mQRCode: Secure QR Code Using Nonlinearity of Spatial Frequency in Light

Hao Pan*
Shanghai Jiao Tong University
panh09@sjtu.edu.cn

Yi-Chao Chen*
Shanghai Jiao Tong University
yichao@sjtu.edu.cn

Lanqing Yang
Shanghai Jiao Tong University
yanglanqing@sjtu.edu.cn

Guangtao Xue
Shanghai Jiao Tong University
gt_xue@sjtu.edu.cn

Chuang-Wen You
National Taiwan University
cwyou2004@gmail.com

Xiaoyu Ji
Zhejiang University
xji@zju.edu.cn

ABSTRACT

Quick response (QR) codes are becoming pervasive due to their rapid readability and the popularity of smartphones with built-in cameras. QR codes are also gaining importance in the retail sector as a convenient mobile payment method. However, researchers have concerns regarding the security of QR codes, which leave users susceptible to financial loss or private information leakage. In this study, we address this issue by developing a novel QR code (called mQRCode), which exploits patterns presenting a specific spatial frequency as a form of camouflage. When the targeted receiver holds a camera in a designated position (e.g., directly in front at a distance of 30 cm from the camouflaged QR code), the original QR code is revealed in form of a Moiré pattern. From any other position, only the camouflaged QR code can be seen. In experiments, the decryption rate of mQRCode was $> 98.6\%$ within 10.2 frames via a multi-frame decryption method. The decryption rate for cameras positioned 20° off axis or $> 10cm$ away from the designated location dropped to 0%, indicating that mQRCode is robust against attacks.

CCS CONCEPTS

• **Security and privacy** → **Domain-specific security and privacy architectures**; *Privacy protections*; • **Human-centered computing** → Ubiquitous and mobile computing systems and tools.

KEYWORDS

screen-camera communication; secure QR code; nonlinearity

ACM Reference Format:
Hao Pan, Yi-Chao Chen, Lanqing Yang, Guangtao Xue, Chuang-Wen You, and Xiaoyu Ji. 2019. *mQRCode*: Secure QR Code Using Nonlinearity of Spatial Frequency in Light. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19), October 21–25, 2019, Los Cabos, Mexico.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3300061.3345428

*Both authors contributed equally to this research.

MobiCom '19, October 21–25, 2019, Los Cabos, Mexico

ACM ISBN 978-1-4503-6169-9/19/10…$15.00
https://doi.org/10.1145/3300061.3345428

1 INTRODUCTION

The matrix barcodes known as quick response (QR) codes are a global phenomenon. QR codes are two-dimensional barcodes that visually encode bits of information in the form of black blocks on a square white grid. The data in a QR code can be accessed by photographing the QR code and processing the image file using a decoder. QR codes can be easily read by built-in cameras of smartphones.

It is easy to generate QR-code images for authentication via mobile applications (APPs). This type of authentication is commonly used for mobile payment systems in retail stores, gate access control in buildings, and even unmanned book rental systems in libraries. For example, the Alipay [3] system in Fig. 1 is an authentication system based on QR code. A user can confirm a transaction using a password and then the APP encrypts the account and payment information to generate a QR code displayed on the screen of the user's mobile device, e.g., a smartphone. The retailer can use Electronic Cash Register (ECR), a terminal machine equipped with a camera, or a smartphone to capture the QR code and decrypt the messages embedded in the code. Afterwards, a transaction request is transmitted to the back-end of the Alipay system. When the transaction is approved, the user receives a notification [4].

Unfortunately, QR code systems are susceptible to security risks in the form of Replay attacks [32, 38, 49] and Synchronized Token Lifting and Spending (STLS) attacks [6]. In both attacks, an attacker sneakingly obtains the victim's QR code

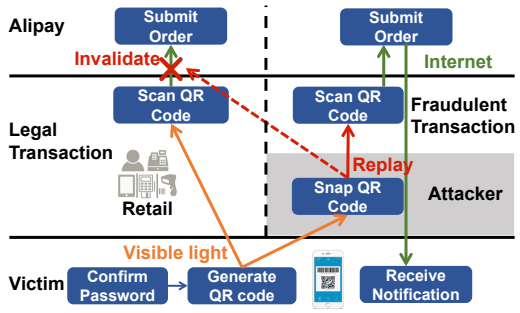


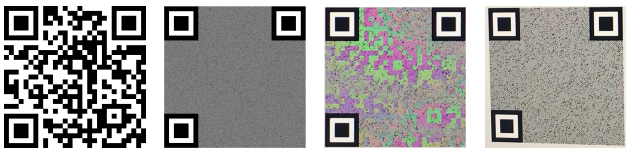
Figure 1: Left is the QR-code-based authentication system for Alipay; Right is the process of STLS attack on QR-code-based payment.

to make other payment or access the victim's private information. As shown in Fig. 1, once the attacker obtains an image of the victim's QR code, the attacker does not need to decrypt the message embedded in the QR code nor know the victim's password. The attacker simply presents QR code to the retail for payment. Detailed threat model is presented in Sec.3.

Due to the widespread adoption of QR codes by retailers (e.g., Walmart and Starbucks), financial organizations (e.g., Paypal and AliPay), and mobile APPs (e.g., WeChat and Weibo), this kind of attack can incur tremendous financial losses [35, 41] or the wide-scale theft of private information [46, 54]. It is the inherent characteristics of the communication channels that allow these attacks. Furthermore, QR codes do not support a secure mutual challenge-response protocol, due to the fact that information can be transmitted in only one direction.

In this study, we develop a novel QR code system, hereafter referred to as MQRCODE[1] to resist such attacks. The proposed scheme exploits nonlinearities in the spatial frequency of light rays to camouflage QR codes from the communication channel. MQRCODE relies only on the existing physical characteristics of the camera and display for encryption; i.e., no additional communication channels or hardware are required. When a QR code is generated (like Fig. 2(a)), MQRCODE encrypts it within a pattern that is regarded as noise (from the perspective of the human visual system) using a designated spatial frequency. An example of the resulting *mQR* code is shown in Fig. 2(b). The image of *mQR* code captured by the receiver from a display is projected onto the image sensor in the camera; however, this projection includes scaling, translation, rotation determined by the relative position between the display. If the camera (i.e., the targeted information receiver) is held precisely at the designated position, i.e., from right distance and angle, nonlinearities in spatial frequency between the projected *mQR* code and the Color Filter Array (CFA) of the camera allow the original QR

[1]The camouflaged QR code image is referred to as *mQR* code.



(a) (b) (c) (d)

Figure 2: *mQR* code can be naturally revealed when the camera is held in the designated position. (a) Original QR code. (b) *mQR* code: Encrypted QR code. (c) Picture taken at designated position. (d) Picture taken at wrong position (off by 15°).

code to be revealed as a Moiré pattern, as shown in Fig. 2(c) [2]. However, if an attacker captures an image of the *mQR* code from any other position, the scaling, translation, rotation of the *mQR* code would be incorrect, with the result that the camera would be unable to make out the intended Moiré pattern. Thus, the physical limitation imposed by the position of the camera would prevent any would-be attackers from decrypting the *mQR* code.

MQRCODE provides a number of benefits. First, MQRCODE is a software-based solution, which requires no communication channels or additional hardware. Second, encryption and decryption rely on the relative position between the *mQR* code image and the camera. Thus attackers are prevented from decrypting the *mQR* codes due to the fact that they cannot occupy the same physical space as the would-be victim. Third, the computational overhead of decryption is low, as it involves simple image processing based on the nonlinear optical interaction between camera and *mQR* code. This makes it possible for deployment on the majority of smartphones currently on the market as well as applications with limited computational resources, such as secure IoT communications devices. Moreover, other applications such as key exchanges and device paring can also be implemented via our MQRCODE system to obtain high security.

We have intensively evaluated a prototype of MQRCODE to verify its effectiveness and robustness on a variety of displays, smartphones, and a Raspberry Pi. Our experiments show that the decoding rate of any unauthorized camera at a distance of $> 10cm$ away from the designated location or at a view angle of $> 20°$) drops to 0, thereby ensuring that the would-be attacker is unable to obtain a usable image. We also conduct a user study of 20 participants with various ages, genders, and occupations to demonstrate the usability of MQRCODE.

The contributions of this work include the following:

- We propose a novel optical encryption method for QR codes based on the nonlinearities in spatial frequency.

[2]A supplementary video shows how an *mQR* code is naturally revealed: https://youtu.be/D10J7WCik8U. Note that Fig. 2(b) is scaled specifically for this manuscript; thus, readers cannot decrypt it using their cameras.

(a) $p(u) = 0.5 + 0.5cos(u)$ (b) $\phi_1(x, y) = x$, $p(\phi_1(x, y))$

Figure 3: Example of periodic and phase functions.

- We propose a model to describe the Color Filter Array for use in camouflaging spatial patterns via phase modulation and frequency modulation.
- We propose two robust decryption schemes for the reconstruction of original QR codes from captured Moiré patterns.
- We implemented the mQRCode system in Android, iOS, and Raspbian (the operating system of RPi), and performed extensive experiments to assess the feasibility and limitations of the proposed mQRCode.

The remainder of this paper is organized as follows: In Sec. 2, we discuss the nonlinearities in spatial frequency and how they can affect camera systems. In Sec. 4, we outline the system flow. The encryption and decryption schemes are detailed in Sec. 5 and Sec. 6, respectively. Sec. 7 outlines the extensive experiments used to evaluate the proposed scheme. Sec. 8 shows the results of the user study. In Sec. 9, we discuss the limitations of mQRCode. In Sec. 10, we present the related works. Finally, conclusions are presented in Sec. 11.

2 BACKGROUND

2.1 Spatial Frequency

Spatial frequency is a characteristic of any structure that is periodic across its position in space. In this paper, we consider a bi-dimensional (2D) spatial structure with curvilinear pattern, which can be described using a frequency term and a phase term as follows:

$$m(x, y) = p(\phi(x, y)) \tag{1}$$

where $m(x, y)$ represents the magnitude at a 2D coordinate (x, y) (i.e., the color of an image), $p(\cdot)$ is a periodic function representing the frequency of the pattern, and $\phi(x, y)$ is a phase function representing the angle of the pattern. For example, Fig. 3(a) shows the periodic function using a cosine wave with a frequency of $1/2\pi$. When the phase function is set to $\phi(x, y) = x$, we obtain a spatial pattern with repetitive horizontal lines, as shown in Fig. 3(b).

