# SCREENID: Enhancing QRCode Security by Fingerprinting Screens

Yijie Li [†#], Yi-Chao Chen [†#],Xiaoyu Ji [‡] Hao Pan[†], Lanqing Yang[†], Guangtao Xue[†*], Jiadi Yu[†],
[†] Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[‡] College of Electrical Engineering, Zhejiang University, China
Email:{yijieli,yichao}@sjtu.edu.cn,xji@zju.edu.cn,{panh09,yanglanqing,gt_xue,jiadiyu}@sjtu.edu.cn

IEEE INFOCOM 2021 - IEEE Conference on Computer Communications | 978-1-6654-0325-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/INFOCOM42981.2021.9488859

***Abstract*—Quick response (QR) codes have been widely used in mobile applications due to its convenience and the pervasive built-in cameras on smartphones. Recently, however, attacks against QR codes have been reported that attackers can capture a QR code of the victim and replay it to achieve a fraudulent transaction or intercept private information, just before the original QR code is scanned. In this study, we enhance the security of a QR code by identifying its authenticity. We propose SCREENID, which embeds a QR code with information of the screen which displays it, thereby the QR code can reveal whether it is reproduced by an adversary or not. In SCREENID, PWM frequency of screens is exploited as the unique screen fingerprint. To improve the estimation accuracy of PWM frequency, SCREENID incorporates a model for the interaction between the camera and screen in the temporal and spatial domains. Extensive experiments demonstrate that SCREENID can differentiate screens of different models, types, and manufacturers, thus improve the security of QR codes.**

***Index Terms*—screen-camera communication; secure QR code;**

## I. INTRODUCTION

Quick response (QR) codes are barcodes comprising white and black blocks. In the past years, with the pervasive built-in cameras on smartphones, QR codes have been widely adopted in mobile applications such as communication, payment, etc. Especially, for mobile payment scenarios, the QR code system (hereafter we name a QR code system as QRCode) is almost a standard module for service providers such as AliPay, WeChat, PayPal, etc [1], [2]. Users just need to show their QR codes on smartphones for a quick transaction, which provides convenient and friendly experience.

The QRCode system works in a straightforward way. As shown in the flowchart in Fig. 1, payment transactions begin with the generation of a legal QR code, which includes the ID of the user in the application, e.g., the AliPay account, a time-stamp, and the secret transaction information in the form of an encrypted token. The user then presents the QR code to the cashier to have it scanned into the transaction system, together with other transaction information such as total amount and currency type. On the server side, the merchant extracts the token from QR code, obtains the user ID from the token and then accesses the database maintained by the company to retrieve the stored secret of the payer. This secret is then converted into a new token for verification. As long as the token is deemed valid, the transaction proceeds.

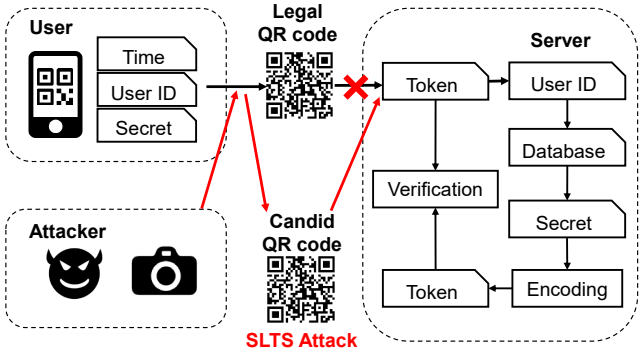



Fig. 1. A flowchart of a typical QRCode system for mobile transaction. Note that an adversary can capture the victim's QR code and then replay it for a fraudulent transaction.

However, the above-mentioned transaction process is far from secure. Recently, researchers have reported that a QR-Code system is susceptible to the Synchronized Token Lifting and Spending (STLS) attack and other similar attacks [3]. In this attack, the adversary first acquires an image of the QR code displayed on the victim's device [4], when the victim is showing the QR code to the cashier, for example. Then the adversary replays the stolen QR code for another transaction, which we call fraudulent transaction. To ensure the success of such an attack, the adversary should also finish the transaction before the legitimate scanning process from the cashier.

To against the STLS attack, researchers have proposed a lot of solutions. Most existing protection schemes aim at safeguarding the QRCode system by inserting codewords [5]–[7] or concealing the original QR codes using visually cryptographic techniques [8]–[13]. Unfortunately, attackers can still succeed in replaying the stolen QR code as they do not need to decrypt the message embedded in the QR code.

In this paper, we sought to enhance the security of the QRCode system by identifying the authenticity of a QR code. We propose SCREENID, which embeds a QR code with information of the screen that is displaying it, thereby the generated QR code can reveal whether the screen is corresponding to it. In SCREENID, we utilize the pulse width modulation (PWM) frequency of screens as the unique screen fingerprint because PWM is one of the most common approaches to control screen brightness for modern screens. From our experiments,

[#] Both authors contributed equally to the research.
[*] Corresponding author.

it can hardly find any two smartphones with the same PWM frequency. PWM frequency makes a good candidate for screen fingerprint as it possesses several properties. First, PWM frequencies are adjusted to different values by screen manufacturers for reasons such as cost control and power consumption. Second, even for the same manufacture, the PWM frequency shows variances due to the variations in the manufacturing processes. In Section IV, we show that PWM frequencies are different even for phones of the same model. Although the results cannot fully guarantee the security of QRCode system, the probability that any two mobile phones have the same frequency is negligible, therefore SCREENID can successfully enhance the security of QRCode system.

On the receiver side, e.g., the camera in a cashier, the PWM frequency is measured by the rolling shutter effect observed on the cashier's camera. The QR code is legal and the transaction proceeds only if the PWM frequency embedded in the QR code and the one measured from the screen are closely match.

In realizing the SCREENID, we overcome several challenges. First, the interaction between screen and camera involves many influencing factors, such as PWM dimming and screen refresh rate, etc. Obtaining an accurate estimation of PWM frequency actually requires that the interactions among these factors be modeled in the temporal as well as the spatial domains (Sec. VI). Second, it was also necessary to compensate for variations of distance and angle between the display device and the scanning device, e.g., a camera (Sec. VII-F). Third, we found that the differences in PWM frequency among smartphones were really small (e.g., $0.1Hz$), particularly when dealing with smartphones of the same model. Thus, we need to increase the frequency resolution, while minimizing the number of captured images (Sec. VII-G).

