XOR-ed visual secret sharing scheme with robust and meaningful shadows based on QR codes

Longdan Tan¹ · Yuliang Lu¹ · Xuehu Yan¹ · Lintao Liu¹ · Xuan Zhou¹

Received: 7 December 2018 / Revised: 11 August 2019 / Accepted: 9 October 2019 / Published online: 9 December 2019
© The Author(s) 2019

Abstract
Quick response (QR) codes are becoming increasingly popular in various areas of life due to the advantages of the error correction capacity, the ability to be scanned quickly and the capacity to contain meaningful content. The distribution of dark and light modules of a QR code looks random, but the content of a code can be decoded by a standard QR reader. Thus, a QR code is often used in combination with visual secret sharing (VSS) to generate meaningful shadows. There may be some losses in the process of distribution and preservation of the shadows. To recover secret images with high quality, it is necessary to consider the scheme’s robustness. However, few studies examine robustness of VSS combined with QR codes. In this paper, we propose a robust (k, n)-threshold XOR-ed VSS (XVSS) scheme based on a QR code with the error correction ability. Compared with OR-ed VSS (OVSS), XVSS can recover the secret image losslessly, and the amount of computation needed is low. Since the standard QR encoder does not check if the padding codewords are correct during the encoding phase, we replace padding codewords by initial shadows shared from the secret image using XVSS to generate QR code shadows. As a result, the shadows can be decoded normally, and their error correction abilities are preserved. Once all the shadows have been collected, the secret image can be recovered losslessly. More importantly, if some conventional image attacks, including rotation, JPEG compression, Gaussian noise, salt-and-pepper noise, cropping, resizing, and even the addition of camera and screen noises are performed on the shadows, the secret image can still be recovered. The experimental results and comparisons demonstrate the effectiveness of our scheme.

Keywords Quick response code · XOR-ed visual secret sharing · Padding codewords · Robust

1 Introduction
A (k, n)-threshold visual secret sharing (VSS), also known as visual cryptography (VC), is a technology that shares a secret image as n different noise-like shadows, i.e., shares or
shadow images. It was first proposed by Noar and Shamir [9]. The shadows are distributed to \(n\) different participants, and \(k\) or more participants can reveal the secret image by superimposing their own shadows. In contrast, fewer than \(k\) participants cannot gain any insight into the original secret image. Stacking qualified shadows to reveal the secret image without computations or any cryptographic knowledge is the advantage of VSS. However, the pixel expansion problem and codebook (basic matrices) design are its inevitable defects.

Probabilistic VSS by Ito et al. [7], proposed to solve the pixel expansion problem, shared a secret image by randomly selecting columns from the basic matrix. Yang [22] proposed different thresholds for probabilistic VSS. A generalized probabilistic VSS was extended further by Cimato et al. [4].

In the VSS scheme based on a random grid (RG) [18, 19, 21] proposed by Kafri and Keren, the pixel expansion problem can be avoided, and designing a codebook is unnecessary. When the secret image is shared, \(n\) noise-like shadows of the same size as the secret image can be generated. The secret image can be reconstructed by stacking the shadows in OR-ed VSS (OVSS) based on RG. However, when more and more shadows are stacked to determine the secret image, the background of the reconstructed secret image will become darker.

In XOR-ed VSS (XVSS) [11], XOR operations (addition modulo two) on the shadows are used to reveal the secret image; the amount of computation needed is low. This approach can avoid the disadvantage of OVSS based on RG [16]. By applying XVSS, better image contrast and resolution can be obtained [12, 17]. The ability of extensively used lightweight computing devices, such as mobile phones and smart devices, to easily perform the XOR operation has led to more applications of XVSS.

QR code as a machine-readable symbol was invented by the Denso Wave Company of Japan. Such codes have been used increasingly widely in the realm of mobile commerce, e-commerce, and social networks to implement mobile payments, deliver product information, perform identification, etc. The codes can be decoded quickly with a standard QR code reader.

A QR code is a binary image with a distribution of dark and light modules that looks random, and data of various formats can be encoded in QR codes. Compared with one-dimensional codes, QR codes have higher capacity and error tolerance. These strengths have led to several studies exploring the combinations of VSS and QR codes in recent years.

Weir et al. [15] applied QR code to authenticating the shadows. By inserting QR code into the optimum location of given shadows, Wang et al. [14] developed a scheme to prevent cheating. Chow et al. [3] designed an \((n, n)\)-threshold secret sharing scheme, in which \(n\) is no less than 3, by relying on QR code’s error correction mechanism. Yang et al. [23] combined VCS and 2D barcodes to perform authentication on smartphones. A scheme with two-level information management based QR codes being machine readable was proposed by Liu et al. [8]; however, decoding shadows was difficult unless the appropriate scanning angle and distance were determined. Wan et al. [13] continuously embedded a secret image into QR codes by integrating the theory of VSS and the error correction mechanism of QR codes. Chen et al. [1] presented a scheme using QR codes for applications involving security and flexible access structures. Chow et al. [2] used cryptographic keys to encrypt QR codes when embedding them into cover QR codes. Fu et al. [5] proposed a VSS scheme utilizing the encoding redundancy of QR codes by using a probabilistic sharing method.

In traditional VSS, the shadows are noise-like and are not robust. When we distribute noise-like shadows in various ways, they may attract the attention of attackers. Additionally, image attacks are possible on some occasions when QR codes are printed, transmitted and
scanned, and channel reliability is not always high; as a result, some shadows may be lost or damaged during processing. In other words, robustness is important in practical applications. The schemes of Chow et al. and Wan et al. reduce the error correction capacity, thus also weakening robustness. The error correction capacity is not reduced in the scheme of Fu et al., but the secret image is recovered by directly aligning the two shadows to perform the calculation, so when an offset, damage or other noise occurs, robustness will be very poor. The robustness aspect was not mentioned in the discussion of the other schemes above.