2.2 Nonlinearity of Spatial Frequency

When two spatial patterns overlap, the nonlinear optical interaction between the patterns creates an additional visible layer (referred to as a Moiré pattern) over the original patterns. In a gray-scale image, each point (x, y) is assigned

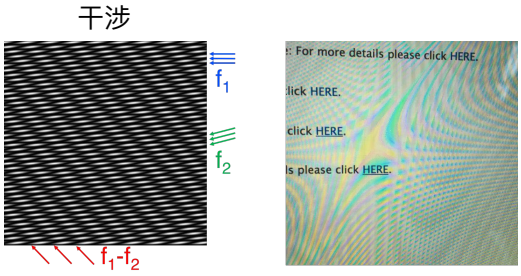


Figure 4: Left is nonlinear optical interaction of two patterns with frequency f_1 and f_2; Right is nonlinear optical interaction when taking a picture of a display using a camera.

a value between 0 and 1 indicating its light reflectance: 0 for black (i.e., no reflected light), 1 for white (i.e., full light reflectance), and intermediate values for shades in between. For color images, the same idea can be applied to each color channel. The fact that the superposition of black and any other shade is always black suggests a multiplicative model for the superposition of images. Therefore, assume m is the superposition of two layers m_1 and m_2:

$$m(x, y) = m_1(x, y) \times m_2(x, y) \tag{2}$$

The multiplicative model produces nonlinearities in the spatial frequency. For example, when m_1 and m_2 use cosine functions with frequency f_1 and f_2 as periodic functions:

$$\begin{aligned} m =& m_1 \times m_2 \\ =& (a_1 + b_1 cos(2\pi f_1 t)) \times (a_2 + b_2 cos(2\pi f_2 t)) \\ =& a_1 a_2 + a_1 b_2 cos(2\pi f_2 t) + a_2 b_1 cos(2\pi f_1 t) \\ & + b_1 b_2 cos(2\pi (f_1 + f_2) t) + b_1 b_2 cos(2\pi (f_1 - f_2) t) \end{aligned}$$

We find that the combination result includes two additional frequencies $(f_1 + f_2)$ and $(f_1 - f_2)$. Human eyes are more sensitive to low frequency signals; therefore, frequency $(f_1 - f_2)$ is easier to observe, as shown in the left of Fig. 4.

Similarly, when m_1 and m_2 are general curvilinear patterns, where $m_1 = p_1(\phi_1(x, y))$ and $m_2 = p_2(\phi_2(x, y))$, the spectrum of their superposition m can be computed based on convolution theorem [55]:

$$M(x, y) = M_1(x, y) \otimes M_2(x, y) \tag{3}$$

where M, M_1, and M_2 represent the Fourier Transform of m, m_1 and m_2, respectively; and the $\otimes$ operator represents the 2D convolution. According to Moiré theorem [5], the periodic function and phase function are independent and can therefore be computed separately. Let m_{nl} represent the evident nonlinear component resulting from the superposition of m_1 and m_2 with frequencies $(f_1 - f_2)$. Due to the fact that m_{nl} is also a curvilinear pattern, $m_{nl} = p_{nl}(\phi_{nl}(x, y))$ is in accordance with Eq. 1. We then decompose Eq. 3 and compute its periodic function $p_{nl}(u)$ and phase function $\phi_{nl}(x, y)$ as follows:

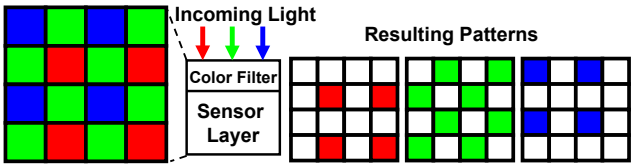


Figure 5: Profile of sensor with the Bayer arrangement of color filters.

$$p_{nl}(u) = IFT(FT(p_1(u)) \cdot FT(p_2(-u)))$$
$$\phi_{nl}(x,y) = \phi_1(x,y) - \phi_2(x,y) \quad (4)$$

where $FT(T)$ and $IFT(T)$ are the Fourier Transform and Inverse Fourier Transform of input T, respectively; and $p_1(u)$, $p_2(u)$, $\phi_1(x,y)$, and $\phi_2(x,y)$ are the corresponding periodic and phase functions for m_1 and m_2.

2.3 Nonlinearity in Camera Systems

Cameras are nonlinear systems susceptible to Moiré patterns when used to take pictures of spatial patterns. The right part in Fig. 4 presents an example image of a display obtained using a camera, showing repetitive curving lines superimposed on the display. This nonlinearity is caused by the CFA on the camera sensor. The CFA is a mosaic of tiny color filters placed over pixel sensors on an image sensor to enable the capture of color information. The Bayer filter is the most common filter [56] used in the cameras built into smartphones. The example in Fig. 5 provides information related to the intensity of light in red, green, and blue in a 4×4 array.

When capturing a picture of a display, the pixels of the display sensors projected onto camera form a spatial pattern layer with spatial frequency f_1, and the CFA forms the other layer with frequency f_2. When the camera is positioned at an appropriate distance and angle to the display, the difference between spatial frequencies $(f_1 - f_2)$ falls within an observable frequency range, such that the nonlinear optical interaction appears as a rippled image (right part of Fig. 4).

To encrypt and decrypt QR codes, mQRCode exploits this nonlinear optical interaction between the CFA and the camouflage pattern, as detailed in the following sections.

3 THREAT MODEL

We envision a mobile payment scenario where a victim is going to pay with mobile payment softwares such as Alipay. In order to finish a payment, the victim should display the payment QR code on his/her smartphone and then the cashier scans the QR code to finish the transaction, as depicted in Fig. 1. Meanwhile, there is an attacker whose goal is to obtain the victim's QR code without the victim's awareness to steal money. We assume that the attacker can either physically

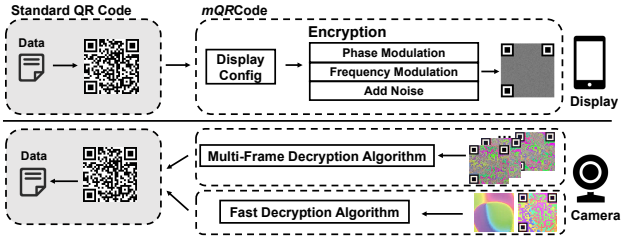


Figure 6: mQRCode system flow.

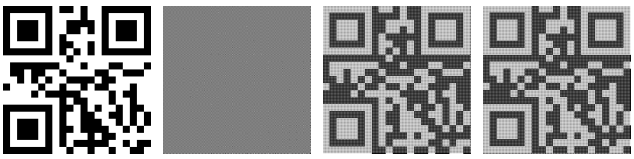
get close to the victim to capture the QR code by carrying a smartphone or a digital camera or use pre-deployed cameras, like on the ceiling, in the wall sockets, or in some decorations, to take a sneak shot of the victim's QR code. There is high possibility for the attacker to capture the QR code because people tend to show their payment QR codes before they get in front of the checkout counter [41].

Once the QR code is captured, the attacker can immediately transfer it to a remote server for a fraudulent transaction. Note that the fraudulent transaction needs to be done before the victim completes the transaction because a QR code can only be used for one transaction. We conduct a quick survey that how long users will open the payment QR code before checkout via an online questionnaire. The result shows that 85% among 30 users tend to have the payment QR code on the phone screen 60 seconds before a clerk scans it. We assume the time span for the attack is enough because the attacker can interrupt the legitimate payment process by physically hindering it or break the progress with social engineering methods such as talking to the cashier to delay the scanning. After the fraudulent transaction, the victim receives a notification about the transaction; however, at the time, the attack has succeeded – the attacker has finished the fraudulent transaction. Moreover, in many reported attacks, the victim simply ignored the notification or was not aware that it was an attack [35, 41].

In the threat model, we assume the attacker cannot: (1) get access to the victim's payment device or install malware on it, (2) obtain pixel-level photograph of the QR code as such a camera and lens are too large to carry (see Sec. 9).

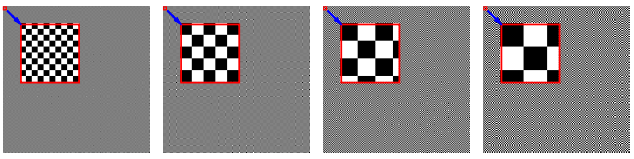
4 SYSTEM FLOW

The objective behind the development of mQRCode is to enhance the security of information transmitted via QR code. Fig. 6 illustrates the system architecture, which comprises two parts: a sender (e.g., a smartphone display) and a receiver (e.g., a smartphone camera). When the sender generates a standard QR code, mQRCode checks the display configuration (e.g., resolution and pixel size) before applying phase modulation (Sec. 5.2) and frequency modulation (Sec. 5.3) on encrypting QR code. The encrypted QR code (i.e., *mQR* code)



(a) QR Code. (b) Green filter. (c) Red filter. (d) Blue filter.

Figure 7: Encrypt a QR code using color filters in Bayer Filter. The green filters in each 2×2 array are symmetric, such that variations in phase make them more difficult for the human eye to distinguish.



(a) $f = 1$. (b) $f = \frac{1}{2}$. (c) $f = \frac{1}{3}$. (d) $f = \frac{1}{4}$.

Figure 8: Example of using various frequencies f to communicate at various distances.

is also camouflaged with noise to handle phase discontinuities (Sec. 5.4). The resulting *mQR* code that appears on the display is highly secure.

On the receiver side, we provide two decryption schemes for different application scenarios. For handheld mode, we hold the camera at a designated distance and angle from the *mQR* code, whereupon our multi-frame decryption scheme (Sec. 6.2) is applied to reconstruct the QR code. For fixed scanners commonly found in stores, the display with the *mQR* code is placed under the fixed scanner, whereupon the fast decryption scheme (Sec. 6.3) is applied. After the original QR code is reconstructed, the standard QR code decoder is used to obtain the embedded data.

We detail the encryption and decryption schemes in the following two sections.

5 ENCRYPTION

mQRCode exploits the nonlinear optical interaction between the CFA and pattern used to camouflage the QR code. According to Eq. 2, without a loss of generality, we assume that the spatial pattern of CFA is $m_{cfa}(x, y)$ and the original QR code (also the decrypted QR code) is $m_{dec}(x, y)$. The goal of the encryption process is to compute the encrypted QR code image $m_{enc}(x, y)$, such that $m_{dec}(x, y) = m_{cfa}(x, y) \cdot m_{enc}(x, y)$.

5.1 Color Filter Array Model

We first model the CFA by formulating $m_{cfa}(x, y) = p_{cfa}(\phi_{cfa}(x, y))$ from Eq. 1. In the Bayer filter (Fig. 5), green filters are located within diagonal grids in each 2×2 array, whereas blue and red filters occupy the remaining locations. Instead of modeling all three channels, mQRCode models only the green filter. This is done for two reasons. First, QR codes contain only black and white blocks; therefore, one color filter is enough for decryption.

Second, the green filter in a 2×2 array is symmetric, such that variations in phase make them more difficult for the human eye to distinguish. As described in Sec. 5.2, mQRCode alternates phases in the representation of black and white QR code blocks. Allowing the generated spatial pattern to be distinguishable with alternations in phases would defeat the purpose of the model (i.e., for encryption). Fig. 7 presents examples of encrypted patterns using green, red, and blue filters. The theoretical explanation of the results is detailed in Appendix A.

