Journal of Information Hiding and Multimedia Signal Processing ©2026 ISSN 2073-4212
Ubiquitous International Volume **17**, Number **1**, March **2026**

A Simple Authentication Method of QR Code Using OCR

Mao-Hsiung Hung[1,*]

[1]School of Computer Science and Mathematics, Fujian University of Technology
No.69, South Xuefu Road, Shangjie Town, Minhou County, Fuzhou, 350118, China
mhhung0502@outlook.com

*Corresponding author: Mao-Hsiung Hung

Received December 9, 2025, revised January 28, 2026, accepted February 5, 2025.

Abstract. *QR code is widely used by smartphone payment, but the security problems are still happening such as payment theft due to malicious replacement of QR code. In this paper, we propose a simple authentication of QR code based on the checksum verification with OCR. The proposed checksum of the message text of QR code is generated and then is pasted beside the QR code. Based on QR code with its checksum, the authentication performs the checksum verification using OCR. The simulation results indicate that the proposed QR code can be successfully authenticated under print-and-scan scenario.*
Keywords: QR code authentication, OCR, checksum

1. **Introduction.** QR code has very high popularity to be used in smartphone payment. Merchants request for payment codes from the third-party platforms like Alipay and WeChat Pay. The payment codes provide media to make payment between merchants and customers. In general scenario, customers use smartphone to scan specific QR codes of merchant's owners. Then, customers input money amounts and then confirm the payments by entering a password. The payment is processed and both the customer and merchant receive a confirmation.

Although QR code brings much convenience of transactions, it causes some security issues. Payment theft is one of the most common security concerns. Thefts maliciously replace merchants' payment QR codes with their own, as shown in Fig.1, which is captured from the news report [1]. Customers scan the fraud code to be redirect to theft's accounts by which thefts steal money from the QR code payment. The kind of theft is hard to unveil promptly, if merchants and customers do not confirm each other.

Figure 1. Malicious replacement of payment QR code [1]

In the past, most of the authentication of QR code depended watermarking with a signature [2]-[3]. The signature is often generated by processing the message text loaded in QR code. The processing often includes hashing and encryption. The hashing transforms the message text into an abstraction or so-called a digest. The digest is generally encrypted into the signature with a private key. Finally, the signature is watermarked into QR code. During authentication, QR code with watermark is extracted out a signature. Then, the extracted signature is decrypted into a digest with a public key. Finally, the decrypted digest is used to compare with the original digest to perform the verification.

Although the watermark of QR code hiding the signature can solve counterfeiting, it is still face the attack problems. When the watermark meets an attacking such as geometrical transformations, the signature could not be successfully extracted from the watermark. As a result, the authentication for the attacked genuine QR codes fails in possibility. Moreover, print-and-scan scenario is still very challenging to these watermarking QR codes for authentication because of imaging distortion after scanning [4]-[6].

Recently, the paper of [7] presents a system to generate and verify secure QR codes using a digital watermarking. The system embeds tamper-resistant information in QR codes to make unauthorized modification more difficult. By the way, the authors develop a neural network to verify the genuinity of scanned QR codes. Another work of [8] proposes a three-layer protection scheme for QR codes. The data is first encrypted by AES. Then, a multiplexing method integrates two independent dichromatic QR codes into a single printed chromatic code. Finally, the dichromatic code is recovered through specific wavelengths. The mechanism provides separate channels as a physical access to restrict unauthorized QR codes.

This paper presents a simple authentication method using OCR to prevent payment thefts. The message text of a QR code is computed into a checksum. The digit images of the checksum is warped and pasted beside the QR code. During the QR code payment, the image of QR code with its checksum is scanned. The verification of OCR's recognition in the checksum area performs to authenticate the QR code. Because the digit image of the checksum is visualized, the fake QR codes are effectively unveiled by the checksum authentication or merchants. In addition, the proposed method of the checksum authentication is quite simple and robust to imaging distortion while code scanning.

The rests of the paper are organized as follows. Section 2 presents the proposed QR code authentication with OCR. Section 3 describes the proposed method in details. The simulation results and comparison are given in Section 4. Section 5 concludes the paper and gives future works.