Compared to OR-based VSS, XVSS schemes have good contrast and resolution; RG-based VSS is also safe; QR code shadows can be decoded by the standard decoder, and error correction can be performed. Therefore, in this paper, we propose a XVSS scheme with robust and meaningful shadows based on QR codes. We use XVSS to share a secret image and generate noise-like shadows with higher visual quality and no pixel expansion. Furthermore, we replace the padding codewords of cover QR codes by the bits of shadows to generate robust and meaningful shadows. As a result, the error correction ability of QR code shadows is as strong as that of cover QR codes. Thus, the shadows are robust to conventional image attacks, such as rotation, JPEG compression, Gaussian noise, salt-and-pepper noise, cropping, resizing, and even the addition of camera and screen noise. The decoding results of QR code shadows are identical to cover QR codes; thus, they will not arouse the suspicions of attackers. We can recover the secret image from at least k shadows. When k is equal to n, the secret can be recovered without loss. Our scheme is robust and is characterized by perfect security and recovery. Experimental results and comparisons will demonstrate the effectiveness and robustness of our scheme.

The rest of our paper is organized as follows. Section 2 introduces QR codes and (k, n)-threshold XVSS. The main idea of the proposed scheme, related algorithms and analyses are described in Section 3. Section 4 presents experimental results, comparisons and a robustness test. Section 5 concludes this paper.

2 Preliminaries

2.1 QR codes

A QR code [10] is a matrix symbol that can be decoded quickly by a standard QR code reader, since three corners of the symbol constitute a kind of a finder pattern that assists in locating the code’s position, and calculating its size and inclination. The black and white modules are randomly distributed in symbols. The size of a symbol depends on the version of the QR code that ranges from 1 to 40 and is denoted by \( V(1 \sim 40) \) and four different levels of error correction denoted by \( E(L, M, Q, H) \); their corresponding error correction abilities are \( L \sim 7\% \), \( M \sim 15\% \), \( Q \sim 25\% \), and \( H \sim 30\% \). A QR code can encode multiple formats of data. Various versions and error correction levels of QR codes have different data capacities [6].

Before encoding, the input data stream is analyzed first to determine the content and whether the default or other extended channel interpretation (ECI) and appropriate mode will be used. There are different modes defined according to character value and assignment related to the default ECI, such as the alphanumeric mode, the numeric mode, the Kanji mode, the 8-bit byte mode, that have different indicators. During encoding, the input data stream will be converted to a bit stream including an ECI header if the ECI is not the default ECI, followed by one or more different mode segments. In the default ECI mode, the bit stream begins with the first mode indicator.
In this paper, we use the default ECI to encode the input data stream. During encoding, the mode indicators are selected according to an analysis to be added at the start of each mode segment to change modes, and the character count indicator and data bit stream follow the mode indicator in each mode. A terminator is appended at the end of the last segment. The final bit stream is divided into a sequence of 8-bit codewords. Padding codewords are inserted after the terminator as necessary to fill the symbol’s capacity. Figure 1 shows the structure of a bit stream consisting of \( n \) segments.

The data codewords are divided according to the version and the error correction level into one or more blocks, and the corresponding error correction codewords in each block are calculated using different generator polynomials. The error correction algorithms are used in each block separately. Data and the corresponding error correction codewords are in turn assembled into the final sequence.

After all codewords have been encoded, data mask patterns are used to mask the encoding area to arrange dark and light modules evenly and prevent position detection patterns from appearing in other areas of symbols.

2.2 \((k, n)\)-threshold RG-based VSS in the scheme of Yan et al.

Despite the existence of many XVSS schemes, this paper chooses the scheme of Yan et al. because it has the advantage of being \((k, n)\)-threshold and having the features of progressive visual effect and lossless recovery such that when \( k = n \), the secret image can be recovered without loss. This section will describe the researchers’ scheme in detail.

In RG-based VSS, the pixel value of a secret image of size \( M \times N \) is denoted by \( s = S(i, j)(1 \leq i \leq M, 1 \leq j \leq N) \), and the secret is shared into shadows denoted by \( SC_n \). Here, a black pixel is denoted by 1, and a white pixel is denoted by 0. The following steps describe the sharing phases of \((2, 2)\) RG-based VSS.

Step 1: Generate 1 RG \( SC_1 \) randomly.
Step 2: Calculate \( SC_2 \) following (1). \( \overline{SC_1(i, j)} \) denotes a reversal of \( SC_1(i, j) \).

Recovery: \( S' = SC_1 \otimes SC_2 \) as in (1). If secret pixel \( S(i, j) \) of \( S \) is 1, the recovery result \( SC_1 \otimes SC_2 = 1 \) is always black. If a secret pixel is 0, because \( SC_1 \) are generated randomly, the recovery result \( SC_1 \otimes SC_2 = SC_1(i, j) \otimes SC_1(i, j) \) has a 50% chance of being black or white.

\[
SC_2(i, j) = \begin{cases} 
SC_1(i, j) & \text{if } S(i, j) = 1 \\
SC_1(i, j) & \text{if } S(i, j) = 0 
\end{cases} \tag{1}
\]

In fact, (1) is equivalent to \( s = sc_1 \oplus sc_2 \) or \( sc_2 = sc_1 \oplus s \). The reason is that if \( s = 0 \Rightarrow sc_2 = sc_1 \oplus 0 \Rightarrow sc_2 = sc_1 \), and if \( s = 1 \Rightarrow sc_2 = sc_1 \oplus 1 \Rightarrow sc_2 = \overline{sc_1} \). The same equation could be extended to (2). Most importantly, the XOR operation may be used during recovery.