In this study, the green filter is modeled as follows:

$$\begin{aligned} m_{cfa}(x, y) &= p_{cfa}(\phi_{cfa}(x, y)) \\ p_{cfa}(u) &= 0.5 + 0.5cos(2\pi u) \\ \phi_{cfa}(x, y) &= ((x + y) \text{mod} 2)/2 \end{aligned} \tag{5}$$

where $m_{cfa}(x, y)$ represents the color reception of the green filter at coordinate (x, y) on the image sensor, $p_{cfa}(u)$ represents the periodic function, and $\phi_{cfa}(x, y)$ represents the phase function. Since x and y range from 1 to the image height/width, values of the phase function become 0 or 0.5. After application of the periodic function, $m_{cfa}(x, y)$ becomes 1 (i.e., receive all light on the green filter) in the diagonal grids in each 2×2 array or 0 (i.e., filter out all light).

5.2 Phase Modulation

To compute $m_{enc}(x, y) = p_{enc}(\phi_{enc}(x, y))$, we let $p_{enc}(u)$ equal $p_{cfa}(u)$:

$$p_{enc}(u) = 0.5 + 0.5cos(2\pi u) \tag{6}$$

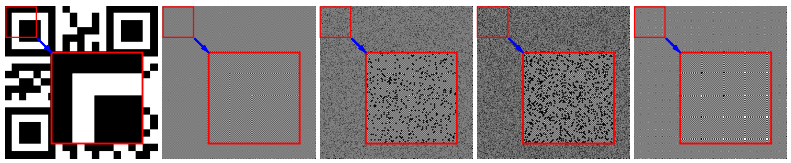
in order to enlarge the contrast of resulting Moiré pattern [52]. Phase modulation is applied by mapping black and white blocks in QR codes to different phases. Due to the fact that m_{dec} (i.e., original QR code) and m_{cfa} (modeled in Sec. 5.1) are known, combining Eq. 2 and 4 we learn the following:

$$\begin{aligned} m_{dec}(x, y) &= p_{dec}(\phi_{dec}(x, y)) \\ &= p_{dec}(\phi_{cfa}(x, y) - \phi_{enc}(x, y)) \\ \implies \phi_{enc}(x, y) &= \phi_{cfa}(x, y) - p_{dec}^{-1}(m_{dec}(x, y)) + 2k\pi, k \in \mathbb{Z} \end{aligned}$$

where p_{dec}^{-1} represents the inverse function of p_{dec}, which maps intensity values to the corresponding phases. The $2k\pi$ term has no impact on the encrypted image m_{enc} because mQRCode uses *cosine* as the periodic function (Eq. 6).

5.3 Frequency Modulation

The numerous applications to which mQRCode could be applied no doubt require that image capture be conducted at different distances from the QR code. For example, in retail



(a) QR code. (b) *mQR* code. (c) 10% dot. (d) 20% dot. (e) Lines.

Figure 9: Addition of noise or camouflaging lines to mitigate observable lines caused by abrupt phase changes. In (b), the zoomed-in blocks into black and white figure shows observable boundaries due to abrupt phase changes.

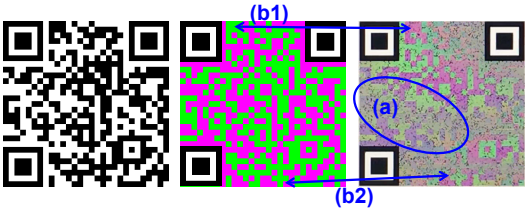


Figure 10: Challenges of decrypting *mQR* codes: (a) blur; (b) phase inversion phenomenon.

situations, the distance between a QR code and a scanner is usually less than $50cm$. When using public displays [7, 23], the distance can range from $10cm$ to $5m$.

To enable support for various communication ranges, we extended Eq. 5 to modulate the frequency of the generated spatial patterns. According to camera pinhole theory [44], the size of an object projected onto a camera sensor is inversely proportional to the distance between the object and the camera sensor as: $S_{cam} = \frac{S_{object} \times L_{focal}}{D}$, where S_{cam} is the size of the object in the captured image, S_{object} is the size of the object in the real world, L_{focal} is the focal length of the camera, and D is the distance between the camera and the object. Although the pinhole camera model does not account for lens distortion, we can derive an estimate with regard to communication distance.

Recall that the spatial frequency of a Moiré pattern is $(f_1 - f_2)$. In our scenario, f_1 is the spatial frequency of the camouflage pattern and f_2 is that of the CFA. Human eyes are more sensitive to low frequency signals; therefore, Moiré patterns become more evident when f_1 is close to f_2. When mQRCode is required to communicate at a longer distance, we can reduce the spatial frequency of the camouflaging pattern so that it remains close to f_2 when projected onto the camera sensor.

We therefore extend Eq. 5 by introducing the frequency term f, as follows:

$$\begin{aligned} m_{cfa}(x,y) &= p_{cfa}(\phi_{cfa}(x,y)) \\ p_{cfa}(u) &= 0.5 + 0.5cos(2\pi u) \\ \phi_{cfa}(x,y) &= ((\lceil xf \rceil + \lceil yf \rceil) \text{mod} 2)/2 \\ f &\in \{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, ...\} \end{aligned} \tag{7}$$

Fig. 8 shows examples of *mQR* codes in which frequency modulation is applied to adjust the communication distance. One drawback we can see in Fig. 8 is that when we use a lower frequency, the boundary of QR code blocks becomes more evident due to the abrupt phase change. We address the issue in Sec. 5.4.

5.4 Phase Discontinuity

At the boundary between white and black QR code blocks, the phase in *mQR* code also changes. An abrupt phase change may cause observable horizontal or vertical lines in the encrypted *mQR* code as shown in Fig. 9(b). This problem can be mitigated by adding noise (Fig. 9(c) and 9(d)) or camouflaging lines (Fig. 9(e)). In practice, adding 10% noise is sufficient for *mQR* codes displayed on a smartphone.

6 DECRYPTION

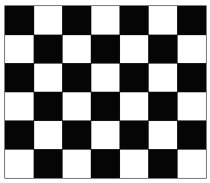
Decrypting the *mQR* code requires that the user holds the camera in a designated position, whereupon the Moiré effect reveals the original QR code, as shown in Fig. 14(a). However, using this image directly to reconstruct the original QR code can be difficult, due to the existence of blurred portions and phase inversion. Examples of blurred portions are presented in Fig. 10(a). Due to the effects of phase inversion, blocks with the same color in the original QR code (i.e., blocks which are modeled using the same phase in *mQR* code) may end up exhibiting different colors in the Moiré pattern, as shown in Fig. 10(b). Specifically, (b1) shows black blocks of the original QR code mapped to purple in the captured picture, whereas (b2) shows black blocks mapped to green.

In this section, we propose a simulation-based analysis to better understand the phenomena and two decryption schemes for different usage scenarios.

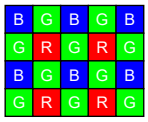
6.1 Phase Inversion

Definition: phase inversion refers to the phenomenon in which blocks of the same color in the original QR code (i.e., blocks modeled using the same phase in *mQR* code) end up exhibiting different colors in the resulting Moiré pattern. We adopted simulation-based analysis to illustrate how phase inversion occurs. Fig. 11(a) shows part of an *mQR* code in which each block represents a pixel. The pixels are either black or white, based on the phase modulation scheme (Eq. 5.2). Fig. 11(b) presents the Bayer CFA of a camera in which each photosensor captures red, green, or blue light.

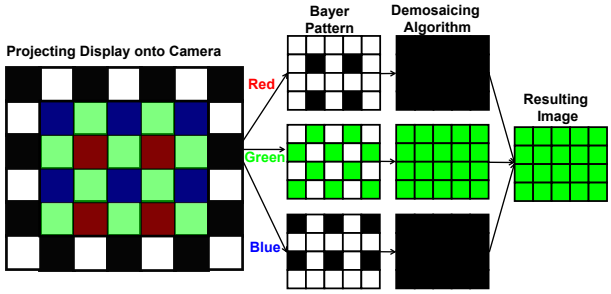
In an ideal scenario, placing a camera with no lens distortion precisely in the designated position will allow the perfect



(a) Encrypted image on display.



(b) Bayer CFA in typical camera.



(c) Alignment that results in green pattern.

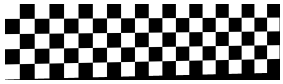
Figure 11: Simulation-based analysis illustrating Moiré patterns in encrypted images.

alignment of pixels on the display, as shown in Fig. 11(c). The raw output from a photosensor is referred to as a Bayer pattern image. Obtaining a full-color image requires a variety of demosaicing algorithms [17, 21, 26, 30], which interpolate red, green, and blue values for each pixel.

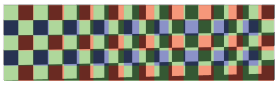
The three channels are then combined to produce the image as it should appear to the eye. In this example, only the green channel captures high-intensity light from the display, such that we see only green. Similarly, if we shift the display by one pixel, only the red and blue channels capture high-intensity light, such that we see a combination of the two as purple. In this ideal scenario, the entire image of the captured *mQR* code provides consistent phase mapping. In other words, black blocks from the original QR code are mapped to either green or purple across the entire image (as illustrated in the middle figure in Fig .10).

However, in cases where there is lens distortion or the camera is not placed precisely in the designated position, the display that is projected onto the camera is somewhat distorted. In the example in Fig. 12, the display is projected onto the photosensor plane with the camera is rotated by 1°. We then compute the Bayer pattern image of the projected image, perform interpolation to obtain the full-color image, and combine the three channels to simulate the image captured by the camera. We can see that the resulting Moiré pattern has green on the left and purple on the right. The simulation clearly illustrates how phase inversion occurs.

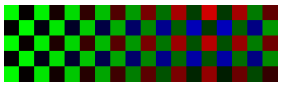
We use the proposed encryption scheme to encrypt a white image (Fig. 13(a)) and obtain its Moiré pattern (Fig. 13(b)).



(a) Distorted image.



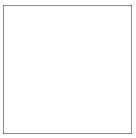
(b) Distorted image overlap with Bayer pattern.



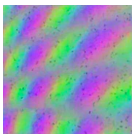
(c) Bayer Pattern Image.

(d) After interpolation.

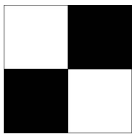
Figure 12: Simulation-based analysis illustrating phase inversion.



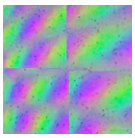
(a) Image A with only a white block.



(b) Image A obtained using a camera.



(c) Image B with 2×2 black and white blocks.



(d) Image B obtained using a camera.