2. **QR Code Authentication Using OCR.** In our proposed authentication, we design a new combination of QR code and its checksum in image form. As shown in Fig.2, a message text such as "www.fjut.edu.cn" is loaded into QR code generator to create its corresponding QR code image. Meanwhile, a checksum generator is proposed to generate a four-digit checksum according to the message text, such as [7,6,5,9] for "www.fjut.edu.cn". Taking the QR code image as a background, we warp and paste four digit images of checksum along the right border in a vertical arrangement. Digit images from 0 to 9 could be prepared at the advance for warping and pasting. As a result, the QR code can carry its checksum to allow the authentication using OCR.

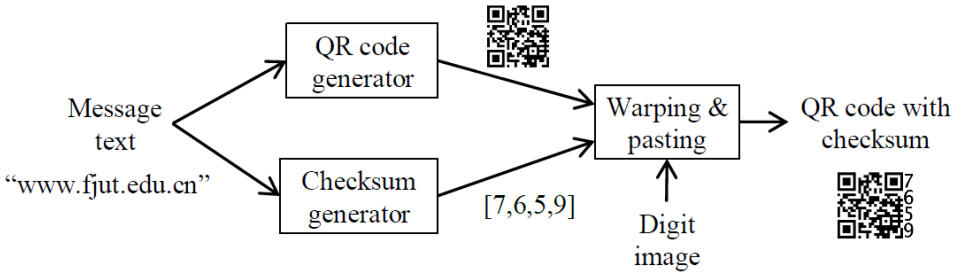


FIGURE 2. Generation of QR code with checksum

During the authentication, the image of QR code with its checksum is scanned into QR code detector, as shown in Fig.3. Then, two areas of QR code and checksum are obtained to respectively input QR code decoder and OCR. The QR code can be decoded out its message text. The message text is computed by checksum generator and the checksum (CS_1) is obtained. Meanwhile, OCR recognizes the checksum area of the inputting image to obtain the recognition result (CS_2). CS_1 and CS_2 compare and if equal to approve the QR code authentication. If CS_1 does not equal to CS_2 then the authentication is not approved.

In the practices of the authentication of the proposed QR code with checksum, we consider three conditions to be encountered and an example is given as the following. Two message texts of "www.fjut.edu.cn" (fjut in abbreviation) and "www.fjvt.edu.cn" (fjvt in abbreviation) are used to generate two QR codes, in which respectively play a genuine code and a fake code. In the other word, fjvt's QR code intents to pretend fjut's QR code. Then, the two checksums of fjut and fjvt are respectively computed into 7659 and 7639. Therefore, fjut's QR code with 7659 and fjut's QR code with 7639 are given to authenticate,

as shown in Fig.4(a)-(b). Although the two codes and their corresponding checksums can be approved by smartphone scanning, fjut's owner can easily distinguish the fake code from different checksums. That also means that 7659 can be approved but 7639 cannot be approved by the vision of fjut's owner. If fjvt's QR code is advertently with fjut's checksum i.e. 7659, as shown in Fig.4(c), smartphone scanning do not approve the fake code even if fjut's owner is unaware or unnoticed to approve it. Therefore, a genuine code can be approved by both of its owner's vision and smartphone scanning, but a fake code cannot be approved by both of that.