\[
s = sc_1 \oplus sc_2 \oplus \cdots \oplus sc_k \tag{2}
\]

| Segment 1 | Segment 2 | \( \cdots \) | Segment \( n \) |
|-----------|-----------|-------------|--------------|
| Mode Indicator 1 | Character Count Indicator 1 | Data 1 | Mode Indicator 2 | Character Count Indicator 2 | Data 2 | \( \cdots \) | Mode Indicator \( n \) | Character Count Indicator \( n \) | Data \( n \) | terminator | Pad codewords |

\( \text{Fig. 1} \) Structure of a bit stream consisting of \( n \) segments
In this paper, we use the RG-based \((k, n)\)-threshold VSS generation phase of the scheme of Yan et al. [18] to generate initial shadows, as shown in Algorithm 1. Afterwards, the initial shadows are embedded into the padding codeword areas of cover QR codes to generate QR code shadows. When the secret image is recovered, the codewords in padding codeword areas of cover QR codes are extracted to reconstruct initial shadows, and the XOR operation is performed on the initial shadows to reveal the original secret image.

**Algorithm 1** RG-based \((k, n)\)-threshold VSS of the scheme of Yan et al. [18].

**Require:** Binary secret image \(S\) of size \(M \times N\), threshold parameters \((k, n)\).

**Ensure:** \(n\) shadows \(SC_1, SC_2, \ldots, SC_n\).

1. For each pixel \(S(i, j)\) (\(1 \leq i \leq M, 1 \leq j \leq N\)), repeat Steps 2-4.
2. Calculate \(sc_1, sc_2, \ldots, sc_k\) satisfying (2), set \(t_1 = sc_1, t_2 = sc_2, \ldots, t_k = sc_k\), where \(t_x\) denotes the temporary pixels, and \(x = 1, 2, \ldots, n - 1, n\).
3. Set \(t_{k+1} = t_{k+2} = \cdots = t_n = 0\).
4. Randomly distribute \(t_1, t_2, \ldots, t_n\) to \(SC_1(i, j), SC_2(i, j), \ldots, SC_n(i, j)\).
5. Output \(n\) shadows \(SC_1, SC_2, \ldots, SC_n\).

The advantage of Algorithm 1 is that a progressive visual effect can be attained when more than the minimum qualified set of shadows are collected, as proven in the cited paper. When \(k\) is equal to \(n\), the secret image can be restored losslessly. However, this scheme is not robust, and the shadow image is noise-like, which can easily arouse suspicion during storage and transmission. By contrast, in this paper the QR code shadows can be scanned and decoded and have the error correction capability. Based on the scheme of Yan et al., we propose our scheme in which the generated QR code shadows are robust and meaningful. The advantages of the researchers’ scheme are also preserved.

## 3 Proposed XOR-ed VSS with robust and meaningful shadows based on QR codes

### 3.1 Main idea of the proposed scheme

In this section, we propose using a \((k, n)\)-threshold VSS based on QR codes to generate robust and meaningful shadows. The secret image is a binary image. First, we compute the size of the secret image, and select \(n\) cover QR codes; the number of padding codewords in every cover QR code is larger than the size of the secret image. Second, we use an RG-based \((k, n)\)-threshold VSS in Algorithm 1 created according to the scheme of Yan et al.[18] to generate \(n\) initial shadows. The initial shadows are noise-like binary images of the same size as the secret image. Third, we encode initial shadows as \(n\) bit streams and convert them to \(n\) 8-bit codeword sequences. Next, we input the content of cover QR codes to create new QR codes. In the encoding phase, we replace the padding codewords after the terminator in \(n\) cover QR codes by \(n\) 8-bit codeword sequences to generate \(n\) QR shadows. Since no checking is performed on the padding codewords, the QR code shadows can be decoded by a standard QR code reader, and the decoding results of QR code shadows and cover QR code images are identical. We need to modify the QR code encoder to create QR code shadows.

For our \((2, 2)\)-threshold scheme considered as an example, Fig. 2 displays the idea of QR code shadows’ generation by the proposed scheme. The condition in (3) will be given in Section 3.4.
3.2 Sharing phase of the proposed scheme

We share a secret image into $n$ noise-like shadows using RG-based $(k, n)$-threshold VSS in Algorithm 1; these $n$ shadows are treated as $n$ bit streams. The bit streams are encoded as 8-bit codeword sequences. We input the decoding results of the cover QR codes to create QR code shadows. During encoding, we substitute the codeword sequences for the padding codewords to generate QR code shadows. The error correction codewords corresponding to the modified data codewords will be changed accordingly. Algorithm 2 describes the details of the corresponding algorithm steps.

**Algorithm 2** Proposed XOR-based VSS with robust and meaningful shadows based on QR codes.

**Require:** secret image $S$; $n$ different binary cover QR code images $C_1, C_2, \cdots, C_n$; threshold parameters $(k, n)$.

**Ensure:** $n$ QR code shadows $SC_1, SC_2, \cdots, SC_n$.

1. Share a secret image using RG-based $(k, n)$-threshold VSS in Algorithm 1 to generate $n$ initial noise-like shadows $S_1, S_2, \cdots, S_n$.
2. Convert the values of pixels in initial shadows $S_1, S_2, \cdots, S_n$ to bit streams $T_1, T_2, \cdots, T_n$. Convert each bit stream into an 8-bit codeword sequence denoted by $Q_i (1 \leq i \leq n)$.
3. Decode cover QR codes $C_1, C_2, \cdots, C_n$ to obtain $n$ decoding results $W_1, W_2, \cdots, W_n$.
4. for $i = 1$ to $n$ do
5. Input $W_i$ to encode QR code shadow $SC_i$. During encoding, replace the padding codewords of the QR code after the terminator by 8-bit codeword sequence $Q_i$;
6. Use the modified codewords to complete the encoding of QR code shadow $SC_i$.
7. end for
8. Output $n$ QR code shadows $SC_1, SC_2, \cdots, SC_n$.