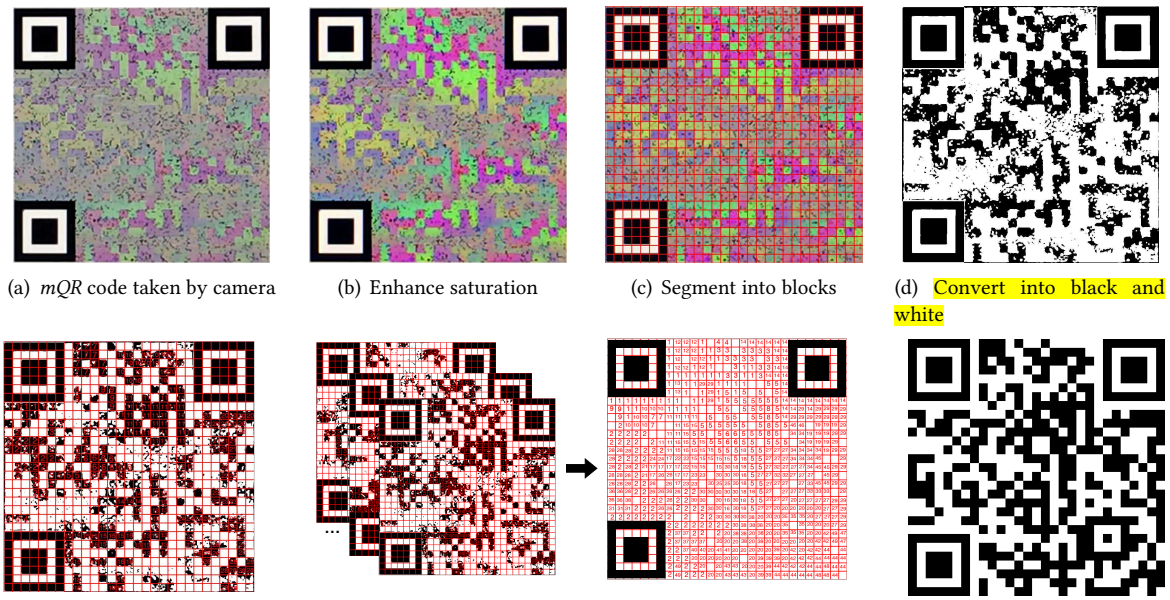
Figure 13: Illustration of difference in Moiré patterns in Images A and B under the same display and camera conditions.

We find that even the encrypted image sharing the same phase information, its corresponding Moiré pattern has more than one color, which illustrates the challenge to design a decryption scheme.

6.2 Multi-Frame Decryption Scheme

We have demonstrated how lens distortion and imprecise camera placement can result in phase inversion, which can hinder efforts to decrypt an *mQR* code using a single frame. Our simulation-based analysis suggests that if the camera distortion matrix and the position of the camera are known, then it should be possible to model the phase inversion and subsequently compensate for any changes in color mapping. However, the modeling of phase inversion requires a priori camera calibration as well as information pertaining to the precise position of the camera. The difficulty in obtaining that kind of information more or less precludes model-based phase inversion cancellation.

To address the above issues, we first propose the mQRCode decryption algorithm which utilizes multiple continuous video frames. The algorithm is based on the observation that when a user holds the camera, there is inevitably a certain amount of camera shake (usually less than $2 \sim 3mm$) [14]. The shifts induce between frames cause blurring and phase inversion occur in various regions of the captured *mQR* code images. Therefore, the original QR code can be reconstructed by taking into account the differences between multiple



(a) *mQR* code taken by camera (b) Enhance saturation (c) Segment into blocks (d) Convert into black and white

(e) Label adjacent blocks with the same color (f) Combine multiple frames (g) Color blocks with black and white

Figure 14: Multi-frame decryption process.

frames. The proposed multi-frame decryption algorithm includes the following steps:

Enhancing color saturation: Fig. 14(a) presents a picture of an *mQR* code obtained using a digital camera. We first enhance the color saturation to enhance contrast among green, red, blue. This is achieved by converting the RGB image to hue, saturation, and value (HSV) coordinates and maximizing the saturation dimension. The picture in Fig. 14(b) illustrates the results of saturation enhancement.

Segmentation: The size of the QR code is determined by its version, ranging from 21×21 blocks (version 1) to 177×177 blocks (version 40). The standard QR code contains three locator marks in fixed locations to allow the QR decoder to identify the presence of a QR code, recognize the QR code version, and calibrate the image. mQRCode leaves these locators untouched. In fact, we use these locators to create perspective cutting lines to modify slanted squares into standard squares. The width and height of the locator marks are then used to compute the size of each QR code block. A segmented *mQR* code is presented in Fig. 14(c).

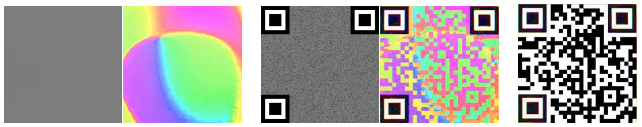
Conversion to black and white: In Sec. 5, we describe the decryption of QR codes by modeling green filters and modulating the phase function to generate *mQR* codes. QR code blocks with the different colors are assigned different phases, resulting in either green or purple separation in the Moiré pattern, as shown in Fig. 14(b). However, phase inversion alters the color mapping in the spatial domain.

To reliably identify blocks with the same phase, we separate green from purple by thresholding the green channel and converting the image to black and white. In other words, when the green intensity of a pixel is higher than a given threshold, then the pixel is changed to white; otherwise, the pixel is changed to black, as shown in Fig. 14(d).

Classification of blocks: After each block is changed to black or white, noise can result in both black and white pixels in a given block. Thus, in classifying each block as black or white, we calculate the proportion value c of black pixels in each block. When c is larger than a given threshold (in this case 0.8), the block is classified as black; otherwise, the block is classified as white.

Labeling adjacent blocks with the same color: Two adjacent black blocks probably have the same phase in *mQR* code. Thus, we loop through all of the black blocks and label them using an index. Adjacent blocks that are both black are labeled using the same index, as shown in Fig. 14(e).

Combining multiple frames: The above steps are repeated for each incoming frame. The labels from the new frame are then combined with existing labels from previous frames as follows: If a block does not have an existing label or is assigned a label in the new frame, then the block is assigned a new label. If a block has an existing label $index_{old}$ and is assigned another label $index_{new}$ in the new frame, then we search among existing frames for blocks with label $index_{old}$ and blocks with label $index_{new}$ in the new frame, and assign them a new label. We continue combining new frames until either all of the blocks are labeled or all of the blocks surrounding an unlabeled block are labeled. An example is presented in Fig. 14(f).



(a) Encrypted reference image and its Moiré pattern. (b) *mQR* code and its Moiré pattern. (c) Decryption result.

Figure 15: Fast decryption process.

Coloring blocks: Each block is then colored black or white in accordance with the labels. The colors of the locator marks are known; therefore, we begin by coloring their neighbors. The rules for color blocks are as follows: i) If two adjacent blocks have the same label, then they are drawn using the same color, and ii) if two adjacent blocks have different labels, then they are drawn using different colors. The original QR code is then recovered after all of the blocks have been colored, as shown in Fig. 14(g).

6.3 Fast Decryption Scheme

The multi-frame decryption scheme is effective when users hold the camera in hand. However, many QR code scanners used in stores nowaday are fixed on a table [8]. Moreover, the multi-frame decryption scheme requires several frames to correctly reconstruct the original QR code. According to our evaluation in Sec. 7, it takes 10.2 frames in average to correctly decrypt an *mQR* code. Therefore, we sought to develop a fast decryption scheme for the scenario with a fixed scanner or requiring a shorter decryption time.

For the sake of illustration, we present the following simple experiment. We first use the proposed encryption scheme to encrypt a white image (as shown in Fig. 13(a)) and its Moiré pattern (Fig. 13(b)). We then encrypt another image with 2×2 black and white blocks, the Moiré pattern of which is shown in Fig. 13(d). The sizes of the two images are the same, and the display and camera are fixed in set positions. Clearly, the shape of the Moiré patterns from two images are similar; however, the colors located at black blocks in the second image are inverted. This suggests that when the camera and the display configuration are unchanged and the Moiré pattern of the encrypted white image is known, then we can predict the Moiré pattern of any *mQR* codes in a given position simply by inverting the green and purple located at black blocks.

This observation is used to guide the design of our fast mQRCode decryption scheme. We employ a QR-code scanner and placed a phone displaying an *mQR* code on the table to facilitate scanning. The phones alternatively display an encrypted white image and an *mQR* code at $10fps$. The camera is configured to use a fixed focal length while recording video at a frame rate of $30fps$. Once the scanner captures from the phone a frame with the encrypted white image and a frame with the *mQR* code, the two frames then undergo processing to reconstruct the original QR code:

Learning the phase inversion pattern: The camera uses the Moiré pattern of the encrypted white image as a reference by which to learn the changes in color mapping resulting from phase inversion (Fig. 15(a)).

Segmentation: For the frame with an *mQR* code, we use the width and height of three known locators to compute the size of each QR code block and segment the *mQR* code (Fig. 15(b)).

Cancelling phase inversion effects: We compare the colors of the captured *mQR* code with the reference image derived in the steps listed above. If the color difference of a given pixel in two images exceeded a given threshold (80, 120, 120 in the RGB channels, respectively), then the pixel is marked as a different color. If more than 80% of the pixels in a block are different, then the block is colored black; otherwise, the block is white (Fig. 15(c)).

The above fast decryption scheme requires only two frames to decrypt the generated *mQR* code, thereby greatly reducing decryption time.

7 EVALUATION

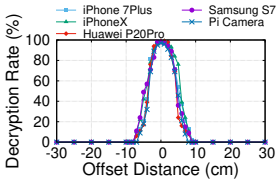
7.1 Experiment Methodology

We generate 20 version-3 (29×29) QR codes to encode random text messages with the error correction level set at "M" (i.e., 15% data restoration) [3] using mQRCode for encryption. The generated *mQR* codes are displayed on 10 displays (4 iOS, 4 Android, 1 desktop display, and 1 laptop display). Furthermore, the codes are configured specifically for receivers position at a specific distance at an angle of 0°. 8 smartphones (4 iOS and 4 Android) with built-in cameras and 2 PiCameras connected to a Raspberry Pi [22] are used to decrypt the *mQR* codes. The cameras are set to record a video of each *mQR* code at $30fps$ for $5sec$. Each experiment is repeated 30 times for each of the 20 *mQR* codes. For the multi-frame decryption method, the display is fixed and the camera is held by an user. For the fast decryption method, both the display and the camera are fixed. We report the averaged percentage of messages that are correctly extracted from the *mQR* codes.

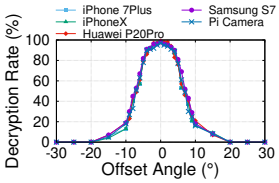
7.2 mQRCode Performance

7.2.1 Decryption Range. The primary objective behind the development of mQRCode is to enhance the security of QR codes. Fig. 16(e) and 16(f) show the decryption rates of the standard QR codes with 5 cameras positioned at various

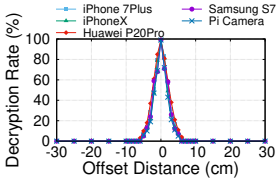
[3] Alipay uses version-2 (25×25) QR codes [48] while WeChat uses version-1 (21×21) [48] QR codes. Version-3 QR codes which carry more data are representative of the amount of data needed by these systems.