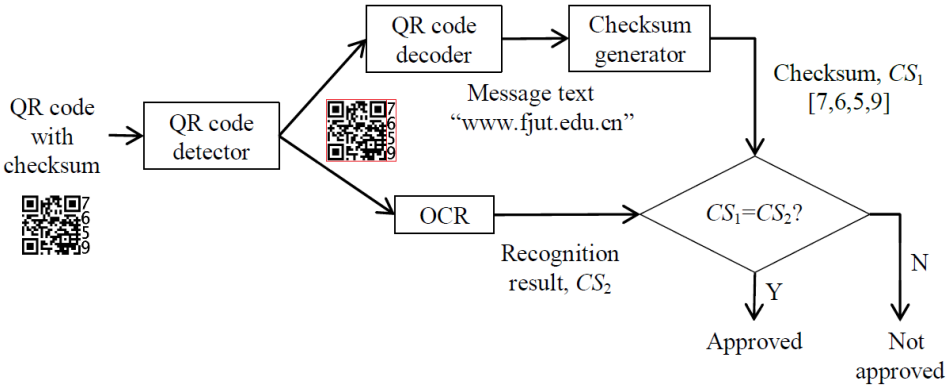


FIGURE 3. QR code authentication using OCR

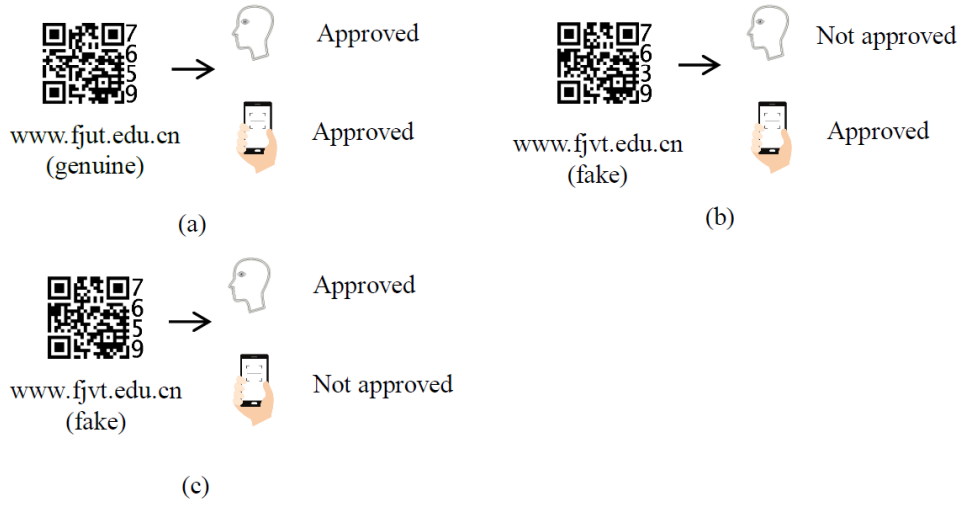


FIGURE 4. Three conditions of QR code authentication using OCR

3. **Proposed Method.** As mentioned in the above section, our proposed authentication requires handlings including checksum generation, image warping for checksum during generation and authentication, and OCR for checksum digits. The following subsections describe the details of these handlings.

3.1. **Proposed checksum generation.** Luhn algorithm [9] is a simple check digit formula which is used to validate identification number. In our work, we modify and extend Luhn algorithm to generate a four-digit checksum for the message text of QR code, as detailed in the following.

An array is used to store the string of a message text which is coded into a QR code, denoted by $\boldsymbol{c}$. The array of $\boldsymbol{c}$ contains n elements such as $[c_0, c_1, c_2, \ldots, c_{n-1}]$, where c_i represents the i-th character's ASCII code. To weigh every elements of $\boldsymbol{c}$, a multiplier array is designed by $[w_1, w_2, w_2, w_2, w_1, w_2, w_2, w_2, \ldots]$, where two multipliers of w_1 and w_2 compose a sequence of w_1, w_2, w_2, w_2. Then, the sequence repeats

until the multiplier array reaches the length of n. w_1 is assign to a large weight and meanwhile w_2 is assign to a small one. w_1 and w_2 are experimentally assigned to 2 and 1/16 respectively. We denote the multiplier array by $\boldsymbol{m}_1$. The weighting calculation performs the element-wise multiplication to $\boldsymbol{c}$ and $\boldsymbol{m}_1$, and then $floor$ operation applies every productions of $\boldsymbol{c} * \boldsymbol{m}_1$ in order to taking integer, as shown in Eq.(1). An array of $\boldsymbol{p}$ stores the weighting results and the symbol of $(*)$ represents the element-wise multiplication.

$$\boldsymbol{p} = floor(\boldsymbol{c} * \boldsymbol{m}_1) \tag{1}$$

After the above calculation, we transform every weighting result to the sum of a quotient and a remainder. Every element of $\boldsymbol{p}$ is divided by a number of b to obtain the quotient and the remainder and then the two numbers add together as shown in Eq.(2). $(/b)$ and $(\%b)$ represent the integer division and modulo with b respectively and an array of t stores the sums of $\boldsymbol{p}/b$ and $\boldsymbol{p}\%b$. b can be assigned to 16.

$$\boldsymbol{t} = \boldsymbol{p}/b + \boldsymbol{p}\%b \tag{2}$$

Then, we perform a summation to all elements of $\boldsymbol{t}$ and assign the result to a variable of s. Finally, 1st digit of checksum (d_1) is calculated by the following equation, where %10 is taking module 10, so that $(s + d_1)$ becomes multiples of 10.