When sharing the secret image, we should ensure that the number of padding codewords in every cover QR code is larger than the size of the secret image. The modified codewords are padding codewords used in the cover QR code to obtain QR code shadows. The shadows can be decoded, and the decoding results of shadows are the same as those of the cover QR codes, which would not raise doubts.
3.3 Recovery phase of the proposed scheme

We decode $n$ QR code shadows to obtain all of their data codewords, decoding results and mode types. We locate the positions of the padding codewords based on the length of the decoding results, the modes and the terminator. We read the codewords at the positions of padding codewords and convert them to $n$ corresponding bit streams. Afterwards, we perform the XOR operation on $t$ bit streams to obtain one bit stream result and put the bits in the bit stream into a new image to reconstruct the secret image.

Algorithm 3 describes the corresponding recovery algorithm’s steps. In the scheme of Fu et al. [5], the authors reconstructed the secret image by performing the XOR operation on the pixel values of shadows, which is an approach different from our scheme.

**Algorithm 3** Secret image recovery algorithm of our proposed scheme.

Require: $t$ shadows $SCi_1, SCi_2, \ldots, SCi_t (k \leq t \leq n)$; threshold parameters $(k, n)$.
Ensure: reconstructed secret image $S'$.

1. for $j = 1$ to $t$ do
2. \hspace{1em} Decode QR code shadow $SCi_j$, and obtain mode indicators, character count indicators and data;
3. \hspace{1em} Add all the lengths of mode indicators, character count indicators, data and the length of the terminator
4. \hspace{1em} Read the codewords after the position of padding codewords of $SCi_j$ to obtain one 8-bit codeword sequence $Q'_{ij}$.
5. \hspace{1em} Convert 8-bit codeword sequence $Q'_{ij}$ to bit streams $T'_{ij} (1 \leq j \leq t)$ of the same length denoted by $L$.
6. end for
7. for $l = 1$ to $L$ do
8. \hspace{1em} $RS(l) = T'_1(l) \oplus \cdots \oplus T'_{j}(l) \oplus \cdots \oplus T'_i(l)$, where $T'_i(l)$ denotes the $l$-th bit of bit stream $T'_{ij}$.
9. end for
10. Create a new image $S'$ using the bit stream $RS$.
11. Output the reconstructed secret image $S'$.

The QR code shadows can be decoded by a standard reader, and the results of decoding them are the same as those of decoding cover QR codes. Hence, the shadows are meaningful and would not arouse suspicion when they are transmitted over a public channel. The threshold of this scheme is $(k, n)$ because the shadows generated by Algorithm 2 and recovered from Algorithm 3 preserve the properties of threshold $(k, n)$.

3.4 Analyses

In this section, we present several theoretical analyses of the proposed scheme.

When padding codewords are added to complete the number of data codewords as necessary, the value of the former is not verified, so we replace them by the bits of initial shadows shared as a result of sharing a secret image using Algorithm 1 to generate QR code shadows. The error correction capacity is preserved; thus, the proposed scheme is robust to some conventional image attacks. The secret image can be restored when some damage or noise occurs in the shadows.
When the input data stream is encoded into data codewords, different mode segments are connected sequentially. The total number of data codewords is denoted by $R$; the length of the mode indicator is 4; the number of bits of the character count indicator in mode segment $i$ is denoted by $B_i$, and the number of bits of data according to the mode type of mode segment $i$ is denoted by $C_i$. The length of the terminator is denoted by $D$. The embeddable data capacity of cover QR code $ER$ is given by (3).

$$M \times N \leq ER = 8 \times R - \sum_{i=1}^{n} (4 + B_i + C_i) - D$$  \hspace{1cm} (3)

The size of the secret image should satisfy the condition $M \times N \leq ER$. For example, in Fig. 3 the size of the secret image is 45*45, which is equal to 2025. It is divided into $n$ noise-like shadows of the same size as the secret image. Once all pixels in all noise-like shadows have been converted into 8-bit codeword sequences, the size of each sequence is 254 bytes. Thus, we select $n$ cover QR codes that have more than 254 padding codewords. In Fig. 3, the counts of padding codewords of cover QR codes are 272, 272,
and 273, respectively, which are larger than 254. Therefore, they can be used to insert noise-like shadows. We use the decoding results of three cover QR codes to create three new QR codes, and in the encoding phase, we replace 254 padding codewords after the terminator by the pixels of the noise-like shadows. The new QR codes are QR code shadows.

QR code shadows can be decoded by a standard reader, and the decoding results of QR code shadows are the same as those of cover QR codes. Hence, the shadows are meaningful, and when they are transmitted over a public channel, they will not arouse suspicion. The threshold of this scheme is \((k, n)\), since the shadows generated by Algorithm 2 and recovered from Algorithm 3 preserve the properties of threshold \((k, n)\).

### 4 Experiments and comparisons

In this section, we describe experiments and comparisons performed to evaluate the effectiveness of the proposed scheme.

#### 4.1 Image illustration

A software implementation of the proposed scheme was developed in the Python programming language. Examples of various threshold results produced by our scheme are depicted in Figs. 3, 4 and 5. Figure 3 shows our \((2, 2)\)-threshold XVSS based on QR codes. All QR codes in Fig. 3 are of version 25 and use error correction level H. Figure 3a shows a secret image \(S\) of size 64 × 64. Figure 3b–c present two cover QR code images \(C_1\) and \(C_2\). Figure 3d–e display the corresponding QR code shadows \(SC_1\) and \(SC_2\). Figure 3f–g illustrate the decoding results of \(C_1\) and \(C_2\). Figure 3h–i show the decoding results of \(SC_1\) and \(SC_2\).