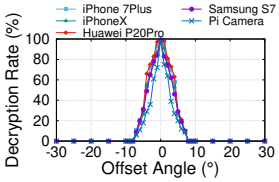
(a) Impact of distance on multi-frame decryption for *mQR* codes.



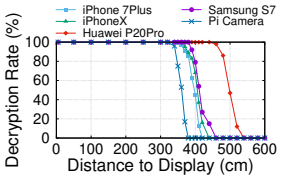
(b) Impact of angle on multi-frame decryption for *mQR* codes.



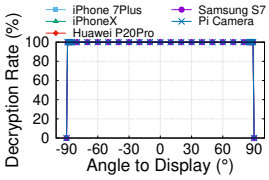
(c) Impact of distance on fast decryption for *mQR* codes.



(d) Impact of angle on fast decryption for *mQR* codes.



(e) Impact of distance on decoding QR codes in front of the display.



(f) Impact of angle on decoding QR codes one meter away from the display.

Figure 16: Decryption rate of *mQR*/QR codes.

angles and distances from the screen (DELL S2340M). We can see that all cameras can decode standard QR codes while being placed within 3 meters from the display. The view angle has little impact to the QR code decryption rate (100% from $-89°$ to $89°$.) These results imply that the standard QR codes are easily sniffed.

We then encrypt these standard QR codes via mQRCode. Figs. 16(a) and 16(c) present the decryption rates of *mQR* codes with 5 cameras positioned at the correct view angle but at various distances from the screen. Figs. 16(b) and 16(d) present the decryption rates with the camera positioned at the correct distance but at various view angles. When the camera is positioned at the designated distance (shifted by $0cm$) and at the designated angle (shifted by $0°$), the decryption rate is 100%. When the camera is $10cm$ or $20°$ away from the designated position, the decryption rate drops to 0. These results demonstrate the efficacy of mQRCode in preventing QR codes from being sniffed.

7.2.2 Frames Required for Decryption. Number of frames required for the decryption is an important factor in assessing

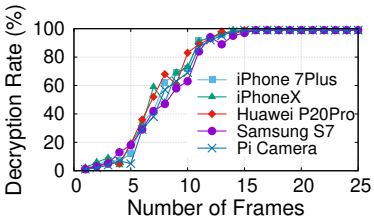
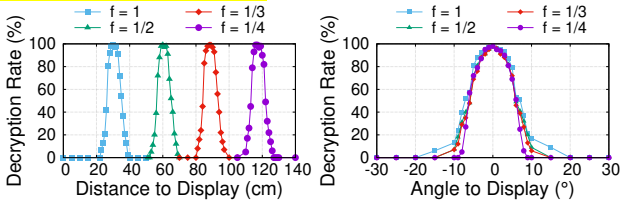


Figure 17: Decryption rate is shown to increase with the number of frames.



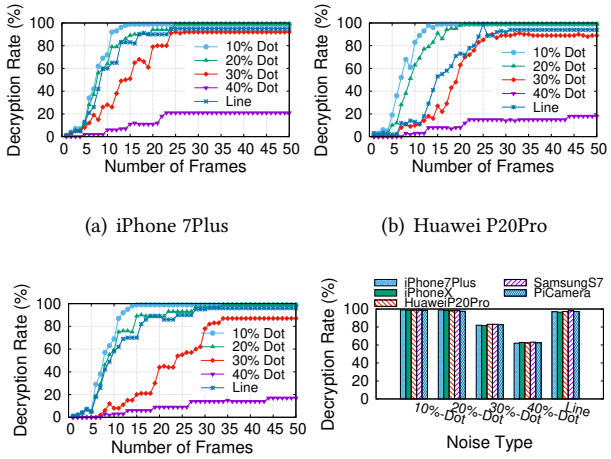
(a) Impact of distance. (b) Impact of angle.

Figure 18: Multi-frame decryption rate with different frequency modulations.

the usability of mQRCode. The fast decryption scheme requires only two frames for decryption; therefore, the following assessment deals only with the multi-frame decryption scheme. We use 5 cameras to record videos of 20 *mQR* code at $30fps$ for 5 seconds and repeat 30 times. We apply the multi-frame decryption method to all the videos and filter out the blurred frames. Fig. 17 shows CDF of the number of frames required to decrypt *mQR* codes when the camera is held in the correct position. We can see that the average number of frames required for multi-frame decryption is 10.2 and with 16 frames all *mQR* codes can be correctly decrypted.

7.2.3 Impact of Frequency Modulation. We evaluate the frequency modulation scheme. The 20 QR codes are modulated with four frequencies and displayed on DELL S2340M. We use iPhone 7Plus to decrypt the *mQR* codes at various positions and the decryption rates are shown in Fig. 18(a) and 18(b). First, we can see that when a smaller modulation frequency is used, the *mQR* codes can be decrypted at a longer distance. It provides the flexibility in designing *mQR* codes for applications targeted at various operating distances. Second, the *mQR* codes still offer the high security since they can only be decrypted at the designated distances and angles.

7.2.4 Impact of Phase Discontinuity. We evaluate the impact of added dot noise or camouflage lines to hide the boundaries resulting from abrupt phase changes. Figs. 19(a)-19(c) show the decryption rates and corresponding number of frames required for the multi-frame decryption scheme after adding 10%-40% dot noise or camouflaging lines. The decryption rate is above 95% when using 13 frames for 10% dot noise and 21 frames for 20% dot noise. When 30% dot noise is added,



(a) iPhone 7Plus (b) Huawei P20Pro

(c) Pi Camera (d) Fast Decryption

Figure 19: Decryption rates of mQRCode with 10%-40% added noise or camouflaging lines: (a)-(c) results obtained using multi-frame decryption scheme; (d) results obtained using fast decryption scheme.

the decryption rates are as follows: iPhone 7Plus (91.2%), Huawei P20Pro (89.5%), and PiCamera (89.7%). When 40% dot noise is added, decryption rates drop to less than 20%, even when using 50 frames.

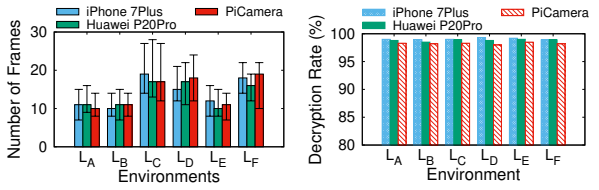
The decryption rates after adding camouflaging lines are as follows: iPhone 7Plus with 13 added frames (94.7%), Huawei P20Pro with 15 added frames (93.3%), and PiCamera with 12 added frames (95.2%). Nonetheless, the decryption rate do not improve with the addition of more frames. Investigating the traces that fails decryption reveals that the increasing dot noise and camouflaging lines add the errors in block color classification, which decrease the decryption rate.

Fig. 19(d) shows that the average decryption rate when using the fast decryption scheme following the addition of dot noise or lines. When only 10% or 20% dot noise is added, the decryption rate is above 97%. When 30% dot noise is added, the decryption rate drops to 80%. When 40% added, the decryption rate drops to 60%.

From the results of these experiments, 10% added noise is sufficient to prevent the camera from capturing boundaries at a distance of 10*cm* from the *mQR* codes, which should be sufficient for most practical applications.

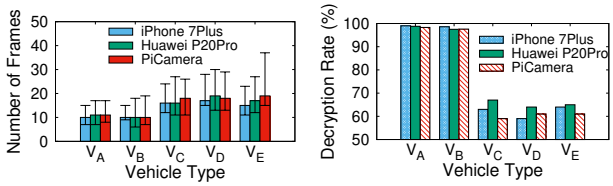
7.2.5 Impact of Environmental Factors. QR codes must be effective and robust under a variety of environments. Lighting is a major factor affecting the QR code decoding rate. It is also an important issue in mQRCode decryption.

We sought to evaluate the impact of lighting by performing experiments under the following conditions: outdoors (8 AM, noon, and 11 PM), under typical office lighting, and indoors with all of the lights turned off. Fig. 20(a) shows



(a) Multi-frame decryption. (b) Fast decryption.

Figure 20: Impact of lighting conditions on multi-frame decryption scheme and fast decryption scheme (error bar shows standard deviation): L_A: Outdoor at 8A.M.; L_B: Outdoor at 12A.M.; L_C: Outdoor at 11P.M.; L_D: Outdoor in a cloudy day; L_E: Office; L_F: Indoor with all lights off.

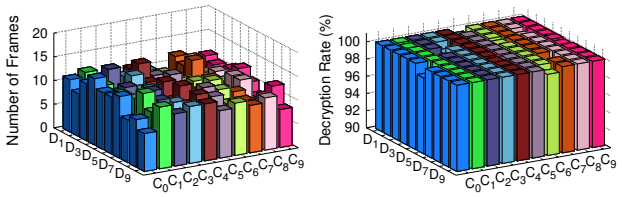


(a) Multi-frame decryption. (b) Fast decryption.

Figure 21: Influence of vehicles on decryption rates (error bar shows standard deviation): V_A: stationary; V_B:high-speed rail; V_C: subway; V_D: bus; V_E: taxi.

the average number of frames required to achieve 95% decryption rate using the multi-frame decryption scheme. A higher number of frames are required in a dark environment (*outside at 11 PM* and *indoors with all of the lights off*) due to the fact that the camera has to employ a higher ISO, which increases the amount of noise and in so doing makes decryption more difficult. Fig. 20(b) shows the average decryption rate using the fast decryption method. Under this scheme, the decryption rate also drops somewhat in a dark environment. This is a clear indication that sufficient light is crucial to the normal operation of mQRCode.

Another factor that can impact the decryption rate is vibration caused by the surrounding environment, as when users are riding on trains, subways, buses, or taxis. Fig. 21(a) shows the average number of frames required for the multi-frame decryption scheme under these scenarios. We can see that the number of frames required on the high-speed rail and subway is similar to the number of when the user is stationary due to the relative stability. However, while riding on a bus or taxi, 1.7 times as many frames are required than in the stationary scenario. The vibration introduced by these vehicles caused the camera or smartphone to deviate from the designated position, and the cameras require a certain amount of time to adjust in response to changes in distance. Note that the Moiré pattern can disappear or



(a) Multi-frame decryption. (b) Fast decryption.