$$d_1 = (10 - (s\%10)\%10 \tag{3}$$

We use a right shifting operation to vary $\boldsymbol{m}_1$ array by assigning $(i+1)$-th element to i-th element and assigning the first element to w_2 such as from $[w_1, w_2, w_2, w_2, w_1, w_2, w_2, w_2, \ldots]$ to $[w_2, w_1, w_2, w_2, w_2, w_1, w_2, w_2, w_2, \ldots]$. The shifting result is denoted by an array of $\boldsymbol{m}_2$. $\boldsymbol{m}_2$ replaces $\boldsymbol{m}_1$ and Eq.(1)-(3) run again to obtain the 2nd digit of checksum (d_2). Similarly, we vary $\boldsymbol{m_2}$ to be $\boldsymbol{m}_3$ such as $[w_2, w_2, w_1, w_2, w_2, w_2, w_1, w_2, w_2, w_2, \ldots]$ to obtain 3rd digit of checksum (d_3) and vary $\boldsymbol{m}_3$ to be $\boldsymbol{m}_4$ such as $[w_2, w_2, w_2, w_1, w_2, w_2, w_2, w_1, w_2, w_2, w_2, \ldots]$ to obtain 4th digit of checksum (d_4).

Table 1 demonstrates the checksum calculation for the string of "www.fjut.edu.cn". The array of $\boldsymbol{c}$ contains the ASCII codes of characters of "www.fjut.edu.cn", as listed in 3rd column. Then the multiplier array of $\boldsymbol{m}_1$ weighs $\boldsymbol{c}$ array to store in $\boldsymbol{p}$ array using Eq.(1), as listed in 4th column. $\boldsymbol{p}$ array transforms into $\boldsymbol{t}$ array using Eq.(2), as listed in 5th column. s variable is equal to the summation of all elements in t array and d_1 variable is calculated by Eq.(3) as 1st digit of checksum. Similarly, the weighting arrays of $\boldsymbol{m}_2$, $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$ are respectively applied to obtain d_2, d_3 and d_4 as 2nd, 3rd and 4th digits of checksum. Finally, the four-digit checksum for "www.fjut.edu.cn" is assigned to $[7, 6, 5, 9]$, as listed in the last row.

TABLE 1. Checksum calculation for the string of "www.fjut.edu.cn"

i	char	$\boldsymbol{c}$	$\boldsymbol{m}_1$	$\boldsymbol{p}$	$\boldsymbol{t}$	$\boldsymbol{m}_2$	$\boldsymbol{p}$	$\boldsymbol{t}$	$\boldsymbol{m}_3$	$\boldsymbol{p}$	$\boldsymbol{t}$	$\boldsymbol{m}_4$	$\boldsymbol{p}$	$\boldsymbol{t}$
0	w	119	2	238	28	1/16	7	7	1/16	7	7	1/16	7	7
1	w	119	1/16	7	7	2	238	28	1/16	7	7	1/16	7	7
2	w	119	1/16	7	7	1/16	7	7	2	238	28	1/16	7	7
3	.	46	1/16	2	2	1/16	2	2	1/16	2	2	2	92	17
4	f	102	2	204	24	1/16	6	6	1/16	6	6	1/16	6	6
5	j	106	1/16	6	6	2	212	17	1/16	6	6	1/16	6	6
6	u	117	1/16	7	7	1/16	7	7	2	234	24	1/16	7	7
7	t	116	1/16	7	7	1/16	7	7	1/16	7	7	2	232	22
8	.	46	2	92	17	1/16	2	2	1/16	2	2	1/16	2	2
9	e	101	1/16	6	6	2	202	22	1/16	6	6	1/16	6	6
10	d	100	1/16	6	6	1/16	6	6	2	200	20	1/16	6	6
11	u	117	1/16	7	7	1/16	7	7	1/16	7	7	2	234	24
12	.	46	2	92	17	1/16	2	2	1/16	2	2	1/16	2	2
13	c	99	1/16	6	6	2	198	18	1/16	6	6	1/16	6	6
14	n	110	1/16	6	6	1/16	6	6	2	220	25	1/16	6	6
				s	153		s	144		s	155		s	131
				d_1	7		d_2	6		d_3	5		d_4	9

3.2. **Image warping of checksum during generation phrase.** Once our proposed checksum is generated according to a QR code, we can warp and paste the four digit images of checksum beside the QR code image. To perform the warping, the QR code is first required to be detected and its border is given at the same time. As shown in Fig.5, the border of a QR code is obtained and is drawn by a red rectangle. We take the right sideline of the border, i.e. the line from point q_1 to point q_5. The line of q_1-q_5 is equally divided into four parts to distribute four digits of checksum, as shown four lines of q_1-q_2, q_2-q_3, q_3-q_4 and q_4-q_5. The digit warping is carried out by a similarity transformation from a digit image to a QR code image [10].

