Poster: Secure Visible Light Communication based on Nonlinearity of Spatial Frequency in Light

Hao Pan*
Lanqing Yang*
Shanghai Jiao Tong University
panh09,yanglanqing@sjtu.edu.cn

Yi-Chao Chen
Shanghai Jiao Tong University
yichao@utexas.edu

Guangtao Xue
Shanghai Jiao Tong University
xue-gt@cs.sjtu.edu.cn

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

Xiaoyu Ji
Zhejiang University
xji@zju.edu.cn

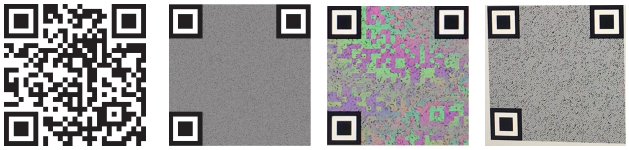
Pai-Yen Chen
University of Illinois at Chicago
pychen@uic.edu

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 *mQR* code), 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 *mQR* codes is $> 98\%$. The decryption rate for cameras positioned 20° off axis or $> 10cm$ from the designated location drops to 0%, indicating that any attackers will be unable to steal a usable image.

1 INTRODUCTION

Quick Response (QR) codes, are two-dimensional matrix barcodes and can be easily read using smartphones with a built-in camera. It is easy to generate QR-code images for authentication via mobile applications (APPs). For example, the Alipay [1] is a mobile payment system based on QR codes.



(a) Original QR code. (b) Encrypted QR code. (c) Picture taken at designated position. (d) Picture taken at wrong position (off by 15°).

Figure 1: *mQR* code can be naturally revealed when the camera is held in the designated position to the screen.

Unfortunately, QR code systems are susceptible to security risks in the form of Replay attacks [3] and Synchronized Token Lifting and Spending (STLS) attacks [2], in which an attacker secretly obtains an image of the victim's QR code to make other payment or access the private information.

To resist these security attacks, we develop a secure visible light communication scheme upon the standard QR code (hereafter our system is referred to as *mQRCode*). The proposed scheme exploits nonlinearities in the spatial frequency of light rays to encrypt and decrypt 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. 1(a)), *mQRCode* encrypts it within a pattern that is regarded as noisy (from the perspective of the human visual system) using a designated spatial frequency. An example of the resulting *mQR* code is shown in Fig. 1(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 in the designated position (in terms of distance and angle), nonlinearities in spatial frequency between the projected *mQR* code and the Color Filter Array (CFA) of the camera allow

*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.
https://doi.org/10.1145/3300061.3343391

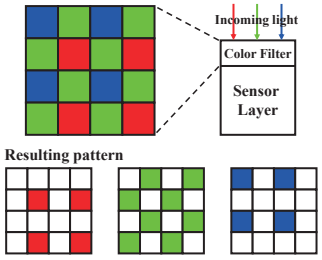


Figure 2: Profile of sensor with the Bayer arrangement of color filters.

the original QR code to be revealed as a Moiré pattern, as shown in Fig. 1(c). However, if an attacker captures an image of the *mQR* code from any other position, the scaling, translation, rotation of the *mQR* code will cause the result that the camera cannot make out the intended Moiré pattern. Thus, the physical limitation imposed by the position of the camera prevents any would-be attackers from decrypting the *mQR* code.

We have intensively evaluated a prototype of *mQRCode* to verify its effectiveness and robustness on a variety of displays, smartphones and PiCameras. Our experiments show that the decoding rate of any unauthorized camera at a distance of $> 10cm$ 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.

2 SYSTEM OVERVIEW

We now present a brief overview of the encryption and decryption in *mQRCode*.

2.1 Encryption

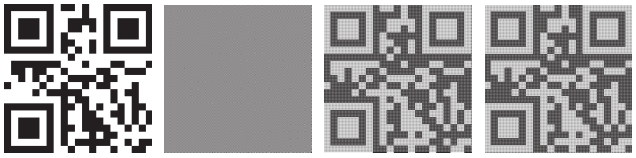
mQRCode exploits the nonlinear optical interaction between the Color Filter Array (CFA) in camera and the QR code pattern used to camouflage.

2.1.1 Spatial Frequency. We consider a bi-dimensional (2D) spatial structure with curvilinear pattern, which can be described using a frequency term and a phase term as $m(x, y) = p(\phi(x, y))$, where $m(x, y)$ represents the color magnitude at a coordinate (x, y), $p(\cdot)$ is a periodic function and $\phi(x, y)$ is a phase function.

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 $m_{dec}(x, y) = m_{cfa}(x, y) \cdot m_{enc}(x, y)$.

2.1.2 Color Filter Array Model. We first model the CFA by formulating $m_{cfa}(x, y) = p_{cfa}(\phi_{cfa}(x, y))$. In the Bayer filter in Fig. 2, 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 [5].