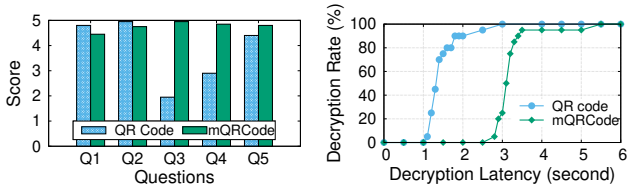
Figure 22: Impact of various cameras and displays. Displays are D_0: iPhone 6; D_1: iPhone 7Plus; D_2: iPhone X; D_3: iPhone XS; D_4: Huawei P20Pro; D_5: Samsung S7; D_6: Nexus 6P; D_7: Google Pixel 2; D_8: DELL S2340M; D_9: MacBookPro 2016; Cameras are C_0: iPhone 6; C_1: iPhone 7Plus; C_2: iPhone X; C_3: iPhone XS; C_4: Huawei P20Pro; C_5: Samsung S7; C_6: Nexus 6P; C_7: Google Pixel 2; C_8: Pi Camera (5MP); P_9: Pi Camera (8MP).

fade if the focus is not set correctly. Fig. 21(b) shows the average decryption rate under the fast decryption scheme in the above-mentioned scenarios. Vibration also affects the performance of the fast decryption scheme because it relies on only two frames, such that any shaking has a profound effect and causes the decryption to fail.

7.2.6 Impact of Displays and Cameras. We also examine how mQRCode works on a variety of mobile devices. Ten devices are used to display *mQR* codes and ten mobile cameras are used to capture videos to decrypt the *mQR* codes. Fig. 22(a) shows the average number of frames required for multi-frame decryption. All of the display-camera pairs work in a similar manner, wherein an average of 11.3 frames is required for decryption. The average decryption rate under the fast decryption scheme is presented in Fig. 22(b). In these tests, mQRCode is proved to be highly robust, with an average decryption rate of 98.6%.

8 USER STUDY

To evaluate the user experience of mQRCode, a within-subject user study [1] involving 20 participants (ten men and ten women) aged 15 to 60 years (average age of 34.51 years) from a variety of occupational backgrounds (e.g. students, lecturers, public servants, shopkeepers, etc.) is conducted. Each participant uses the mobile phones pre-installed with our scanning APP. We customize our APP by implementing both QR and *mQR* code scanning functions. To guide the users in properly putting the codes within the desired area, we overlay a "hint box" (similar to the QRCodeFinder box [60]) on the user interface of our APP. The evaluation procedure consists of three phases: the pre-study orientation, the experiment, and the post-study questionnaire. First, a pre-study orientation (5 min) is conducted to explain the



(a) User Rating. (b) Decryption latency.

Figure 23: User study results. (a) The average rating towards five questions. (b) The CDF of decryption latency.

goal of this study and advise them on operations of scanning both QR and *mQR* codes with our APP. In the experimental phase (30 min), we give each participant a mobile phone and ask them to use our APP to perform 50 scanning tasks, i.e., decoding 50 QR codes (25 are the actual Alipay payment QR codes and 25 are WeChat QR codes) and their corresponding *mQR* codes. In other words, each scanning task includes two subtasks: scanning a standard QR Code (QR-code subtask) and the corresponding *mQR* code (*mQR*-code subtask). The displaying orders of codes are counterbalanced, i.e., with half of the tasks beginning with scanning a standard QR code and the other half with scanning an *mQR* code. The *decryption latency* for scanning a code can be measured from the starting time when a user launches the APP till the ending time that the user successfully decrypts a generated code with the APP. Finally, a post-study questionnaire (Appendix B) gathers the user feedback, which is ranked in a 5-point Likert scale (1 ("Strongly Disagree") ~ 5 ("Strongly Agree")), after completing all the code scanning tasks. Paired t-tests are used to check for differences in the opinions towards scanning QR and *mQR* codes.

Fig. 23(a) summarizes participant responses towards five questions (in Appendix B.2) in the post-study questionnaire. The five questions assess if participants can properly position the phones. (*Q1*), if they feel the decryption is slow (*Q2*), if they feel the system is secure enough (*Q3*), the satisfaction (*Q4*), and the willingness (*Q5*).

Participants agree that it is simple to put the *mQR* codes in the hint box where scanning is easy according to the average rating (4.45) towards *Q1*, which shows no significant difference from that collected in the QR-code subtasks (4.8) (p = 0.11, p means p-value [57]). Fig. 23(b) shows the decryption latency of mQRCode and the standard QR code. Over 90% of the participants can successfully scan and decrypt the standard QR codes within 1.8 seconds while over 90% of the participants can successfully scan and decrypt *mQR* codes within 3.4 seconds. This increase is contributed by the time that participants has to slightly adjust the shooting distance and angle along the edge of the hint box to find the correct

position for decryption. However, participants expresses that they still felt mQRCode can decrypt in an acceptable speed according to the slightly lower average rating (4.7) towards *Q2*, which shows no significant difference from that (4.9) in QR-code subtasks ($p = 0.16$).

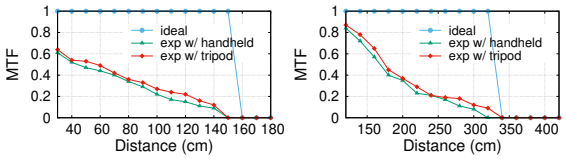
Moreover, participants agree that mQRCode can securely prevent someone nearby taking a sneaky photo of the *mQR* code according to the average rating (4.95) towards *Q3*, which is significantly greater than that (1.95) collected in the QR-code subtasks ($p < 0.05$). Overall, the acceptable decryption latency (*Q2*) and great security enhancement (*Q3*) make participants significantly more satisfied with and more willing to use mQRCode than the standard QR code system, which is reflected by that the average ratings (*Q4*: 4.85 and *Q5*: 4.8) collected in the *mQR*-code subtasks are significantly higher than those (*Q4*: 2.9 and *Q5*: 4.4) collected in the QR-code subtasks ($p < 0.05$ for both *Q4* and *Q5*).

9 LIMITATION AND DISCUSSION

In this section, we discuss the limitation of mQRCode. The motivation of the mQRCode is to enhance the security of QR codes while the goal of the attacker is to obtain the QR codes generated by the victim in order to make illegal transactions or steal private information. There are two ways for the attackers to decrypt *mQR* codes. First, we assume that the attacker knows the designated position from which to take pictures to decrypt *mQR* codes. This is technically possible based on the assumption that the attacker knows the decryption schemes; however, the designated position is very close to the victim (i.e., $< 50cm$), which is unlikely to go unnoticed. In fact, we would argue that this type of attack method is infeasible.

Second, the attacker could attempt to clearly capture every pixel of an *mQR* code and then decrypt it using the knowledge introduced in the paper. Specifically, information pertaining to the Bayer CFA is in the public domain, which means that $p_{cfa}(x, y)$ and $\phi_{cfa}(x, y)$ are widely known. Remember that p_{dec} can be computed with $p_{cfa}(x, y)$ and $p_{enc}(x, y)$ using Eq. 4. The attacker could construct $\phi_{enc}(x, y)$, by creating a matrix matching the size of the captured image. For pixels that are black, the corresponding elements in the matrix would be set to 0s and all other elements would be set to 1s. According to the following equation: $m_{dec}(x, y) = p_{dec}(\phi_{cfa}(x, y) - \phi_{enc}(x, y))$, the attacker can then compute $m_{dec}(x, y)$ (i.e., the original QR code) since the other three terms are known.

This means that the second attack method is theoretically feasible as long as the attacker can capture every pixel of the *mQR* codes from the screen. But how easy would it be for the attacker to capture pixel-level information? According to the ideal pinhole imaging principle, the spatial resolution



(a) 18-105mm lens (b) 70-200mm lens

Figure 24: MTF of Nikon D7000 with two lenses.

is determined by the pixel pitch of the display and the camera as well as the camera focal length. However, in practice, spatial resolution is largely impaired by lens distortion and aliasing [39]. One well-known metric used to quantify spatial resolution is the Modulation Transfer Function (MTF). The modulation of an image represents its contrast in spatial domain, as follows: $M = \frac{S_{max} - S_{min}}{S_{max} + S_{min}}$, where S_{max} and S_{min} represent the maximal and minimal pixel values within an image. A higher M is indicative of higher contrast. MTF is used to define the modulation ratio between a captured image and the displayed image: $MTF = \frac{M_{cap}}{M_{disp}}$, where M_{cap} and M_{disp} respectively represent the modulation of the captured image and the displayed image. We use a displayed image with $M_{disp} = 1$, such that MTF is between 0 and 1. The MTF is proportional to the spatial resolution. We can set a cutoff threshold for MTF. When the MTF value is below the threshold, there is no way to differentiate among black and white pixels. Fig. 24 shows the MTF of the Nikon D7000 with an AF-S NIKKOR 18 $\sim$ 105mm lens and an AF-S NIKKOR 70 $\sim$ 200mm lens. When we set the cutoff threshold to 0.3 [58], the camera with 105mm focal length is unable to differentiate individual pixels when the camera is farther than 108cm. We can notice the distance is much smaller than 140cm computed using the ideal pinhole imaging principle. When the camera is held in hand (which is a more realistic case for the attacker), MTF is further decreased by 3.92%. Give the size of the camera (110mm long) and the maximal attack distance (108cm), it's likely that the attack is noticed.

When a telephoto lens with the 200mm focus is used, the maximal attack distance is increased to 210cm. However, the size of the lens is also increased to 88.5$mm \times$ 202.5mm (diameter and length) which makes it even harder to disguise. Although lenses with longer focal lengths (e.g., 800mm [34]) are available in the market, their sizes and prices make them hard to be used in the attacks. Overall, it would be reasonable to conclude that mQRCode greatly reduces the risks involved in leaking information via QR code.

10 RELATED WORK

QR codes have been implemented as an information-sharing medium over a wide spectrum of real-world applications. QR

codes can be conveniently scanned using cameras on smartphones; however, they suffer from limitations [6, 32, 38, 49] in terms of security. Researchers have devised a number of visual or optical cryptography schemes to hide information in camouflaged visual patterns. Next, we summarize related studies on the secure exchange of information with visual patterns. Previous works focused on the following issues: 1) designing QR codes to enable the efficient encoding/decoding of information, 2) visual and optical cryptographic solutions to enable the secure exchange of information, and 3) leveraging Moiré patterns to hide messages.

10.1 Design of QR-Codes

Traditional QR codes comprise a number of black and white blocks that represent specific pieces of information. Commercial enterprises [45] and researchers [16] have further customized QR codes by slightly altering the patterns to incorporate colors, logos, and other features as a form of personalization. ARTCode [59], halftone QR codes [13], and PiCode [29] embed information entirely in a human-readable content to offer a more pleasant and informative user experience. To exploit diverse hardware capabilities, Strata [28] proposed a layered coding scheme to support a range of capture resolutions and deliver information at corresponding rates. The technology of mQRCode is orthogonal to these works; i.e., it is fully applicable to all of these schemes. In fact, it is perfectly feasible to design mQRCode code with information embedded in a human-readable format.

10.2 Optical and Visual Cryptography

Most of the studies listed above aimed to improve the readability and encoding efficiency of QR codes; however, none of them address issues pertaining to the security of QR codes. Most existing QR code applications requiring secure communication, such as mobile payment [40] and authentication [12, 15, 36], encrypt messages directly in QR codes. However, recent studies have shown that encrypting messages does not mitigate the threat because attackers performing Replay and STLS attacks need only an image of QR code and do not need to decrypt the messages [6].