As shown in Fig.5, the image of 1st digit of checksum of 7 warps into the QR code image with the similarity transformation with two corresponding point pairs, i.e. (p_1, q_1) and (p_2, q_2), where p_1 and p_2 are respectively the top-left and bottom-left corners in the digit image of 7. Finally, the digit image of 1st digit of checksum is warped and pasted to align with the line of q_1-q_2. A homogeneous transformation matrix is determined by the two point pairs and then the matrix is used to the affine transformation of the warping from digit image to QR code image. Similarly, the images of 2nd, 3rd and 4th digits of checksum are warped and pasted to align on lines of q_2-q_3, q_3-q_4 and q_4-q_5, respectively.

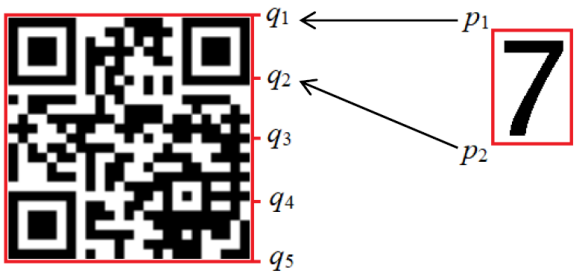


FIGURE 5. Warping of 1st digit of checksum

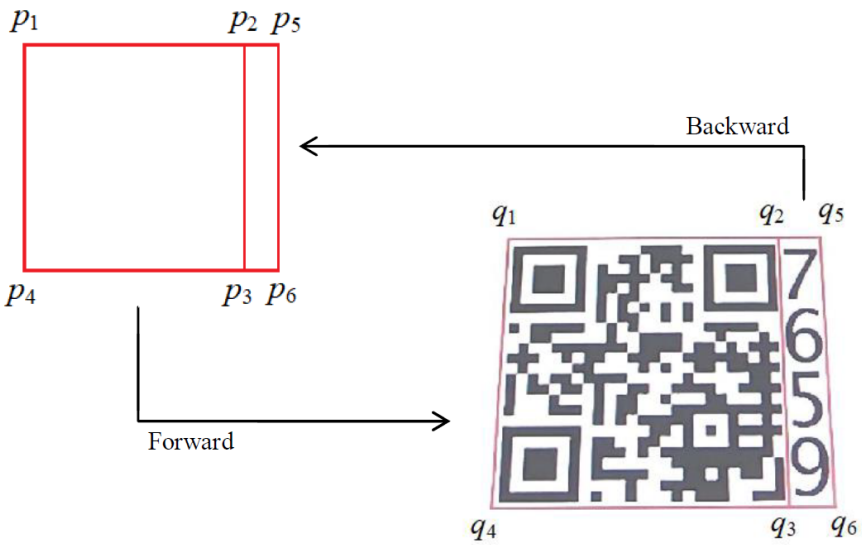


FIGURE 6. Forward and backward warping between normalized framework and captured image

3.3. **Image warping of checksum during authentication phrase.** During authentication phrase, when an image of QR code with checksum is captured by a camera, an image warping of checksum is needed for OCR. We prepare a normalized framework which can contain two images of QR code and checksum, as a square attaching a rectangle drawn by red lines, as shown in Fig.6. The linkage of four points of p_1, p_2, p_3 and p_4 encloses around the square, while the points of p_2, p_5, p_6 and p_3 link into the rectangle. The QR code is detected in a capture image and its border is enclosed by q_1, q_2, q_3 and q_4. The points of p_1, p_2, p_3 and p_4 in the normalized framework and the points of q_1, q_2, q_3 and q_4

compose the four pairs of (p_1, q_1), (p_2, q_2), (p_3, q_3) and (p_4, q_4). The four pairs can be used to compute a projective transformation matrix for the forward warping [10] from the square of normalized framework to the border of QR code. At the same time, the points of p_5 and p_6 is also warped into the points of q_5 and q_6 using the projective transformation. As a result, the quadrilateral enclosed by q_2, q_5, q_6 and q_3 can catch the four digits of checksum. Similarly, the other projective transformation matrix is computed using the four pairs of (q_2, p_2), (q_5, p_5), (q_6, p_6) and (q_3, p_3). The backward warping can perform the digits of checksum into the normalized rectangle enclosed by p_2, p_5, p_6 and p_3. Finally, the normalized digits are obtained for OCR.