Figure 3j presents the reconstructed secret image \(S'\), which is the result of performing the XOR operation on the shadows embedded in \(SC_1\) and \(SC_2\). The pixels in Fig. 3j are the same as those in Fig. 3a, which indicates that \(S\) and \(S'\) are identical. Thus, we observe that the secret has been restored losslessly.

Figure 4 visualizes our \((2, 3)\)-threshold XVSS based on QR codes. All the QR codes are of version 25 and use error correction level H. Figure 4a shows a 64 × 64 secret image \(S\). Figure 4b–d present cover QR code images \(C_1\), \(C_2\) and \(C_3\). Figure 4e–g shows QR code shadows \(SC_1\), \(SC_2\) and \(SC_3\). Figure 4h–j illustrates the results of decoding \(C_1\), \(C_2\) and \(C_3\). Figure 4k–m display the results of decoding \(SC_1\), \(SC_2\) and \(SC_3\). Figure 4n shows the result of performing the XOR operation on the initial shadows embedded in \(SC_1\) and \(SC_2\). Figure 4o presents the result of performing the XOR operation on the initial shadows embedded in \(SC_1\) and \(SC_3\). Figure 4p displays the result of performing the XOR operation on the initial shadows embedded in \(SC_2\) and \(SC_3\). Considering Fig. 4n–p, we can partially reveal the secret image.

Figure 4q shows the reconstructed secret image \(S'\) that results from performing the XOR operation on the initial shadows embedded in \(SC_1\), \(SC_2\) and \(SC_3\). The pixels in Fig. 4q are the same as those in Fig. 4a, and we observe that \(S\) and \(S'\) are identical; thus, the secret has been restored losslessly.

Figure 5 illustrates our \((3, 3)\)-threshold XVSS based on QR codes. All QR codes are of version 17 and use error correction level H. Figure 5a shows a 45 × 45 secret image \(S\). Figures 5b–d present cover QR code images \(C_1\), \(C_2\) and \(C_3\). Figure 5e–g display QR code shadows \(SC_1\), \(SC_2\) and \(SC_3\). Figure 5h–j show the results of decoding \(C_1\), \(C_2\) and
Fig. 4 Our (2, 3)-threshold XVSS based on QR codes of version 25 using error correction level H

Figures 3, 4 and 5 show three different thresholds of our XVSS based on QR codes of different versions. Thus, \((k, n)\)-threshold of our XVSS based on QR codes can be realized.

\(C_3\). Figure 5k–m present the results of decoding \(SC_1\), \(SC_2\) and \(SC_3\). Figure 5q shows the reconstructed secret image \(S'\) that results from performing the XOR operation on the initial shadows embedded in the \(SC_1\), \(SC_2\) and \(SC_3\). The pixels in Fig. 5n are the same as those in Fig. 5a. \(S'\) contains the entirety of the information of \(S\) and recovers the secret losslessly.

Figures 3, 4 and 5 show three different thresholds of our XVSS based on QR codes of different versions. Thus, \((k, n)\)-threshold of our XVSS based on QR codes can be realized.
4.2 Robustness test

Since the modifications of the cover QR code image are padding codewords that are not verified during encoding, the QR code can be decoded correctly, and the error correction capacity is preserved. Our scheme is as robust as the error correction capability of QR codes. As long as the QR code shadows can be decoded normally in a given noise environment,
we can recover the data of the secret image from them. In this section, we consider our (2, 2)-threshold scheme as an example. Figure 6 shows the robustness of shadows after the addition of camera and screen noise.

Figure 6a presents the secret image $S$. Figures 6b–c show the cover QR code images $C_1$ and $C_2$. Figures 6d–f present QR code shadows $SC_1$ and $SC_2$. Figures 6f–g illustrate the results of decoding $SC_1$ and $SC_2$. Figures 6h–i display the images $P_1$ and $P_2$ obtained by photographing $SC_1$ and $SC_2$ on the screens. Figures 6j–k illustrate the results of decoding $P_1$ and $P_2$. Figure 6l shows the reconstructed secret image $S'$, which is the same as $S$.

Fig. 6 Robustness test of our (2, 2)-threshold XVSS
The effects of various attacks on two shadows, including rotation, cropping, resizing, addition of noise and JPEG compression are tested and summarized in Table 1 to further show the robustness of our scheme. Table 1 demonstrates that conventional attacks, including rotation by different angles, resizing the shadows by various factors, JPEG compression of various quality levels, addition of salt-and-pepper noise and Gaussian noise of certain proportions, and partial cropping of the shadows can be resisted. Such attacks can be encountered during transmission and delivery of shadows, so the robustness of our scheme is very useful for practical applications.

### 4.3 Comparisons with related studies

In this section, we first describe the differences between the schemes developed by Chow et al. [3], Wan et al. [13] and Fu et al. [5] using diagrams, analyze the characteristics of various schemes, and finally present an objective evaluation in Table 2.

Figure 7 shows an experimental result of the scheme of Chow et al. [3]. Figure 7a presents a secret image \( S \). Figures 7b–d display cover QR code images \( C_1, C_2 \) and \( C_3 \). Figures 7e–g show QR code shadows \( SC_1, SC_2 \) and \( SC_3 \). Figure 7h presents the reconstructed secret image \( S' \). Figure 7i illustrates the decoding information of \( S \) and \( S' \).