To enable the complete concealment of visual images, existing visual cryptography (VC) techniques [43] encode a secret image into shared images with camouflaged visual patterns such that stacking a sufficient number of shared images reveals the original secret image. In [10, 20, 27, 36, 51], VC technology has been applied to QR codes to check the identity of individuals accessing QR codes or to control permissions related to accessing protected data. However, those works require that users scan multiple images or exchange key images in advance in order to recover the original QR code. In contrast, the scheme proposed in this study requires only that users hold the camera in a designated position to immediately obtain embedded messages.

Images can also be concealed via optical encryption. Double Random Phase Encoding [53] and its numerical derivatives [2, 24, 37, 47, 50] seek to encrypt images using a series of optical lenses and optical infrastructure. In [9, 11, 31], these technologies were used to hide encrypted QR codes. Nonetheless, optical encryption methods require specialized optical hardware, whereas mQR code relies only on common smartphones to decrypt hidden QR codes.

10.3 Leveraging Moiré Patterns

Moiré patterns have been used in a variety of research projects [18, 25, 33] to hide images. Lebanon et al. [33] explored the superimposition of grating patterns to create Moiré patterns of facial images to be visualized by humans. Hersch et al. [25] created moving Moiré components running up and down at different speeds and in different orientations when applying translation to the revealing layer. Desmedt et al. [19] created a scheme by which to secretly share information in realistic images. Tsai et al. [52] enabled the creation of Moiré art and allowed visual decoding by superimposing grating images printed on separate transparencies. These approaches require two semi-transparent layers to overlap each other to reveal the hidden image. Unlike those works, mQRCode and [42] exploit the nonlinear optical interaction between a camera (specifically the CFA) and a camouflaging pattern to hide QR codes. Moreover, the simple black and white blocks in QR codes make it difficult to apply existing methods in a manner that is not visually obvious.

11 CONCLUSION

We present mQRCode, a system that achieves secure and robust QR code communication. mQRCode provides a number of benefits over existing systems. First, mQRCode is a software-based solution, which requires no additional hardware or communication channels. Second, encryption and decryption rely on the relative position between the *mQR* code and the camera. Thus attackers are prevented from decrypting the *mQR* codes due to the fact that they cannot occupy the same physical space as the would-be victim. Third, the computational overhead of decryption is low. Extensive experimental assessments demonstrated the efficacy of mQRCode in a range of environmental situations.

ACKNOWLEDGMENTS

We are grateful to our shepherd and anonymous reviewers for their constructive feedback. We appreciate all 20 participants in our study study. This work is supported by NSFC grant U1736207 and 61572324, and Startup Fund for Youngman Research at SJTU.

REFERENCES

[1] 2010. Encyclopedia of Research Design AU - Salkind, Neil. (2010). https://doi.org/10.4135/9781412961288

[2] Ayman Alfalou and C Brosseau. 2009. Optical image compression and encryption methods. *Advances in Optics and Photonics* 1, 3 (2009), 589–636.

[3] Alipay. 2019. Alipay: Experience fast, easy and safe online payments. https://intl.alipay.com/

[4] AliPay. 2019. Alipay Service (Merchant scans with Integrator and Acquirer). https://global.alipay.com/service/barcode/7

[5] Isaac Amidror. 2009. *The Theory of the Moiré Phenomenon: Volume I: Periodic Layers*. Vol. 38. Springer Science & Business Media.

[6] Xiaolong Bai, Zhe Zhou, XiaoFeng Wang, Zhou Li, Xianghang Mi, Nan Zhang, Tongxin Li, Shi-Min Hu, and Kehuan Zhang. 2017. Picking up my tab: Understanding and mitigating synchronized token lifting and spending in mobile payment. In *26th USENIX Security Symposium (USENIX Security 17)*. 593–608.

[7] Matthias Baldauf, Markus Salo, Stefan Suette, and Peter Fröhlich. 2013. Display pointing: a qualitative study on a recent screen pairing technique for smartphones. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*. British Computer Society, 45.

[8] Barcodes. 2019. QR Code Barcode Scanner. https://www.barcodesinc.com/cats/barcode-scanners/qr.htm

[9] John Fredy Barrera, Alejandro Mira, and Roberto Torroba. 2013. Optical encryption and QR codes: secure and noise-free information retrieval. *Optics express* 21, 5 (2013), 5373–5378.

[10] Xiaohe Cao, Liuping Feng, Peng Cao, and Jianhua Hu. 2016. Secure QR code scheme based on visual cryptography. In *2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016)*. Atlantis Press.

[11] PA Cheremkhin, VV Krasnov, VG Rodin, and RS Starikov. 2017. QR code optical encryption using spatially incoherent illumination. *Laser Physics Letters* 14, 2 (2017), 026202.

[12] Yang-Wai Chow, Willy Susilo, Guomin Yang, Man Ho Au, and Cong Wang. 2016. Authentication and transaction verification using QR codes with a mobile device. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 437–451.

[13] Hung-Kuo Chu, Chia-Sheng Chang, Ruen-Rone Lee, and Niloy J Mitra. 2013. Halftone QR codes. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 217.

[14] Kevin J. Connolly. 1998. *The psychobiology of the hand*. Cambridge University Press.

[15] Gibson Research Corporation. 2019. Secure Quick Reliable Login. https://www.grc.com/sqrl/sqrl.htm.

[16] Russ Cox. 2012. QArt Codes. https://research.swtch.com/qr/draw

[17] Adrian Davies and Phil Fennessy. 2012. *Digital imaging for photographers*. Focal Press.

[18] Yvo Desmedt and Tri Van Le. 2000. Moiré cryptography. In *ACM Conference on Computer and Communications Security*. Citeseer, 116–124.

[19] Yvo Desmedt and Tri Van Le. 2000. Moiré cryptography. In *ACM Conference on Computer and Communications Security*. Citeseer, 116–124.

[20] Wen-Pinn Fang. 2011. Offline QR code authorization based on visual cryptography. In *2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 89–92.

[21] Sina Farsiu, Michael Elad, and Peyman Milanfar. 2006. *Multi-frame demosaicing and super-resolution of color images*. Technical Report. California Univ Santa Cruz Electrical Engineering Dept.

[22] Raspberry Pi Foundation. 2019. Raspberry Pi. https://www.raspberrypi.org/

[23] Matthias Geel, Daniel Huguenin, and Moira C Norrie. 2013. PresiShare: opportunistic sharing and presentation of content using public displays and QR codes. In *Proceedings of the 2nd ACM International Symposium on Pervasive Displays*. ACM, 103–108.

[24] JongWook Han, Choon-Sik Park, Dae-Hyun Ryu, and Eun-Soo Kim. 1999. Optical image encryption based on XOR operations. *Optical Engineering* 38, 1 (1999), 47–55.

[25] Roger David Hersch and Sylvain Chosson. 2004. Band moiré images. In *ACM Transactions on Graphics (TOG)*, Vol. 23. ACM, 239–247.

[26] Keigo Hirakawa and Patrick J Wolfe. 2008. Spatio-spectral color filter array design for optimal image recovery. *IEEE Transactions on Image Processing* 17, 10 (2008), 1876–1890.

[27] Gwoboa Horng, Tzungher Chen, and Du-Shiau Tsai. 2006. Cheating in visual cryptography. *Designs, Codes and Cryptography* 38, 2 (2006), 219–236.

[28] Wenjun Hu, Jingshu Mao, Zihui Huang, Yiqing Xue, Junfeng She, Kaigui Bian, and Guobin Shen. 2014. Strata: layered coding for scalable visual communication. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 79–90.

[29] Wenjian Huang and Wai Ho Mow. 2013. PiCode: 2D barcode with embedded picture and ViCode: 3D barcode with embedded video. In *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 139–142.

[30] Yizhen Huang and Yangjing Long. 2008. Demosaicking recognition with applications in digital photo authentication based on a quadratic pixel correlation model. *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

[31] Shuming Jiao, Wenbin Zou, and Xia Li. 2017. QR code based noise-free optical encryption and decryption of a gray scale image. *Optics Communications* 387 (2017), 235–240.

[32] Harshith Keni, Montana Earle, and Manki Min. 2017. Product authentication using hash chains and printed qr codes. In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 319–324.

[33] Guy Lebanon and Alfred M Bruckstein. 2001. Variational approach to moiré pattern synthesis. *JOSA A* 18, 6 (2001), 1371–1382.

[34] Lensora. 2019. Lenses with the longest focal length. http://www.lensora.com/list_lenses.asp?sel=zoom_max

[35] Tao Li. 2019. QR code scams rise in China, putting e-payment security in spotlight. https://www.jianshu.com/p/b9657161933a

[36] Jianfeng Lu, Zaorang Yang, Lina Li, Wenqiang Yuan, Li Li, and Chin-Chen Chang. 2017. Multiple schemes for mobile payment authentication using QR code and visual cryptography. *Mobile Information Systems* 2017 (2017).

[37] M Madjarova, M Kakuta, M Yamaguchi, and N Ohyama. 1997. Optical implementation of the stream cipher based on the irreversible cellular automata algorithm. *Optics letters* 22, 21 (1997), 1624–1626.

[38] Vasileios Mavroeidis and Mathew Nicho. 2017. Quick Response Code Secure: A Cryptographically Secure Anti-Phishing Tool for QR Code Attacks. In *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*. Springer, 313–324.

[39] Don P Mitchell and Arun N Netravali. 1988. Reconstruction filters in computer-graphics. In *ACM Siggraph Computer Graphics*, Vol. 22. ACM, 221–228.

[40] Sana Nseir, Nael Hirzallah, and Musbah Aqel. 2013. A secure mobile payment system using QR code. In *2013 5th International Conference on Computer Science and Information Technology*. IEEE, 111–114.