We verified the efficacy of a SCREENID prototype using $50$ screens under a variety of camera settings. We summarize the contribution of this paper as follows:

- We demonstrated the feasibility of using PWM frequency of screens for fingerprint, and the fingerprint can be embedded in to a QR code to check its authenticity. From our dataset, $99.3\%$ pairwise frequency differences of screens are larger than $0.1Hz$.
- We proposed SCREENID, which incurs no additional hardware for the QRCode system and has no requirements for user behavior. SCREENID can be implemented via a simple software update.
- We proposed to model the interaction between the camera and the screen in both temporal and spatial domains and achieved high estimation accuracy of PWM frequency, i.e., within $0.03Hz$.
- We conducted exhaustive experiments and demonstrated the efficacy of the SCREENID system. The evaluation shows the identifiable success rate (i.e. True Positive Rate) is above $94.3\%$ across the same model and the wrongly accepted rate (i.e. False Positive Rate) against attack attempts is below $0.8\%$.

## II. Related Work

### A. Visible Light Communication

Many works exploited the interaction between the light and camera. The visual microphone [14] used the rolling shutter effect to measure the vibration of objects to reconstruct the speech. Danakis et al. [15] exploited the rolling shutter effect to increase data rates in light-to-camera communications. LiTell [16], iLAMP [17], and Pulsar [18] have applied visible light communication to indoor localization based on frequencies specific to individual light sources. LiShield [19] uses the flickering of smart LEDs to protect visual privacy.

Our work also relies on the rolling shutter effect to capture the light frequency. Different from previous works which dealt with a single light source (e.g., a light bulb or a LCD screen), one of the challenges we faced is that many light sources (i.e., many pixels on OLED screen) flickering asynchronously at a given frequency. Obtaining accurate frequency estimation requires modeling interactions in temporal and spatial domains.

### B. Visual Cryptographic Security

Most previous studies have focused on the encryption of authentication codes [5]–[7], [20] directly within barcodes. Chen [7] embedded authentication data including codewords and signatures within QR code. Nonetheless, even those researchers concede that their systems are vulnerable to Replay and STLS attacks [3] without hardware fingerprint. To avoid STLS attacks, POSAUTH [3] requires double scanning which changes users' habits. Zhou et al. [21] utilize various brightness pixels as hardware fingerprint, but is restrict with completely dark environment and pixel aging problem.

Visual cryptography (VC) techniques [22] have also been developed to ensure that messages remain concealed. Essentially, a secret image is embedded within shared images aimed at camouflaging the hidden data, which can be revealed only through the stacking of multiple shared images or capturing with a specific approach [8]–[13]. mQRcode [12] utilizes moire patterns to encrypt the original QR code, thereby making the information impervious to extraction unless viewed from a specific position. Nonetheless, those methods necessitate the exchange of key images with a server or restrict the user to a pre-determined position to recover the original QR code.

Unlike the methods mentioned above, our approach utilizes hardware fingerprint to avoid STLS attacks, which does not require additional equipment and allows the user complete freedom in the use of their device.

## III. Background

### A. Pulse Width Modulation (PWM) Dimming Control

There are two mechanisms commonly used to control screen brightness. One is analog dimming [23], which adjusts screen brightness by regulating the voltage output from a DC power supply (i.e., DC dimming) (Fig. 2). The other is pulse width modulation (PWM) dimming [23] which adjusts screen brightness by digital encoding an analog signal to appear for a given period of time, wherein the power is either fully on or fully off.

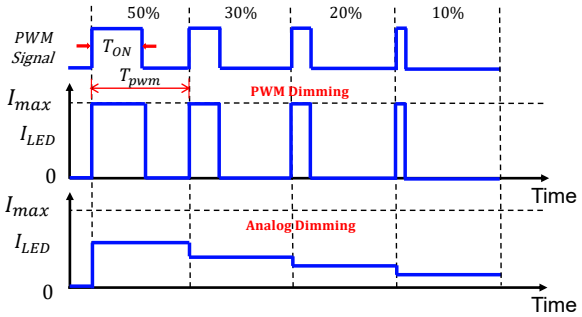



Fig. 2. An illustration of PWM dimming and analog dimming

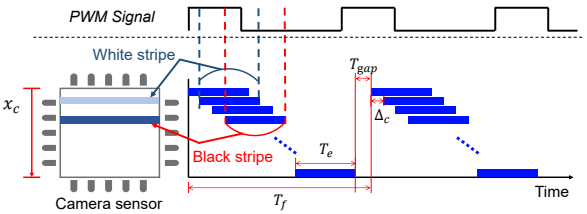
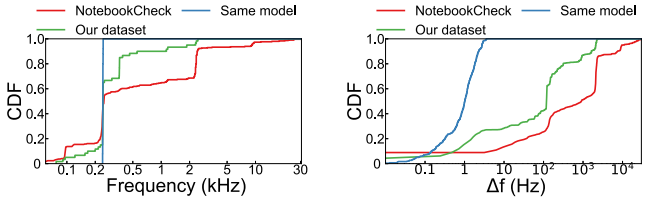
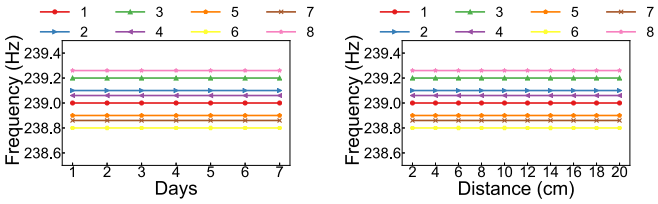



Fig. 3. Sampling performed by camera sensors. Some of the rows sample during the PWM on-time, and others sample during the off-time, respectively producing black (darker) and white (lighter) stripes in the captured image.




(a) PWM frequencies. (b) Pairwise PWM frequency differences.

Fig. 4. CDF of PWM frequencies and pairwise differences of 300 screens reported in NotebookCheck [26] and 50 screens (30 are of the same model) we collected.




(a) Time. (b) Distance.

Fig. 5. We measure the PWM frequencies of 8 screens of the same model across days and at various distances to show its stability.

Essentially, the screen brightness is adjusted by regulating the on-time ($T_{ON}$) in each cycle ($T_{pwm}$) [24]. Since human eyes are insensitive to changes of high frequency, we do not notice the flickering, but rather perceive a difference in brightness. Essentially, the screen appears brighter when the on:off ratio is larger.

PWM dimming has several advantages over analog dimming. PWM dimming does not impose chromatographic shifts (color cast due to lighting unit response sensitivity), as the current is always equal to the full-amplitude current $I_{max}$ or 0. PWM also permits high dimming resolution over a wide range of values. This has led to the widespread adoption of PWM dimming for OLED screens. Note that it is rapidly gaining popularity for mobile devices.

### B. Rolling Shutter Camera

Complementary metal-oxide semiconductor (CMOS) technology is widely used to fabricate the cameras used in modern smartphones. The sequential sampling by different rows in the camera sensor is referred to as a rolling shutter [25]. Fig. 3 illustrates the sampling process when capturing a video. The latency between the start of each exposure in each row is denoted by $\Delta_C$. Note that $\Delta_C$ remains a constant regardless of the ISO and exposure time and whether the sensor is used to capture a still image or video segment. The sampling rate $f_c$ of a rolling shutter camera is calculated as follows: $f_c = 1/\Delta_C$.