| Type of attack          | Performance | Parameter | Can \( S' \) be reconstructed (yes/no)? | Type of attack | Parameter | Can \( S' \) be reconstructed (yes/no)? |
|-------------------------|-------------|-----------|----------------------------------------|----------------|-----------|----------------------------------------|
| JPEG Compression        |             | Q = 90    | yes                                    | Resizing       | 0.5       | yes                                    |
|                         |             | Q = 80    | yes                                    |                | 0.8       | yes                                    |
|                         |             | Q = 70    | yes                                    |                | 1.1       | yes                                    |
|                         |             | Q = 60    | yes                                    |                | 1.4       | yes                                    |
|                         |             | Q = 50    | yes                                    |                | 1.7       | yes                                    |
|                         |             | Q = 40    | yes                                    |                | 2.0       | yes                                    |
|                         |             | Q = 30    | yes                                    | Gaussian noise | (2,4,10%) | yes                                    |
|                         |             | Q = 20    | yes                                    | (mean,variance, p) | (2,4,20%) | yes                                    |
| Salt-and-pepper noise   |             | 10%       | yes                                    |                | (2,4,30%) | yes                                    |
|                         |             | 20%       | yes                                    |                | (2,4,40%) | yes                                    |
|                         |             | 30%       | yes                                    |                | (2,4,50%) | yes                                    |
|                         |             | 40%       | yes                                    |                | (2,4,60%) | yes                                    |
|                         |             | 50%       | yes                                    |                | (2,4,70%) | yes                                    |
|                         |             | 60%       | yes                                    |                | (2,4,80%) | yes                                    |
|                         |             | ≥70%      | no                                     |                | (2,4,90%) | no                                     |
| Cropping                |             | 1/100     | yes                                    | Rotation       | 15°       | yes                                    |
|                         |             | 1/25      | yes                                    |                | 30°       | yes                                    |
|                         |             | 9/100     | yes                                    |                | 45°       | yes                                    |
|                         |             | 4/25      | yes                                    |                | 60°       | yes                                    |
|                         |             | 1/4       | yes                                    |                | 75°       | yes                                    |
|                         |             | ≥9/25     | no                                     |                | 90°       | yes                                    |
### Table 2 Comparisons of our scheme with existing schemes

| Scheme                        | Metric                          | Maximum embeddable capacity                                                                 | Reduction of the error correction capacity of a shadow | Threshold | Is the scheme lossless (robust)? |
|-------------------------------|---------------------------------|--------------------------------------------------------------------------------------------|------------------------------------------------------|-----------|---------------------------------|
| Our par Scheme                |                                 | $8 \times R - \sum_{i=1}^{n} (4 + B_i + C_i) - D$                                           | $(k, n)$                                             | 0         | Yes                             |
| Scheme of Chow et al.         | r per error correction block    | $c w_i$ per error correction block                                                           | $(n, n)(n \geq 3)$                                     | No        |                                 |
| Scheme of Wan et al.          | Slightly less than r per error correction block | $x * y / n$                                                                               | $(k, n)$                                             | No        |                                 |
| Scheme of Fu et al.           |                                 | $< 8 \times R - \sum_{i=1}^{n} (4 + B_i + C_i) - D$                                          | 0                                                    | $(k, n)$  | No                              |
Figure 8a shows a secret image \( S \), which is a QR code. Figures 8b–d present cover QR code images \( C_1, C_2 \) and \( C_3 \). Figures 8e–g show QR code shadows \( SC_1, SC_2 \) and \( SC_3 \). Figure 8h presents the reconstructed secret image \( S' \). The pixels in Fig. 8h are the same as those in Fig. 8a. Thus, the secret image has been recovered losslessly. Figure 8j displays the decoding information of \( S \) and \( S' \).

Figure 9a shows a secret image \( S \). Figures 9b–d present cover QR code images \( C_1, C_2 \) and \( C_3 \). Figures 9e–g illustrate QR code shadows \( SC_1, SC_2 \) and \( SC_3 \). Figures 9h–j display the decoding results of \( C_1 \) and \( SC_1, C_2 \) and \( SC_2 \), and \( C_3 \) and \( SC_3 \), respectively. Figure 9k shows the reconstructed secret image \( S' \). Figures 9l–n present the results of shadows \( SC_1, SC_2 \) and \( SC_3 \) with 10% salt-and-pepper noise. Figure 9o displays the XOR-ed result of \( l, m \), and \( n \).

Figure 10 shows sharing and recovery results of (2, 4)-threshold VSS using the scheme of Fu et al. \[5\]. Figures 10a–d display QR code shadows denoted by \( S_1, S_2, S_3, \) and \( S_4 \) that can be decoded by a standard QR reader. Figure 10e presents the result of decoding \( S_1 \). Figure 10f displays the secret image \( S \). Figure 10g and h show recovered images reconstructed by performing XOR operations on \( S_1 \) and \( S_2 \), and \( S_1 \) and \( S_4 \), respectively.

---

**Fig. 7** (3, 3)-threshold VSS on QR codes of version 4 using error correction level H for the scheme of Chow et al. \[3\]
The secret image in Fig. 7 for the scheme of Chow et al. [3] must be a QR code image; the secret image has not been fully restored. Figures 3, 5 and 8 show that in our scheme, the secret image can be any binary image, including a QR code image, and if the threshold is \((n, n)\), the secret image is recovered losslessly. The threshold parameter is \((k, n)\) in our scheme, while the value is \((n, n)(n \geq 3)\) in the scheme of Chow et al..

The secret in Fig. 9 for the scheme of Wan et al. [13] is a binary image. Wan et al. embed the secret image starting from coordinates \((7, 7)\) to the lower right corner continuously within the range of the error correction capability. If some noise is added to the shadows in Fig. 9, we observe that the reconstructed secret image also exhibits the added noise, as shown in Fig. 9n. However, if more noise is added to the shadows, the secret image will probably not be revealed. Nonetheless, as shown in Fig. 6n, for our scheme, we can recover the secret image losslessly if the camera and screen noises are added simultaneously.

Fu et al. [5] also utilize the encoding redundancy of QR codes to share a secret image by a probabilistic sharing method and preserve the error correction capability in the cover QR code. All shadows can be decoded by a standard QR reader. However, the differences between the above scheme and our method are described as follows.