Here, the green filter was modeled as follows:



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

Figure 3: Encrypt a original QR code using three color filters in Bayer Color Filter Array.

$$\begin{aligned} m_{cfa}(x, y) &= p_{cfa}(\phi_{cfa}(x, y)) \\ p_{cfa}(u) &= 0.5 + 0.5cos(2\pi u) \\ \phi_{cfa}(x, y) &= ((x + y) \mod 2)/2 \end{aligned} \tag{1}$$

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.

2.1.3 Phase Modulation. To compute $m_{enc}(x, y) = p_{enc} \cdot (\phi_{enc}(x, y))$, we let $p_{enc}(u)$ equal $p_{cfa}(u)$ in order to enlarge the contrast of resulting Moiré pattern [4]. Phase modulation is applied by mapping black and white blocks in QR codes to different phases, combining the Eqs. 1 and [4], so we learn the following:

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

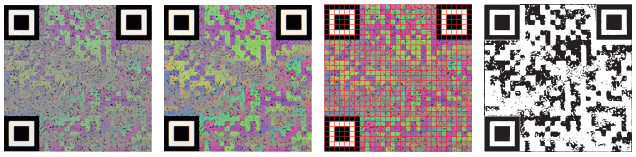
where p_{dec}^{-1} represents the inverse function of p_{dec}, which maps intensity values to the corresponding phases. The $2k\pi$ term has no impact on the encrypted image m_{enc} because *mQR* code uses *cosine* as the periodic function.

2.1.4 Handle 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. This problem can be mitigated by adding dot noise or camouflaging lines.

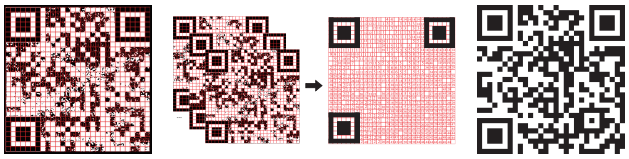
2.2 Decryption

Decrypting the *mQRCode* requires that the user holds the camera in a designated position, whereupon the Moiré effect reveals the original QR code, as shown in Fig. 4(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.

To address the above issues, we proposed the *mQRCode* decryption algorithm which utilizes multiple continuous video frames. The algorithm is based on the observation that when a user holds the camera to capture images of a *mQR* code, there is inevitably a certain amount of camera shake. The shifts induced between frames cause blurring



(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 4: Multi-frame decryption process.

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

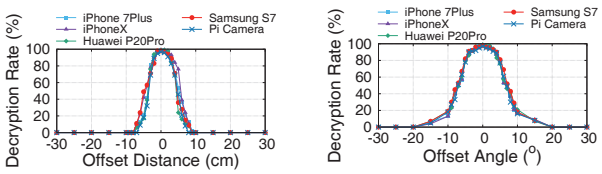
Enhancing color saturation: Fig. 4(a) presents a picture of an *mQR* code obtained using a digital camera. We first enhance the color saturation to enhance contrast among green, red, blue. The picture in Fig. 4(b) illustrates the results of saturation enhancement.

Segmentation: The size of QR code is determined by its version. *mQRCode* leave these locator marks untouched. We use three locators 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. 4(c).

Conversion to black and white: QR code blocks with different colors are assigned different phases, resulting in either green or purple separation in the Moiré pattern, as shown in Fig. 4(b). However, phase inversion alters the color mapping in spatial domain. To reliably identify blocks with the same phase, we separate green blocks by thresholding the green channel and converting the image to black and 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. 4(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 and is assigned a label in the new frame, then the block is assigned a new label. Otherwise, we search among existing frames for blocks with the existing label $index_{old}$ and blocks with label



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

Figure 5: Decryption rate of *mQR* codes.

$index_{new}$ in the new frame to be assigned 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. 4(f).

Coloring blocks: Each block is colored black or white in accordance with the labels. The rules for color blocks are as follows: If two adjacent blocks have the same label, they are drawn using the same color; Otherwise, they are drawn using different colors. The original QR code is then recovered after all blocks have been colored, as shown in Fig. 4(g).

3 EVALUATION

We have implemented *mQRCode* both as an iOS APP and an Android APP. We generate 10 version-3 (29×29) QR codes to encode random text messages with the error correction level set at "M" using *mQRCode* for encryption. Figs. 5(a) presents the decryption rates of *mQR* codes with 5 cameras positioned at the correct view angle but at various distances from the screen. Figs. 5(b) presents 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" (i.e., recorded by attackers' cameras).

REFERENCES

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

[2] 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.

[3] Siwon Sung, Joonghwan Lee, Jinmok Kim, Jongho Mun, and Dongho Won. 2015. Security Analysis of Mobile Authentication Using QR-Codes. *Computer Science & Information Technology - Computer Science Conference Proceedings* (2015).

[4] 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.

[5] Wikipedia. 2019. Wikipedia, Color Filter Array. https://en.wikipedia.org/wiki/Bayer_filter/