While capturing a video, the time during which one frame is captured is denoted by $T_f$. Thus, for a video at $30fps$, $T_f = 1/30$. Note that there is a time gap between the end of the last row in one frame and the start of the first row in the following frame. The time gap can be derived as follows: $T_{gap} = T_f - T_e - (X_c - 1)\Delta_C$, where $X_c$ indicates number of rows of camera sensor and $T_e$ represents the exposure time.

When using a rolling shutter sensor to capture an image from a screen with PWM-induced flicker, the light intensity indicates the total number of photons received by a given row throughout the duration of the exposure. Fig. 3 shows the black (darker) and white (lighter) stripes created by the sensor rows, while recording an image in which PWM signals periodically turn the screen on and off. Examples of this phenomenon are shown in Fig. 6. Details pertaining to the modeling of the stripes and their use in computing the PWM frequency are presented in Section VI.

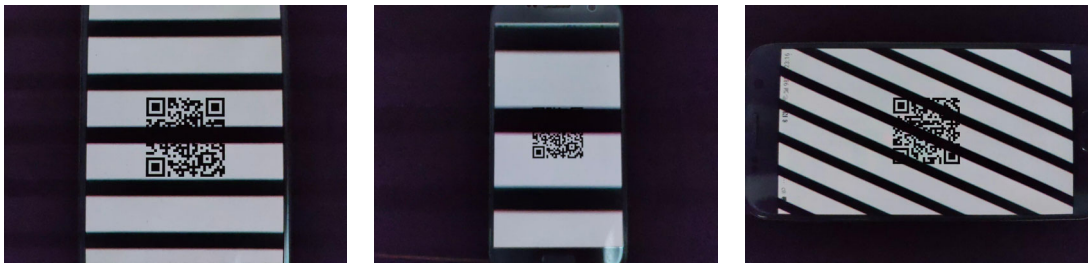
## IV. Preliminary Study

If PWM dimming frequency is to be used as a feature by which to identify screens, then we must first verify its uniqueness and stability.

### A. Uniqueness of PWM Frequency

We first conducted a preliminary study to assess the uniqueness of PWM frequencies across screens from different and the same models.We used a light sensor ADPD2212 [27] sampling at $80kHz$ to measure the PWM frequencies of 50 phone screens. We crawled 300 screen benchmarks collected by NotebookCheck [26] (covering mainstream smartphones from 2016 to 2019). Fig. 4(a) illustrates the distribution of PWM frequencies among smartphone screens. We can see that 97% of the PWM frequencies were below $10kHz$ and 68.3% were below $2kHz$. Note that most current smartphones support video capture at $1920 \times 1080$ using a frame rate of $30fps$. This means that it should be possible to achieve sampling rates of at least $32.4kHz$, which is far beyond the Nyquist sampling rate ($> 2 \times 10kHz$).

Fig. 4(b) shows the CDF of the pairwise differences in PWM frequency among screens. Note that the frequency resolution in the NotebookCheck dataset is $0.1Hz$ so it cannot distingush screens using a close PWM frequency. In our dataset, the frequency resolution is $0.01Hz$. We can see that among all screens and screens of the same model, 95% pairwise



(a) screen captured at $5cm$ (b) screen captured at $10cm$ (c) screen captured rotated by $90°$

Fig. 6. Images of screens captured using a camera. The position of the camera was fixed, while the position and direction of the screens was varied. The stripes varied in the screen in terms of width and angle.

differences are larger than $1.1Hz$ and $0.18Hz$, respectively. The results revealed that a frequency resolution of $0.1Hz$ should be sufficient to differentiate among $99.3\%$ screens.

### B. Stability of PWM Frequency

We also assessed the uniformity of PWM frequencies under various conditions. Fig. 5 shows the PWM frequencies of $8$ screens (Samsung S7 [28]) of the same model measured in various days and from various distances. The variation in PWM frequency was at most $0.01Hz$ which implies the PWM frequency is stable.

## V. Threat Model

In this paper, we aim to enhance the security of a QR code by embedding the screen fingerprint into the displayed QR code. An adversary may destroy the security by launching a STLS attack as following:

An adversary can intercept a QR code without the awareness of the victim during a transaction, for example. Then the stolen QR code can be replayed by the attacker to achieve a successful fraudulent transaction. The whole replay attack process should be finished before the token expires. The attackers may interrupt or delay the legitimate progress for attack by using some social-engineering methods,e.g., talking to the victim or the cashier [3], [29], [30].

We make the following assumptions on the adversary.

- **No access to the victim device**. We consider the adversary can physically obtain the QR code displayed on the victim device from a certain distance. For example, she may capture the QR code using a smartphone or a dedicated camera present at the payment scene.
- **Malware attack**. We assume that the adversary can infect victim's phone with a malicious app, which does not have system privileges but can screenshot or take camera permissions to obtain QR code. For instance, the reflection of QR code on the glass of POS scanner may be sniffed by the malicious app using front camera [3].
- **No limitation to software and hardware.** The adversary may utilize state-of-the-art image photographing, recording and synthesis software techniques. She can also utilize any devices such as high-quality digital cameras and high-speed networks for a replay attack.

## VI. Measuring PWM Frequency Using a Camera

In this section, we examine the process of measuring PWM frequency using rolling shutter effect. Note that our use of the rolling shutter for frequency measurement must contend with many light sources flickering asynchronously. We addressed this issue by modeling the screen-camera interaction in temporal as well as spatial domain.

### A. PWM Dimming of OLED screens

Under normal conditions, OLED screens use PWM dimming control. Figs. 6(a)(b)(c) show the black and white bands caused by the interaction between flicker and the rolling shutter. Note that the bands from an OLED screen varied in terms of width and angle, depending on the position of the screen. This can be attributed to the fact that each pixel in an OLED screen is controlled by an LED light source and the PWM cycle of each row is asynchronous. As shown in Fig. 7, the OLED pixels are grouped into rows, the flickering of which is delayed by latency $\Delta_S$ from the previous row. Note that PWM latency $\Delta_S$ is kept constant among rows in order to prevent fluctuations[1] in current [31].

### B. Using a Camera for Sampling

The fact that the brightness of OLED screens varies as a function of time and space means that the rolling shutter effect can be viewed as a process of sampling in the temporal and spatial domains. PWM dimming control produces square wave signals. Since we are primarily interested in frequency $f_{pwm}$, without a loss of generality, a sinusoidal wave can be used to describe the PWM signals we are interested in, as follows:

$$T(x_s, t) = \cos(2\pi f_{pwm}(t + x_s\Delta_S) + \theta) \quad (1)$$

where $T(x_s, t)$ denotes the signal emitted from the $x_s^{th}$ row of the screen at time $t$. $x_s$ ranges from $0, 1, ..., X_s - 1$ where $X_s$ indicates the number of rows across the entire screen. $\theta$ denotes the initial phase of the signal in the first row.