3.4. **OCR for checksum digits.** After the image warping of the checksum, OCR is allowed to recognize the normalized digits. During training phrase, firstly, binarization segmentation detects digits from the checksum area, and then a feature extractor capture digit's features, such as convexity, number of holes, horizontal projection and vertical projection. Secondly, we choose SVM with RBF kernel and apply one-versus-all strategy to build an OCR classifier of 10 classes meeting $0, 1, 2, \ldots, 9$. Finally, we use a set of digit images containing $0, 1, 2, \ldots, 9$ to train our OCR classifier under a specific font. During testing phrase, the features of the four-digit checksum input into the trained OCR to return recognition results.

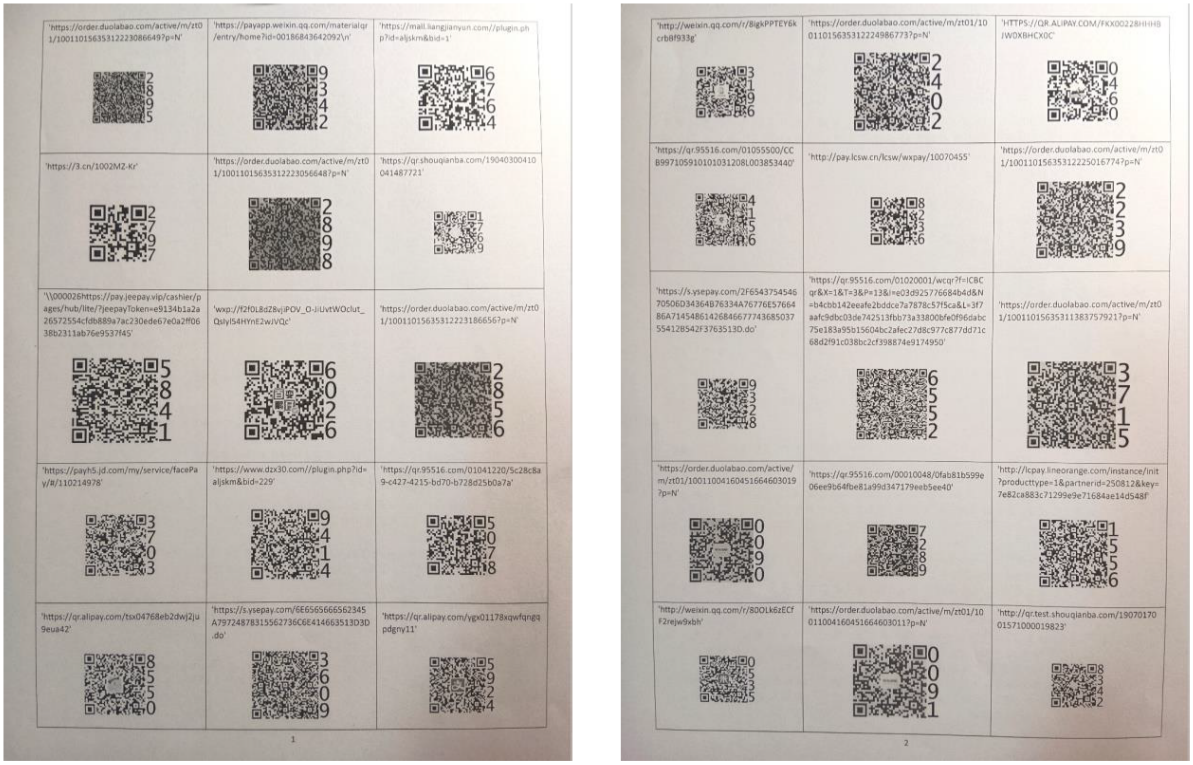
4. **Experimental Results and Comparisons.** To evaluate our proposed method, we collected 30 QR codes for payment from the internet. 30 images of QR code are listed in two tables of 5 rows by 3 columns in two pages. We computed every QR codes and generated their checksums and then attached the checksums beside the corresponding QR codes. To simulate print-and-scan scenario, we printed the two pages of the QR code and then captured two images using smartphone. As shown in Fig.7(a)-(b), each QR code with its checksum is displayed with their corresponding message texts in a single grid. Fig.8(a)-(b) demonstrated the simulation results for testing images. We first detected QR codes and drew the code borders by red lines. Then, we decoded QR codes and computed their checksums, as shown in red in the bottom-left corners of the code borders. Finally, we recognized the digit images of the checksums and displayed the results in blue in the top-right corners of the code borders. All of the checksums of the QR codes can be recognized correctly and authenticated successfully.