1. Fu et al. replace the padding codewords of the generated QR codes with the secret image and subsequently recalculate the error correction codewords to obtain the modified data codewords. In contrast, we replace the padding codewords with the initial shadows during encoding. Thus, Fu et al. do not modify the encoder, while we do. They embed the entire image into the continuous area of the padding codewords of cover QR codes and need a continuous area, while we do not.

2. During recovery, Fu et al. directly perform the XOR operation on the pixel values at the matching locations of two shadows to obtain the secret data. In contrast, we extract the codewords at the position of the padding codewords corresponding to shadows and subsequently perform the XOR operation on the extracted codewords to restore the
secret image. Thus, we modify the decoder to extract the codewords, while Fu et al. do not.

3. As the XOR operation is performed directly on the shadow images, the error correction codewords are useless, and their recovery phase is not robust. When shift conversion or other image attacks are performed on the shadows, the secret image cannot be restored. In contrast, the error correction codewords are used in our scheme to obtain correct data.
codewords of the initial shadows even if some image attacks are experienced, so that the secret image can be restored losslessly, demonstrating the robustness of our scheme.

We summarize the comparisons of our scheme with the schemes of Chow et al. [3], Wan et al. [13], and Fu et al. [5] in terms of the image embeddable capacity of a cover QR code, the extent of reduction of the original error correction capability, threshold and whether the secret image can be restored without loss when attacks are performed on the shadows. The detailed comparisons are shown in Table 2.

We have different ways of embedding secrets into cover QR codes from the schemes of Chow et al., Wan et al. and Fu et al.; hence, the maximum embeddable capacities of cover QR codes vary. Our scheme replace the padding codewords by initial shadows shared from secret images, so our scheme’s maximum embeddable capacity is the total number of data codewords minus the number of data codewords used to encode the data for the cover QR code, given by Eq. (3). However, the error correction ability is preserved. Therefore, when we use $n$ shadows with some degree of noise to recover the secret image, we can recover it losslessly, and our threshold is $(k, n)$.

In the scheme of Chow et al., the maximum capacity of their scheme is the error correction capacity $r$ per error correction block. When they modify $cw(i)$ per error correction block of each cover QR code to share the secret QR code, the corresponding error correction capacity of the generated shadow is reduced by $cw(i)$ per error correction block, where $i$ represents the sequence number of the error correction block. The threshold is $(n, n)(n \geq 3)$.

Wan et al. divide a secret image into $n$ parts and replace the codewords in the partial area of each cover QR code starting from coordinates $(7, 7)$ and proceeding to the lower right corner continuously by the corresponding parts of the secret. Therefore, the maximum embeddable capacity of each cover QR code is slightly less than $r$ per error correction block. When they share the secret image into the QR code cover, the corresponding error correction capacity of the generated shadow is reduced by $x * y / n$, where $x$ and $y$ represent the length and width of the embeddable image respectively. The threshold is $(k, n)$. 

Fig. 10 Sharing and recovery results of (2, 4)-threshold VSS for the scheme of Fu et al. [5]
Since the error correction capacities in the schemes of Chow et al. and Wan et al. are reduced, when image attacks such as rotation, resizing, JPEG compression, addition of salt-and-pepper noise and Gaussian noise, and cropping performed on the shadows occur, the secret image will not be restored losslessly.

Fu et al. embed the secret image into the padding codewords’ area of the cover QR code continuously. The data codewords and the error correction codewords are staggered, so finding a continuous area in the padding area of the QR code is challenging. Thus, the embeddable capacity is less than the value in our scheme. On the other hand, the error correction capacity is not reduced. Nonetheless, the method of recovering the secret image in the researchers’ scheme is not robust in the event of image attacks. The threshold is \((k, n)\).

In the case where the error correction level of the QR code is \(H\), the number of data code words is slightly less than the error correction capacity. It is assumed that the padding codewords in the QR code occupy half of the data codewords. The maximum embeddable capacity of our scheme is about half of the error correction capacity; the embeddable capacity of the Chow et al.’s scheme is error correction capacity \(r\); the maximum embeddable capacity of the Wan et al.’s scheme is slightly less than the error correction capacity \(r\); the embeddable capacity of Fu et al.’s scheme is smaller than our scheme. Therefore, when comparing the maximum embeddable capacity, the maximum embeddable capacity of the scheme of Fu et al.’s is smaller than our scheme, which is smaller than that of Wan et al., and the scheme of Wan et al. is smaller than that of Chow et al..

If the number of padding codewords is approximately equal to the number of data codewords, the maximum embeddable capacity of our scheme is similar to that of Wan et al.’s scheme, and other rankings remain unchanged. When a QR code with a lower error correction level is selected, the number of data codewords is larger than the error correction capacity. When the number of padding codewords approaches the total number of data codewords, the maximum embeddable capacity of our scheme is the largest.

5 Conclusion

This paper investigates a robust XVSS based on QR codes. The proposed approach is robust because the error correction capacity of QR codes is preserved. The proposed scheme utilizes the defect that the padding codewords are not checked when a QR code is encoded and replaces the padding codewords of the QR code by the bits of shadows shared from the secret image in the process of encoding QR code shadows. As demonstrated in the experiments, the \((k, n)\)-threshold XVSS can be implemented using QR codes; QR code shadows are meaningful because they can be decoded by a QR reader, and the proposed scheme is more robust than the existing related schemes. The robustness test indicates that our scheme is robust to conventional image attacks, including JPEG compression, Gaussian noise, salt-and-pepper noise, cropping, rotation, resizing, and even the addition of camera and screen noises. However, the size of the secret image is limited by the number of padding codewords of the cover QR code, which is a problem that needs to be addressed in future studies.