When using a rolling shutter camera to capture an image from an OLED screen, the screen with $X_s$ rows is mapped to $X_{s\to c}$ rows in the captured picture (as shown in Fig. 8). In other words, we assume that $M$ rows on the camera sensor

[1]A pixel requires a larger current during its PWM duty cycle. By introducing a phase latency between rows, the total current required by the all pixels remain more stable across time.

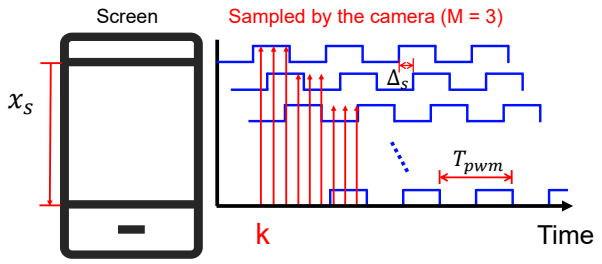



Fig. 7. Asynchronous PWM dimming in OLED screens and example of sampling using a rolling shutter camera where the projection ratio $M = 3$.

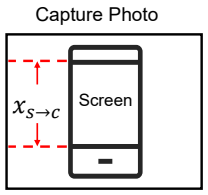
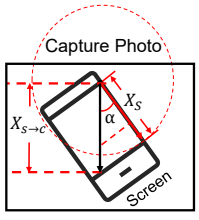
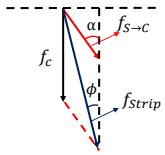



Fig. 8. Projection ratio between screen and camera.




(a) Projection ratio




(b) Stripe angle

Fig. 9. Projection ratio and resulting stripe angle in arbitrary direction.

capture one row from the screen, where the projection ratio $M$ can be denoted as follows:

$$M = \frac{X_{s\to c}}{X_s}$$

Fig. 7 presents an example of image sampling in which the screen and camera are placed in parallel to induce flicker and sampling is performed sequentially from top to bottom. Note that we later remove this assumption and model the interaction under arbitrary screen placements. The red arrows indicate the samples obtained by individual camera rows. $M = 3$ indicates that three camera rows capture the same row on the screen. Assuming that the camera begins sampling at time $k$, the captured signals can be written as follows:

$$\begin{aligned} R(x_c,t) &= \cos(2\pi f_{pwm}((t+k+x_c\Delta_C)+x_s\Delta_S)+\theta) \\ &= \cos(2\pi f_{pwm}(t+k)+2\pi f_{pwm}(x_c\Delta_C+\lfloor\frac{x_c}{M}\rfloor\Delta_S)+\theta) \end{aligned} \tag{2}$$

where $R(x_c, t)$ represents the signal received at camera row $x_c$ and time $t$. $x_c$ ranges from $0...X_c - 1$ where $X_c$ indicates the number of camera rows. $\Delta_C$ represents the rolling shutter interval (Fig. 3).

Thus, the intensity of pixels in $x_c$ row, $I(x_c)$, is the sum of photons received throughout the entire exposure, which can be derived as follows:

$$I(x_c) = \int_{t=k+x_c\Delta_C}^{k+x_c\Delta_C+T_e} R(x_c,t)dt \tag{3}$$

where $T_e$ represents the exposure duration.

### C. Sampling Under Arbitrary Screen Placement

It would be unreasonable to expect all cellphone users to hold their devices at precisely the same angle while the QR code is being read. Thus, we considered sampling with the mobile device held at an arbitrary angle relative to the camera. Fig. 9(a) shows the situation in which the screen is placed at angle $\alpha$ relative to the direction of the rolling shutter. For this arbitrary case, we can obtain a new projection ratio as follows:

$$M = \frac{X_{s\to c}\cos\alpha}{X_s} \tag{4}$$

As in Eq.2, the received signal from line $x_c$ at cross-angle $\alpha$ ($0° \le \alpha \le 360°$) can be modeled as follows:

$$\begin{aligned} R(x_c,t,\alpha) = &\cos(2\pi f_{pwm}(t+k) \\ &+ 2\pi f_{pwm}(x_c\Delta_C+\lfloor\frac{x_c cos\alpha}{M}\rfloor\Delta_S)+\theta) \end{aligned}$$

The intensity of pixels in row $x_c$ indicated by $I(x_c)$ throughout the exposure can be derived as follows:

$$I(x_c) = \int_{t=k+x_c\Delta_C}^{k+x_c\Delta_C+T_e} R(x_c,t,\alpha)dt$$

Specifically, the stripe pattern changes as a function of device rotation and distance from the camera (Sec. VI-A). Fig. 9(b) indicates the variation in angle due to rotation and projection ratio $M$. The scanning performed by the rolling shutter camera and the PWM signal can be considered frequency vectors $f_c$ and $f_s$, which are derived as follows:

$$f_c = \frac{1}{\Delta_C}, f_s = \frac{1}{\Delta_S}$$

Due to the degree of projection ratio between the screen and camera, it is possible to convert the actual PWM signal frequency vector into a capture coordinate as $f_{S\to C} = f_s M$. The combined stripe pattern vector (denoted by $f_{Strip}$) is shown in Fig. 9(b). Thus, the resulting angle of the stripe patterns relative to the horizontal (denoted by $\phi$) can be derived as follows:

$$\tan\phi = \frac{f_{S\to C}\cos\alpha + f_c}{f_{S\to C}\sin\alpha} = \frac{\Delta_C M\cos\alpha + \Delta_S}{\Delta_C M\sin\alpha} \tag{5}$$

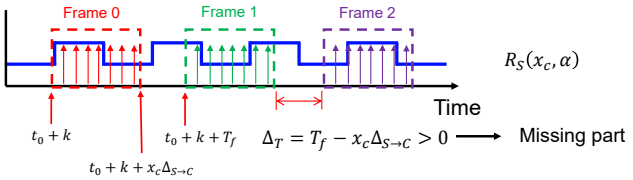
when $\alpha = 0°$ or $180°$, the refresh direction is the same or the inverse direction of the rolling shutter, such that the stripe patterns are aligned parallel to the bottom of the image.

### D. Sampling Using Multiple Frames

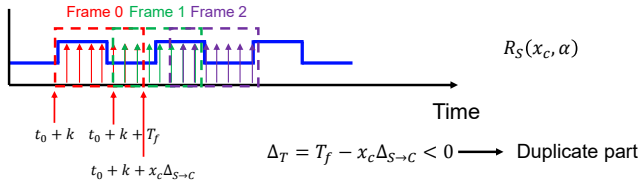
The frequency resolution is limited by the length of the data; If we capture only one photo, the frequency resolution is insufficient to determine PWM frequencies of different screens. One intuitive solution is to concatenate multiple frames of a video. However, concatenating frames from an OLED screen is a nontrivial problem, due to the presence of gaps between the frames. The start of exposure in the first row in the previous frame is denoted as $t$, whereas the start of exposure in the first row in the subsequent frame is $t + T_f$, where $T_f$ denotes the frame duration, as shown in Fig. 3.