We take SVQR code [2] to make comparison with our proposed method in several aspects such as complexity, encryption, watermarking, construction costs and scope of application, as listed in Table 2.

Table 2. Method comparison

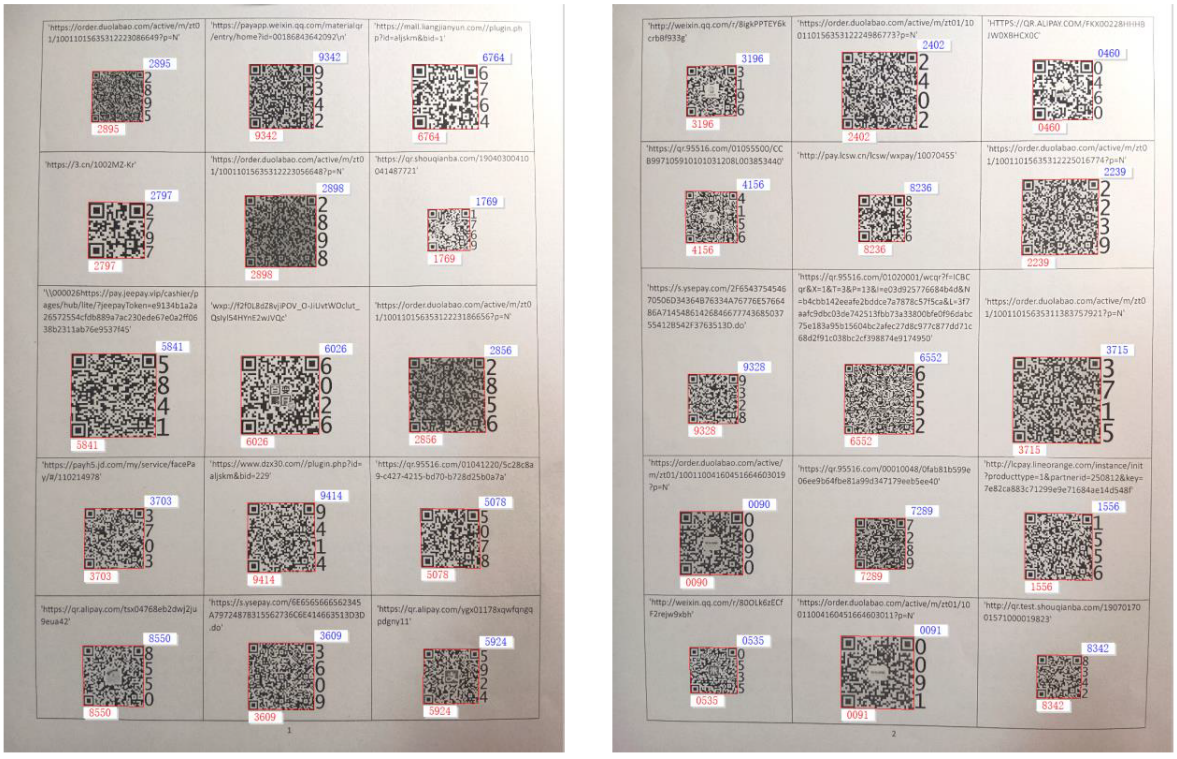
•	SVQR code	Our proposed
Complexity	Very high, multiple processes include hashing, encryption and watermarking.	Low, two steps is merely needed, including checksum generation and digit image pasting.
Encryption	Yes, the QR code's hashing value is encrypted into a signature.	No, the checksum represents as a plain text to convert into an image form.
Watermarking	Yes, the encrypted signature is watermarked to embed into QR code.	No, the image of the checksum pastes beside QR code.
Construction costs	More than in general, it needs an extra certificate authority for code encryption and decryption.	Very low, only checksum generation is needed.
Scope of application	Very limited, only useful in electronic form.	Very wide, available for print-and-scan scenario.