Acknowledgments The authors thank the reviewers for their comments and suggestions for improving the quality of the manuscript. This study is supported by the National Natural Science Foundation of China (Grant Number 61602491) and the Key Program of the National University of Defense Technology (Grant Number ZK-17-02-07).
Open Access  This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Cheng Y, Fu Z, Yu B (2018) Improved visual secret sharing scheme for qr code applications. IEEE Transactions on Information Forensics and Security 13(9):2393–2403
2. Chow YW, Susilo W, Tonien J, Vlahu-Gjorgjevska E, Yang G (2018) Cooperative secret sharing using qr codes and symmetric keys. Symmetry 10(4):95–107
3. Chow YW, Susilo W, Yang G, Phillips JG, Pranata I, Barmawi AM (2016) Exploiting the error correction mechanism in qr codes for secret sharing. Lecture Notes in Computer Science, Springer 9722:409–425
4. Cimato S, De Prisco R, De Santis A (2006) Probabilistic visual cryptography schemes. The Computer Journal 49(1):97–107
5. Fu ZYC, Yu B (2018) Visual cryptography scheme with meaningful shares based on qr codes. IEEE Access 6:59567–59574
6. Information - automatic identification and data capture techniques - qr code barcode symbology specification. Iso/iec(18004:2015) (2015)
7. Kuwakado H, Tanaka H (1999) Image size invariant visual cryptography. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 82(10):2172–2177
8. Liu Y, Fu Z, Wang Y (2016) Two-level information management scheme based on visual cryptography and qr code. Application Research of Computers 33(11):3460–3463
9. Naor M, Shamir A (1995) Visual cryptography.: Advances in cryptology-EUROCRYPT’94 Lecture Notes in Computer Science, Workshop on the Theory and Application of Cryptographic Techniques, Springer, pp 1–12
10. Samretwit D, Wakahara T (2011) Measurement of reading characteristics of multiplexed image in qr code. In: International Conference on Intelligent NETWORKING and Collaborative Systems, pp 552–557
11. Shen G, Liu F, Fu Z, Yu B (2017) Perfect contrast XOR-based visual cryptography schemes via linear algebra. Designs, Codes and Cryptography 85(1):15–37
12. Tuyls P, Hollmann HD, Van Lint JH, LMGM T (2005) XOR-Based visual cryptography schemes. Des Codes Crypt 37(1):169–186
13. Wan S, Lu Y, Yan X, Liu L (2017) Visual secret sharing scheme with (k,n) threshold based on qr codes. In: International Conference on Mobile Ad-Hoc and Sensor Networks, pp 374–379
14. Wang G, Liu F, Yan WQ (2016) 2D Barcodes for visual cryptography. Multimedia Tools and Applications 75(2):1223–1241
15. Weir J, Yan WQ (2011) Authenticating visual cryptography shares using 2D barcodes. International Workshop on Digital Watermarking, Springer 7128:196–210
16. Wu X, Sun W (2013) Random grid-based visual secret sharing with abilities of or and xor decryptions. Journal of Visual Communication and Image Representation 24(1):48–62
17. Yan X, Liu X, Yang CN (2018) An enhanced threshold visual secret sharing based on random grids. J Real-Time Image Proc 14(1):61–73
18. Yan X, Lu Y (2017) Progressive visual secret sharing for general access structure with multiple decryptions. Multimedia Tools and Applications 77(2):1–20
19. Li P, Liu ZQ, Yang CN (2018) A construction method of (t, k, n)-essential secret image sharing scheme. Signal Processing 65:210–220
20. Yan X, Wang S, El-Latif AAA, Niu X (2015) Visual secret sharing based on random grids with abilities of AND and XOR lossless recovery. Multimedia Tools and Applications 74(9):3231–3252
21. Yan X, Wang S, Niu X (2014) Threshold construction from specific cases in visual cryptography without the pixel expansion. Signal Process 105:389–398
22. Yang CN (2004) New visual secret sharing schemes using probabilistic method. Pattern Recogn Lett 25(4):481–494
23. Yang CN, Liao JK, Wu FH, Yamaguchi Y (2016) Developing visual cryptography for authentication on smartphones. In: International Conference on Industrial IoT Technologies and Applications, Springer, pp 189–200

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
Longdan Tan was born in China, in 1987, received the B.Sc. degree with honor rank in Computer Application, China in 2008, M.Sc. degree in Information Security in 2011 from Hefei electronic engineering institute. She now is a PhD candidate at National University of Defense Technology, Hefei, P. R. China. Her areas of interests are multimedia security and secret image sharing.

Yuliang Lu was born in China, in 1964, received the B.Sc. degree with honor rank in Computer Application, China in 1985 and M.Sc. degree in Computer Application in 1988 from Southeast University. He now is a professor at National University of Defense Technology, Hefei, P. R. China. His areas of interests are computer application and information processing.
Xuehu Yan was born in China, in Feb 1984, received the B.Sc. degree with honor rank in Information and Calculate Science, China in 2006, M.Sc. degree in Computational Mathematics in 2008, and doctoral degree in Computer Science and Technology in 2015 from Harbin Institute of Technology. He now is an Associate Professor at National University of Defense Technology, Hefei, P. R. China. His areas of interests are visual cryptography, secret image sharing, information hiding, cryptography and multimedia security. He has published more than 100 papers in these areas. He is an Associate Editor of the International Journal of Digital Crime and Forensics (IJDCF).

Lintao Liu was born in China, in Dec 1989, received the B.Sc. degree with honor rank in Computer Application, China in 2012, M.Sc. degree in Information Security in 2015 from National University of Defense Technology. He now is a PhD candidate at National University of Defense Technology, Hefei, P. R. China. His areas of interests are cryptography, multimedia security and biometrics.
Xuan Zhou was born in China, in Aug 1991, received the B.Sc. degree with honor rank in Computer Application, China in 2013, M.Sc. degree in Information Security in 2016 from Hefei electronic engineering institute. He now is a PhD candidate at National University of Defense Technology, Hefei, P. R. China. His areas of interests are network security and moving target defense.