Thus, the received signal associated with continuous captures through multiple frames can be denoted as follows:

$$\begin{aligned} R(x_c,t,\alpha,n) = &\cos(2\pi f_{pwm}(t+nT_f+k) \\ &+ 2\pi f_{pwm}(x_c\Delta_C+\lfloor\frac{x_c\cos\alpha}{M}\rfloor\Delta_S)+\theta) \\ &(n\in 0,1,2,...) \end{aligned} \tag{6}$$




(a) Case1: frame duration $T_f >$ sampling duration $x_c\Delta_{s\to c}$




(b) Case2: frame duration $T_f <$ sampling duration $x_c\Delta_{s\to c}$

Fig. 10. Problems associated with desynchronization due to fluctuations in frame rates and sampling intervals.

where $n$ denotes the frame number beginning at 0.

Note that Eq. 6 can be treated as a spatial frequency related to row index $x_c$ at specific time $t+nT_f+k$. $\cos(2\pi f_{pwm}(t+nT_f+k)$ can be regarded as the initial phase in the spatial domain, denoted by $\theta_0$. Thus, transforming Eq. 6 into time domain using one sample per line, we can derive the true sampling interval in the communication between screen and camera as follows:

$$\Delta_{s\to c} = \frac{x_c\Delta_C + \lfloor \frac{x_c \cos\alpha}{M} \rfloor \Delta_S}{x_c} \tag{7}$$

where the sampling rates for extracting frequency is denoted by $f_s = 1/\Delta_{s\to c}$.

However, variations among cameras in terms of rolling shutter effects lead to differences in $\Delta_C$ and interval $\Delta_S$ between the lines in the PWM signal. Variations in frame rates and sampling intervals can also lead to *information duplication* or *information loss*. Essentially, situations involving desynchronization can be divided into two cases:

- **Case1:** Frame duration $T_f$ exceeds the sampling duration $x_c\Delta_{s\to c}$, leading to information loss.
- **Case2:** Frame duration $T_f$ is shorter than the sampling duration $x_c\Delta_{s\to c}$, leading to information duplication.

Fig. 10 presents an example of desynchronization due to variations in frame rates and sampling intervals in three continuous frames. As shown in Fig. 10(a), there exists a time gap $\Delta_T = T_f - x_c\Delta_{s\to c}$ between frame 0 and frame 1 when $\Delta_T > 0$. The signal transmitted during this gap period is not detected in any of the received frames, such that the captured video is non-continuous (incomplete).

*Information duplication* can occur in two sequential frames when $\Delta_T < 0$. As shown in Fig. 10(b), frame 0 and frame 1 contain a portion of the tail replicated from frame 0. In these situations, the camera receives identical stripe signals in two sequential frames. Recovering the original signal requires interpolation to deal with *information loss* or the removal of duplicated frame sections and concatenation of the remaining sections to deal with *information duplication*, where the length of the lost or duplicated data is denoted by $\frac{\Delta_T}{\Delta_{s\to c}}$. Specifically, we insert a specific number of zeroes (matching the length of the lost data) into the interval for *information loss*. Conversely, we remove a specific number of tails (matching the length of the data duplication) for *information duplication*.

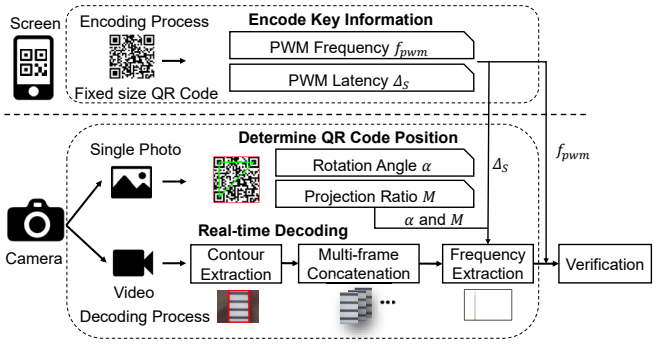



Fig. 11. Overview of SCREENID system.

## VII. SYSTEM DESIGN

Fig. 11 illustrates the architecture of SCREENID system, comprising two parts: encoding by the sender (smartphone screen) and decoding by the receiver (camera).

### A. Encoding

Verifying the authenticity of a screen requires that additional information be embedded in the QR code, namely $(i)$ PWM frequency $f_{pwm}$ and $(ii)$ PWM latency between two rows $\Delta_S$. The size of the QR code to be generated is adjusted automatically based on the resolution and size of the screen, such that the length of the edge is of the fixed number of pixels (580 pixels in our implementation) across screens.

Using SCREENID requires that $f_{pwm}$ and $\Delta_S$ are both known in advance. $f_{pwm}$ can be obtained when the user first uses the system and presents his/her phone to the cashier. The cashier then uses the Frequency Extraction scheme to compute $f_{pwm}$. In practice, the device on the cashier side would use 10 $f_{pwm}$ measurements from which to derive the median $f_{pwm}$. Estimating PWM latency $\Delta_S$ is not a trivial matter; therefore, we propose the method below.

### B. Determining PWM Latency $\Delta_S$

PWM latency $\Delta_S$ is determined from an image captured at an angle of 90° (i.e., $\alpha = 90°$ in Fig. 9) using a camera with a known configuration. As indicated in Eq. 5, when $\alpha = 90°$, the angle of stripe $\phi$ is a function of $\Delta_S$ that satisfied $\tan\phi = \frac{\Delta_S}{\Delta_C M}$, therefore $\Delta_S = \Delta_C M \tan\phi$, where $\Delta_C$ is the rolling shutter interval and $M$ is projection ratio computed using Eq. 4.

In our preliminary analysis, we found little variation in PWM latency $\Delta_S$ among phones of the same model; therefore, it was necessary to measure $\Delta_S$ only once for each phone model. This could easily be achieved using crowdsourcing

(a) Original code. (b) Rotation angle. (c) Contour.

Fig. 12. Methods used to obtain the rotation angle and projection ratio using a QR code of fixed size

where one user of a given phone model performs the calibration and shares the results. Likewise, this information could be provided by the manufacturer when the device is first released.

### C. Decoding

Verifying that the captured QR code is authenticated involves embedding the rolling shutter interval $\Delta_C$ in camera and PWM latency $\Delta_S$ in the QR code for use in estimating the PWM frequency. The QR code is accepted only if the estimated PWM frequency and the PWM frequency embedded in the QR code are a close match lower than a threshold.

The decoding process is performed in two steps: (i) determining the position of the QR code and (ii) extracting the PWM frequency.