[41] Bridget O'Donnell. 2019. Steals money sneakily by scanning people's QR code. https://www.thatsmags.com/shanghai/post/27482/man-steals-money-sneakily-scanning-people-s-qr-codes-in-shanghai

[42] Hao Pan, Yi-Chao Chen, Guangtao Xue, Chuang-Wen Bing You, and Xiaoyu Ji. 2018. Secure QR Code Scheme Using Nonlinearity of Spatial Frequency. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, 207–210.
[43] P Punithavathi and S Geetha. 2017. Visual cryptography: A brief survey. *Information Security Journal: A Global Perspective* 26, 6 (2017), 305–317.
[44] Abigail Raney. 2017. Pinhole Camera Theory Summary. https://ourpastimes.com/pinhole-camera-theory-summary-12210465.html
[45] Roger. 2009. Marc Jacobs QR Code. https://2d-code.co.uk/marc-jacobs-qr-code/
[46] Pavitra Shankdhar. 2015. Security Attacks via Malicious QR Codes. https://resources.infosecinstitute.com/security-attacks-via-malicious-qr-codes/
[47] Guohai Situ and Jingjuan Zhang. 2005. Multiple-image encryption by wavelength multiplexing. *Optics letters* 30, 11 (2005), 1306–1308.
[48] Jeroen Steeman. 2019. QR code Data Capacity. https://blog.qr4.nl/page/QR-Code-Data-Capacity.aspx
[49] Siwon Sung, Joonghwan Lee, Jinmok Kim, Jongho Mun, and Dongho Won. 2015. Security Analysis of Mobile Authentication Using QR-Codes. 5 (2015).
[50] Enrique Tajahuerce, Osamu Matoba, Steven C Verrall, and Bahram Javidi. 2000. Optoelectronic information encryption with phase-shifting interferometry. *Applied Optics* 39, 14 (2000), 2313–2320.
[51] S Thamer and B Ameen. 2016. A new method for ciphering a message using QR code. *Comput. Sci. Eng.* 6, 2 (2016), 19–24.
[52] Pei-Hen Tsai and Yung-Yu Chuang. 2013. Target-driven moire pattern synthesis by phase modulation. In *Proceedings of the IEEE International Conference on Computer Vision*. 1912–1919.
[53] G Unnikrishnan, J Joseph, and Kehar Singh. 2000. Optical encryption by double-random phase encoding in the fractional Fourier domain. *Optics letters* 25, 12 (2000), 887–889.
[54] Timothy Vidas, Emmanuel Owusu, Shuai Wang, Cheng Zeng, Lorrie Faith Cranor, and Nicolas Christin. 2013. QRishing: The susceptibility of smartphone users to QR code phishing attacks. In *International Conference on Financial Cryptography and Data Security*. Springer, 52–69.
[55] Eric W Weisstein. 2003. Convolution. (2003).
[56] Wikipedia. 2019. Wikipedia, Color Filter Array. https://en.wikipedia.org/wiki/Bayer_filter/
[57] Wikipedia. 2019. Wikipedia, p-value. https://en.wikipedia.org/wiki/P-value
[58] Charles Sumner Williams and Orville A Becklund. 1989. *Introduction to the optical transfer function*. Wiley New York etc.
[59] Zhe Yang, Yuting Bao, Chuhao Luo, Xingya Zhao, Siyu Zhu, Chunyi Peng, Yunxin Liu, and Xinbing Wang. 2016. ARTcode: preserve art and code in any image. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 904–915.
[60] Pengtao Zhang. 2019. How to design a good interface of scanning QR codes? https://www.jianshu.com/p/b9657161933a

A RATIONALE BEHIND PHASE MODULATION

In this section, we derive the rationale behind the phase modulation results using the green, red, and blue filters shown in Fig. 7.

Step 1: Modeling the green, red, and blue filters in CFA.

As described in Sec. 5.1, the green filter can be modeled as follows:

$$m^g_{cfa}(x,y) = p^g_{cfa}(\phi^g_{cfa}(x,y))$$
$$p^g_{cfa}(u) = 0.5 + 0.5cos(2\pi u)$$
$$\phi^g_{cfa}(x,y) = ((x+y) \mod 2)/2$$

where $m^g_{cfa}(x,y)$ represents the color reception of the green filter at coordinate (x,y) on the image sensor, $p^g_{cfa}(u)$ represents its periodic function, and $\phi^g_{cfa}(x,y)$ represents its phase function.

Next, we model the red filter. We let $m^r_{cfa}(x,y)$ represent the color reception of the red filter at coordinate (x,y) on the image sensor, $p^r_{cfa}(u)$ represent the periodic function. Here, we let $p^r_{cfa}(u)$ equals to $m^g_{cfa}(x,y)$ so the phase function of the red filter can be modeled as follows:

$$\phi^r_{cfa}(x,y) = \begin{cases} 0.5, & \text{if x is odd;} \\ ((x+y) \mod 2)/2, & \text{if x is even.} \end{cases}$$

We observed that the blue filter array has the similar distribution with the red one. We let the period function $p_{cfa-b}(u)$ in the $m_{cfa-b}(x,y)$ equal to $p^g_{cfa}(x,y)$ and $p_{cfa-r}(x,y)$, the corresponding phased function can be modeled as follows:

$$\phi_{cfa-b}(x,y) = \begin{cases} ((x+y+1) \mod 2)/2, & \text{if x is odd;} \\ 0.5, & \text{if x is even.} \end{cases}$$

Step 2: Phase Modulation

According to the phase modulation method introduced in Sec. 5.2, we first obtain the phase information on the encrypted image when using the green filter.

$$\begin{aligned} \phi_{enc}(x,y) =& \phi^g_{cfa}(x,y) - p^{-1}_{dec}(m_{dec}(x,y)) + 2k\pi, k \in \mathbb{Z} \\ =& ((x+y) \mod 2)/2 - \\ & p^{-1}_{dec}(m_{dec}(x,y)) + 2k\pi, k \in \mathbb{Z} \end{aligned}$$

By putting $\phi_{enc}(x,y)$ into the equation: $m_{enc}(x,y) = p_{enc}(\phi_{enc}(x,y))$, where the $p_{enc}(u) = 0.5 + 0.5cos(2\pi u)$, we can obtain the encrypted image:

$$m_{enc}(x,y) = \begin{cases} 0.5 + 0.5cos(2\pi(((x+y) \mod 2)/2 - 0.5)), \\ \qquad \text{if } m_{dec}(x,y) = 0 \\ 0.5 + 0.5cos(2\pi((x+y) \mod 2)/2), \\ \qquad \text{if } m_{dec}(x,y) = 1 \end{cases}$$

When $m_{dec}(x,y) = 0$, which means that the color at coordinate (x,y) of the original QR Code is black, the corresponding phase information in the encrypted image includes the equal amount of black and white pixels with the same distribution; when $m_{dec}(x,y) = 1$, which means that the color at

coordinate (x, y) of the original QR code is white, the corresponding phase information in the encrypted image also has the same amount of black and white pixels (with an opposite arrangement from those in black QR code blocks.)

Summarizing the above, we find that the encrypted image using green filter share the same color information so it is difficult for the human eyes to distinguish the difference between them. The example image is shown in Fig. 7(b).

Next, we calculate the result of an encrypted image using the red filter.

$$\phi_{enc}(x,y) = \phi^{g}_{cfa}(x,y) - p^{-1}_{dec}(m_{dec}(x,y)) + 2k\pi, k \in \mathbb{Z}$$
$$= \begin{cases} 0.5 - p^{-1}_{dec}(m_{dec}(x,y)) + 2k\pi, k \in \mathbb{Z}, \\ \qquad\qquad \text{if x is odd;} \\ ((x+y) \mod 2)/2 - \\ \qquad p^{-1}_{dec}(m_{dec}(x,y)) + 2k\pi, k \in \mathbb{Z}, \\ \qquad\qquad \text{if x is even.} \end{cases}$$

When $m_{dec}(x, y) = 1$, which means that the color at a coordinate (x, y) of the original QR code is white, the corresponding phase information in the encrypted image can be calculated as follows:

$$\phi_{enc}(x,y) = \begin{cases} 0.5 - 0 = 0.5, \\ \qquad\qquad \text{if x is odd;} \\ ((x+y) \mod 2)/2 - 0 = ((x+y) \mod 2)/2, \\ \qquad\qquad \text{if x is even.} \end{cases}$$

Similarly, when $m_{dec}(x, y) = 0$, which means that the color at a coordinate (x, y) of the original QR code is black. The corresponding phase information of encrypted image can be calculated as follows:

$$\phi_{enc}(x,y) = \begin{cases} 0.5 - 0.5 = 0, \\ \qquad\qquad \text{if x is odd;} \\ ((x+y) \mod 2)/2 - 0.5 \\ \quad = ((x+y) \mod 2 - 1)/2, \\ \qquad\qquad \text{if x is even.} \end{cases}$$

When $m_{dec}(x, y) = 1$, by putting $\phi_{enc}(x, y)$ above into the equation: $m_{enc}(x, y) = p_{enc}(\phi_{enc}(x, y))$, where $p_{enc}(u) = 0.5 + 0.5cos(2\pi u)$, we can obtain the encrypted image as follows:

$$m_{enc}(x,y) = \begin{cases} 0, & \text{if x is odd;} \\ 0.5 + 0.5cos(2\pi(((x+y) \mod 2)/2)), \\ \qquad\qquad \text{if x is even.} \end{cases}$$

From the above equations, we can see that in the black area of the original QR code, three-fourth of the same area in the encrypted image becomes white pixels so they look whiter to human eyes. Similarly, in the white area of the original QR code, three-fourth of the same area in the encrypted image becomes black pixels and appears to be blacker to human eyes.

Summarizing the above, we find that the encrypted image using the red filter still carries color information, which can explain the result in Fig. 7(c). Considering that the blue filter has the similar spatial distribution with the red filter, we can draw a conclusion that an encrypted image using the blue filter also carries color information as shown in Fig. 7(d).

B MQRCODE USER EXPERIENCE QUESTIONNAIRE

B.1 About you

In this section, you will be presented with questions about yourself.

1. **Your gender**:
   - ☐ Female
   - ☐ Male
2. **How old are you?**
   - ☐ Below 18
   - ☐ 18-24
   - ☐ 25-34
   - ☐ 35-44
   - ☐ 45-54
   - ☐ Above 55
3. **What is your occupation?**
   Please specify: _____

B.2 About the user experience

In this section, you will be presented with five questions about your opinions and attitudes towards the standard QR Code (i.e., unencrypted QR Code) and the MQRCODE (i.e., encrypted QR Code).

After completing tasks of scanning the standard QR Code, please rate your opinions toward standard QR Code with the following aspects.

4. **Q1: Simple to put the codes in the hint box where scanning is easy?**
   Disagree ☐—☐—☐—☐—☐ Agree
5. **Q2: Decryption rate is fast enough?**
   Disagree ☐—☐—☐—☐—☐ Agree
6. **Q3: The standard QR Code system is secure enough to prevent fraudulent charges?**
   Disagree ☐—☐—☐—☐—☐ Agree

7. **Q4: Are you satisfied with the standard QR Code system?**
   Very dissatisfied □—□—□—□—□ Very satisfied

8. **Q5: Are you willing to use the standard QR Code system?**
   Not willing □—□—□—□—□ Willing

After completing tasks of scanning the *mQR* codes, please rate your opinions toward mQRCode with the following aspects.

9. **Q1: Simple to put the codes in the hint box where scanning is easy?**
   Disagree □—□—□—□—□ Agree

10. **Q2: Decryption rate is fast enough?**
    Disagree □—□—□—□—□ Agree

11. **Q3: The mQRCode system is secure enough to prevent fraudulent charges?**
    Disagree □—□—□—□—□ Agree

12. **Q4: Are you satisfied with the mQRCode system?**
    Very dissatisfied □—□—□—□—□ Very satisfied

13. **Q5: Are you willing to use the mQRCode system?**
    Not willing □—□—□—□—□ Willing