5. **Conclusions and Future Works.** This paper has presented a simple authentication of QR code using OCR. We developed a new checksum generation according to the message text loaded in QR code. The checksum is represented by image form and pasted beside the QR code. During the authentication, the digit image in the checksum area is detected and recognized by OCR to provide the checksum verification. The simulation results indicate that the proposed QR code can be successfully authenticated



(a) Page 1 (b) Page 2

FIGURE 7. Testing images of two sets of QR codes



(a) Page 1 (b) Page 2

FIGURE 8. Simulation results of two sets of QR codes

under print-and-scan scenario. As a result, the proposed authentication is effectively to prevent from payment theft of QR code.

The proposed checksum generation is designed by the reference of Luhn algorithm. Like other many hash functions, our method calculates hash values i.e. checksums from a huge domain to a finite value domain. The hashing collision is unavoidable that means two different message texts probably generate the same checksum. The situation limits that our approach cannot become a completely reliable authentication of QR code. However, QR codes are often provided to merchant owners or individuals by third-party payment corporations. Given a merchant QR code with its checksum, a theft want to obtain his own QR code with the same checksum, which is very difficult because the generation of QR codes is completely determined in the side of third-party payment corporations. Thefts almost cannot bypass third-party payment corporations to produce fake QR codes which disguise genus QR codes under the same checksum. Despite this, the proposed authentication method is still quite promising. The future research will be taken to discuss the problem of the checksum collisions.

Acknowledgment. The author would like to express sincere thanks to the anonymous reviewers for their invaluable comments, which truly helped towards an effective presentation of this paper.

REFERENCES

[1] https://www.163.com/dy/article/J2R3ON2F0517BACO.html
[2] S. J. Liu, J. Zhang, J. S. Pan, and C. J. Weng, *SVQR: A Novel Secure Visual Quick Response Code and Its Anti-counterfeiting Solution*, Journal of Information Hiding and Multimedia Signal Processing, Vol.8, No.5, pp.1132-1140, 2017.
[3] S. J. Liu, J. Zhang, J. S. Pan, and C. J. Weng, *A Novel Information Imbedding and Recovering Method for QR Code based on Module Subdivision*, Journal of Information Hiding and Multimedia Signal Processing, Vol.9, No.2, pp.515-522, 2018.
[4] H. Cai, B. Yan, J. S. Pan, and J. L. Ye, *Print-Scan Resistant Two-Level QR Code*, Journal of Information Hiding and Multimedia Signal Processing, Vol.10, No.2, pp.300-312, 2019.
[5] A. N. Jasim and S. H. Abdulhussain, *A Comprehensive Review of Digital Watermarking techniques: Applications, Characteristics, Classification and Related Aspects*, Journal of Information Hiding and Multimedia Signal Processing, Vol.16, No.4, pp.1209-1251, 2025.
[6] J.-S. Pan, M. Zhu and S.-C. Chu, *Robust Digital Watermarking with Parallel Compact Sparrow Search Algorithm Applied for QR Code*, Journal of Information Hiding and Multimedia Signal Processing, Vol.13, No.2, pp.124-144, 2022.
[7] S. A. Alsuhibany, *Innovative QR Code System for Tamper-Proof Generation and Fraud-Resistant Verification, Sensors*, Vol.25, No.13, pp.3855-3873, 2025.
[8] P. N. San Agustin-Crescencio, L. Hernandez-Gonzalez, P. Guevara-Lopez, O. U. Juarez-Sandoval, J. Ramirez-Hernandez and E. S. Estevez-Encarnacion, *A Comprehensive QR Code Protection and Recovery System Using Secure Encryption, Chromatic Multiplexing, and Wavelength-Based Decoding*, Applied Sciences, Vol.15, No.17, pp.9708-9722, 2025.
[9] Luhn, H. P., *Computer for Verifying Numbers*, US patent 2950048A, 1960.
[10] J. E. Solem, Programming Computer Vision with Python, O'Reilly Media Inc., 2012.