### D. Measuring Rolling Shutter Interval $\Delta_C$

Rolling shutter interval $\Delta_C$ must be known a priori in order to compute the PWM frequency. However, we observed inaccuracies in the rolling shutter interval provided by Android Camera2 API [32]. Therefore, we programmed SCREENID to calibrate camera to obtain an accurate indication of rolling shutter interval (denoted as $\widehat{\Delta_C}$). Calibration involves capturing an image of an LCD screen with a known PWM frequency $f_{pwm}$. Based on Eq. 6, the wavelength (i.e., the width of a pair of black and white stripes) denoted by $W_{lcd}$ can be derived as $W_{lcd} = 1/f_{pwm}/\widehat{\Delta_C}$. Thus, the accurate rolling shutter interval can be estimated as $\widehat{\Delta_C} = 1/f_{pwm}W_{lcd}$. Note that it is easy to calibrate for camera on cashier when carrying out factory examination.

### E. Optimizing the Camera Configuration.

The images obtained using the camera serve two purposes. The first purpose involves estimating the position of the QR code and extracting the information embedded. We set the camera to auto-mode to obtain the optimal configuration for ambient lighting conditions to obtain a clear image. The second purpose involves measuring the PWM frequency, which requires a clear indication of the stripes. The following configuration was adopted for capturing video frames:

**Exposure time.** Many smartphones use PWM dimming only under a low brightness ratio, which tends to extend the exposure time. Note however that an excessively long exposure time would allow the occurrence of more than one PWM cycle, thereby compromising stripes clarity. Thus, we configured SCREENID to operate at an exposure time of $0.3ms$.

**ISO.** SCREENID was configured to use the highest available ISO, as this tends to enhance contrast between the black and white stripes. High ISO operations also tend to generate noise; however, the fact that most of the noise it at lower frequencies means that it's easily filtered out using a highpass filter.

**Aperture.** SCREENID was configured to use the largest available aperture as this tends to blur the background, making it easier to extract the contours of the screen.

### F. Determining QR Code Position

Estimating the distance and angle between the screen and the camera involves taking a single photo in auto-mode in order to detect the QR code. The position symbols in the the QR code (blue blocks in Fig. 12(b)) to form a right-angle triangle (green in Fig. 12(b), which can be used to estimate rotation angle $\alpha$.

Locating the position markers makes it possible to extract the contours of the QR code (red in Fig. 12(c)) and measure the size of the QR code in the photo. The size of the displayed QR code is fixed; therefore, it is possible to compute projection ratio M as a fraction of the length of a pixel along the both sides of the original QR code.

### G. PWM Frequency Extraction Using Multiple Frames

After extracting essential information from QR code, the camera switches to video mode for frequency extraction. As shown in Fig. 11, this process includes three following steps:

**Contour Extraction:** Each frame is first transformed into a gray-scale image to enhance contrast. The image then undergoes denoising and binarization, before the boundaries are detected using a function based on the Sobel operator [33].

**Multi-frame Concatenation:** From among the multiple columns in the contour, we select the longest column as a data sample of screen in a frame to maximize the data length. The frequency resolution ($f_{res}$) of the Fourier transform is $f_{res} = f_{sample}/N$ [34], where $f_{sample}$ indicates the sampling rates and $N$ indicates the number of samples. Identifying the characteristic PWM frequency requires a frequency resolution of $f_{res} = 0.1Hz$ or smaller. This can be achieved by concatenating columns from multiple frames to increase $N$, as described in Sec. VI-D.

**Frequency Extraction:** For a usual condition, the camera of smartphones can sample at $f_{sample} = 80kHz$ with $4000$ columns at frame rates of $30fps$. Thus, it requires $\frac{f_{sample}}{f_{res} \times 4000 \times 30} = 6.7$ seconds. Improving system efficiency requires reducing the time required to capture frames, while estimating the PWM frequency with high precision.

This was achieved by exploiting the fact that the sampling rate (e.g., $80kHz$) usually exceeds the Nyquist rate required to reconstruct PWM signals (e.g., $500Hz$). In order to estimate the PWM frequency with hight precision, we improve the frequency resolution by utilizing the zero-padding FFT [38]. Zero-padding FFT is a well-known method to refine the frequency granularity by padding zeros. The target FFT resolution $f_{res}$ is $f_{res} = f_{sample}/N$. Thus, the number of points we need can be calculated as $N = f_{sample}/f_{res}$. As for a sequence of samples of length $n$ (e.g. $[x_1, x_2, x_3, \cdots, x_n]$), we add

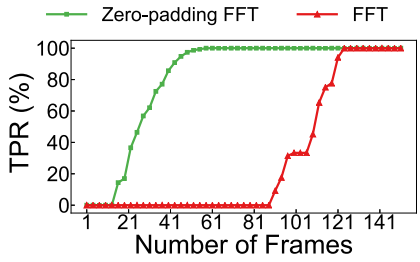
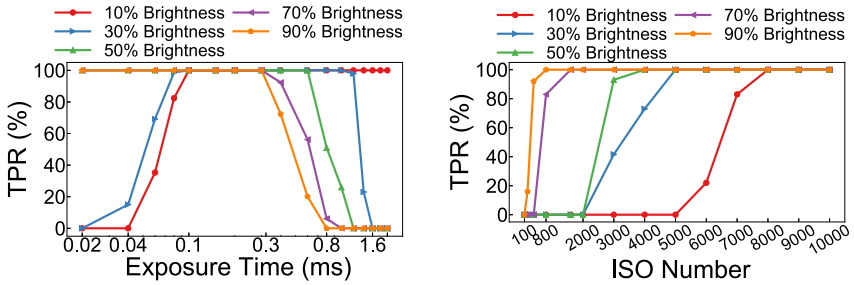



Fig. 13. Verification TPR of different frequency extraction schemes.




(a) Impact of exposure time.

(b) Impact of ISO number.

Fig. 14. The TPR on impact of exposure time and ISO number under different brightness ratio.

zeros to the end of time-domian signal of each frame to increase the whole length to $N$ to obtain sufficient frequency resolution. Then, we apply the Hanning window [35] to make the concatenation smoother and finally use Fourier transform to extract PWM frequency from the collected data.

**Verification:** SCREENID compares the estimated PWM frequency with the one embedded in the QR code. When the difference is less than a given threshold ($0.03Hz$ in current study), the QR code is accepted. Otherwise, it is rejected.

## VIII. EVALUATION

### A. Experiment Setup

We generate version-3 ($29 \times 29$) QR codes[2] using SCREENID.The size of QR codes is $580 \times 580$ pixels. 50 smartphone screens and 5 smartphone cameras are used in our evaluation. Among 50 screens, 30 screens are of the same model.

Unless otherwise stated, we evaluated SCREENID as follows. For each screen, we displayed 40 QR codes where 10 embedded the correct PWM frequency of the screen for authentication while the other 30 embedded that of another screen randomly selected from 50 screens for attack.

