Press, Taylor and Francis Group, 2012.
26. C.-N. Yang, A.-G. Peng, and T.-S. Chen. Mtvss: (m)isalignment (t)olerant (v)isual (s)ecret (s)haring on resolving alignment difficulty. *Signal Processing*, 89(8):1602–1624, 2009.