We utilize *TPR (True Positive Rate)* and *FPR (False Positive Rate)* as the metrics for performance evaluation, which are defined as follows:

- $TPR = \frac{Number\ of\ accepted\ authorized\ QR\ code}{Number\ of\ verification\ attempts}$
- $FPR = \frac{Number\ of\ accepted\ unauthorized\ QR\ code}{Number\ of\ attack\ attempts}$

The higher *TPR* is and meanwhile the lower *FPR* is, the better SCREENID performs.

### B. Microbenchmark

*1) Number of Required Frames:* Here we evaluate how many frames are required to accurately estimate the PWM frequency. We used 5 cameras to capture a 60-second video of each of 50 screens. We compare traditional FFT with zero-padding FFT. The average TPR under various number of frames is reported in Fig. 13. Traditional FFT needed 121 frames to get enough frequency resolution to achieve 100% TPR. Zero-padding FFT [38] can refine the frequency granularity by padding zeros. Results show that zero-padding FFT only needs 65 frames to extract correct PWM frequency, respectively. In summary, zero-padding FFT is 1.86 $\times$ faster than traditional FFT in the whole processing time.

*2) Camera Configuration:* To find the best exposure time and ISO, we measure the TPR under various settings and show the results in Fig. 14. For the exposure time, we balance the brightness ratios and set exposure time to $0.3ms$. For ISO, we can see that the higher ISO is, the higher the TPR is; therefore, we set ISO to the highest available for the camera.

*3) Threshold:* SCREENID is authorized only when the difference between estimated frequency and true frequency smaller than the threshold. However, when the threshold become larger, the TPR and the FPR both become higher. Fig. 15 shows the results of TPR and FPR on the impact of threshold. To consider a better trade-off, we select $0.03Hz$ as the threshold where the average TPR is above 94.3% across the same model, the FPR is below 0.8% across the same model and 0.5% across our dataset.

### C. Overall Performance

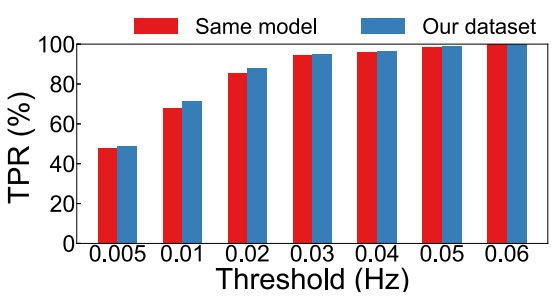
We evaluate the overall performance using 5 cameras and 50 screens. Fig. 16 shows the TPR and FPR of 50 screens. The first 30 screens are of the same model. The average TPR is 98.9% and 94.3% for all screens and phones of the same model. The TPR for each screen is higher than 92%. The FPR for each screen in the same model is lower than 1.5%, and for other models, the FPR is 0%, respectively. The results illustrate that SCREENID can successfully enhance the security of QRCode system.

## IX. SECURITY ANALYSIS

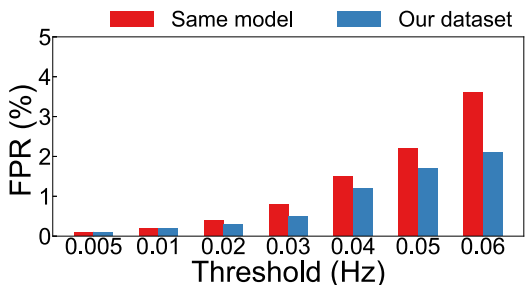
In this section, we discuss various attacks and the ability of SCREENID to deal with them. We assume that the attacker is able to determine the PWM frequency, but is not cooperating with the cashier responsible for scanning the QR code.

**Can attackers mimic the victim's PWM frequency by changing their phone's PWM frequency?** PWM dimming is applied to an on-chip display controller [23] using one of two methods [42]. The first is to control the display controller via $I^2C$ interface, and produce corresponding PWM signals. The attackers cannot change their PWM frequency without replacing the display controller chip. The second involves

[2]Alipay uses version-2 ($25 \times 25$) QR codes [37] while WeChat uses version-1 ($21 \times 21$) [37] QR codes. Version-3 QR codes which carry more data are representative of the amount of data needed by these systems.

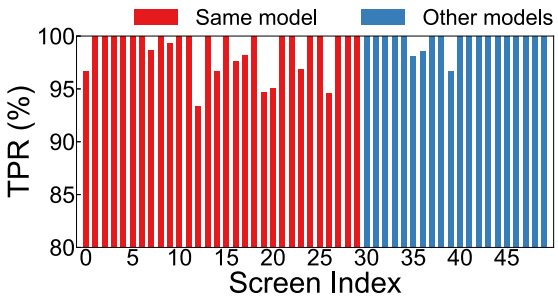



(a) TPR on the impact of threshold

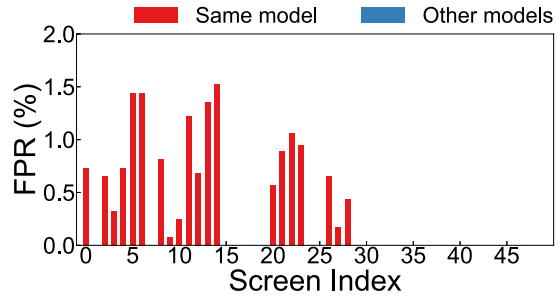



(b) FPR on the impact of threshold

Fig. 15. The TPR and FPR on the impact of threshold of 50 screens we collected (30 are of the same model).




(a) TPR for each screen.




(b) FPR for each screen.

Fig. 16. The TPR and FPR of 50 screens where the first 30 (red) are of the same model. The two groups are ordered by their true PWM frequencies.

the CPU outputting a PWM signal through a pin connected to display controller. The attackers should have the root permission to modify low-level drivers and identify the CPU pin if he want to alter PWM frequency. However, increasingly strict security restriction on the Android system [43] makes it nearly impossible to modify PWM frequency.

**Can attackers mimic the victim's PWM frequency by adding rolling stripes over the QR codes?** According to Eq. 6, the stripes on the screen (i.e., the victims screen) are a function of PWM frequency $f_{pwm}$ as well as the rolling shutter interval $\Delta_C$. The fact that the attacker does not know the configuration of the camera on cashier (i.e., $\Delta_C$) makes it impossible to produce the correct rolling stripes over the QR codes; i.e., reflection of QR code on glass of cashier only obtain $\Delta_C$ of front camera, not the camera of cashier.

**Can attackers mimic the victim's PWM frequency by using an external lighting device?** If we assume that the attacker has a lighting device capable of flickering at an arbitrary rate, it might be possible to light their screen at a rate mimicking the PWM frequency of the victim. However, we do not view this kind of attack method as practical, as it would be easily detected by the cashier.

**Can attackers happen to have a phone with the same PWM frequency as that of the victim's phone?** In the proposed system, the threshold we select is $0.03Hz$. As shown in Fig. 4(b), $99.8\%$ of the pairwise difference in frequency is larger than $0.03Hz$. This would require that an attacker use $450$ phones to achieve $90\%$ success rate. Such an attack would indeed be possible; however, it would be prohibitively expensive. SCREENID is meant to enhance the security of QR codes; however, it cannot guarantee security. Note that SCREENID does not need customized techniques so that it can meanwhile incorporate with existing authentication schemes.

In summary, SCREENID provides robust protection against attacks using hardware (not easily manipulated) and joint features ($f_{pwm}$, $\Delta_S$, and $\Delta_C$), which are difficult to compromise at the same time. It is expected that PWM dimming will become the overwhelming standard as OLED gain popularity [44] thanks to their light weight, wide viewing angle, and high color accuracy [23].

## X. CONCLUSION

In this study, we proposed SCREENID to enhance the security of QR codes by fingerprinting the screens displaying the QR codes. Only QR codes displaying on its specific screen are accepted. SCREENID exploits PWM frequency as a unique screen fingerprint and model the interaction between the camera and the screen in both temporal and spatial domains to extract high estimation of PWM frequency. Extensive experiments demonstrate the efficacy of SCREENID in differentiating screens to enhance the security of QR codes in real-world situations.

## ACKNOWLEDGEMENT

This work is supported in part by the NSFC (U1736207, 62072306, 61936015) and Startup Fund for Youngman Research at SJTU and Program of Shanghai Academic Research Leader (20XD1402100).

## REFERENCES


[1] Alipay. Alipay: Experience fast, easy and safe online payments., 2020.

[2] WeChatPay. Wechat pay: Beyond payment, leading mobile payment platform, 2020.

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

[4] JobTubeDaily. Your mobile money could be stolen by this attack!, 2018.

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

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

[7] Changsheng Chen. Qr code authentication with embedded message authentication code. *Mobile Networks and Applications*, 22(3):383–394, 2017.

[8] Ching-Nung Yang, Jung-Kuo Liao, Fu-Heng Wu, and Yasushi Yamaguchi. Developing visual cryptography for authentication on smartphones. In *International Conference on Industrial IoT Technologies and Applications*, pages 189–200. Springer, 2016.

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

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

[11] S Thamer and B Ameen. A new method for ciphering a message using qr code. *Comput. Sci. Eng.*, 6(2):19–24, 2016.

[12] Hao Pan, Yi-Chao Chen, Lanqing Yang, Guangtao Xue, Chuang-Wen You, and Xiaoyu Ji. mqrcode: Secure qr code using nonlinearity of spatial frequency in light. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–18, 2019.

[13] Jonathan M McCune, Adrian Perrig, and Michael K Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *2005 IEEE Symposium on Security and Privacy (S&P'05)*, pages 110–124. IEEE, 2005.

[14] Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham Mysore, Fredo Durand, and William T. Freeman. The visual microphone: Passive recovery of sound from video. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 33(4):79:1–79:10, 2014.

[15] Christos Danakis, Mostafa Afgani, Gordon Povey, Ian Underwood, and Harald Haas. Using a cmos camera sensor for visible light communication. In *2012 IEEE Globecom Workshops*, pages 1244–1248. IEEE, 2012.

[16] Chi Zhang and Xinyu Zhang. Litell: robust indoor localization using unmodified light fixtures. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 230–242, 2016.

[17] Shilin Zhu and Xinyu Zhang. Enabling high-precision visible light localization in today's buildings. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 96–108, 2017.

[18] Chi Zhang and Xinyu Zhang. Pulsar: Towards ubiquitous visible light localization. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 208–221, 2017.

[19] Shilin Zhu, Chi Zhang, and Xinyu Zhang. Automating visual privacy protection using a smart led. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 329–342, 2017.

[20] Kuan-Chieh Liao and Wei-Hsun Lee. A novel user authentication scheme based on qr-code. *Journal of networks*, 5(8):937, 2010.

[21] Zhe Zhou, Di Tang, Wenhao Wang, Xiaofeng Wang, Zhou Li, and Kehuan Zhang. Beware of your screen: Anonymous fingerprinting of device screens for off-line payment protection. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 77–88, 2018.

[22] P Punithavathi and S Geetha. Visual cryptography: A brief survey. *Information Security Journal: A Global Perspective*, 26(6):305–317, 2017.

[23] Michael Barr. Pulse width modulation. *Embedded Systems Programming*, 14(10):103–104, 2001.

[24] Hidemitsu Sugiyama, Shinichiro Haruyama, and Masao Nakagawa. Brightness control methods for illumination and visible-light communication systems. In *2007 Third International Conference on Wireless and Mobile Communications (ICWMC'07)*, pages 78–78. IEEE, 2007.

[25] Chia-Kai Liang, Li-Wen Chang, and Homer H Chen. Analysis and compensation of rolling shutter effect. *IEEE Transactions on Image Processing*, 17(8):1323–1330, 2008.

[26] NotebookCheck. Pwm ranking - the best displays for the eyes, 2019.

[27] Analog Devices. Adpd2212 datasheet, 2016.

[28] DeviceSpecifications. Device specifications, 2019.

[29] Bridget O'Donnell. Steals money sneakily by scanning people's qr code., 2019.

[30] Tao Li. Qr code scams rise in china, putting e-payment security in spotlight., 2019.

[31] Jan Genoe, Koji Obata, Marc Ameys, Kris Myny, Tung Huei Ke, Manoj Nag, Soeren Steudel, Sarah Schols, Joris Maas, Ashutosh Tripathi, et al. 30.2 digital pwm-driven amoled display on flex reducing static power consumption. In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 488–489. IEEE, 2014.

[32] GoogleDevelopers. Sensor rolling shutter skew of camera2, 2020.

[33] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.

[34] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[35] Alfredo Testa, Daniele Gallo, and Roberto Langella. On the processing of harmonics and interharmonics: Using hanning window in standard framework. *IEEE Transactions on Power Delivery*, 19(1):28–34, 2004.

[36] L Rabiner, RW Schafer, and C Rader. The chirp z-transform algorithm. *IEEE transactions on audio and electroacoustics*, 17(2):86–92, 1969.

[37] Jeroen Steeman. Qr code data capacity, 2019.

[38] Shannon Hilbert. Fft zero padding, 2013.

[39] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[40] Qualcomm. Hexagon dsp sdk, 2019.

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

[42] Texas Instruments. Lm36923 highly efficient triple-string white led driver, 2015.

[43] Tiwari Mohini, Srivastava Ashish Kumar, and Gupta Nitesh. Review on android and smartphone security. *Research Journal of Computer and Information Technology Sciences*, 2320:6527, 2013.

[44] statista. Shipments of flexible and rigid amoled panels worldwide from 2015 to 2022, 2018.