This paper appears as: I. Barron, H. J. Yeh, K. Dinesh and G. Sharma, "Dual Modulated QR Codes for Proximal Privacy and Security," IEEE Trans. Image Proc., vol. 30, 2021, DOI: 10.1109/TIP.2020.3037524.

Dual Modulated QR Codes for Proximal Privacy and Security

Irving Barron, Hsin Jui Yeh, Karthik Dinesh, and Gaurav Sharma, *Fellow, IEEE*

***Abstract*—The ubiquitous presence of surveillance cameras severely compromises the security of private information (e.g. passwords) entered via a conventional keyboard interface in public places. We address this problem by proposing dual modulated QR (DMQR) codes, a novel QR code extension via which users can securely communicate private information in public places using their smartphones and a camera interface. Dual modulated QR codes use the same synchronization patterns and module geometry as conventional monochrome QR codes. Within each module, primary data is embedded using intensity modulation compatible with conventional QR code decoding. Specifically, depending on the bit to be embedded, a module is either left white or an elliptical black dot is placed within it. Additionally, for each module containing an elliptical dot, secondary data is embedded by orientation modulation; that is, by using different orientations for the elliptical dots. Because the orientation of the elliptical dots can only be reliably assessed when the barcodes are captured from a close distance, the secondary data provides "proximal privacy" and can be effectively used to communicate private information securely in public settings. Tests conducted using several alternative parameter settings demonstrate that the proposed DMQR codes are effective in meeting their objective – the secondary data can be accurately decoded for short capture distances (6 in.) but cannot be recovered from images captured over long distances ($>$ 12 in.). Furthermore, the proximal privacy can be adapted to application needs by varying the eccentricity of the elliptical dots used.**

***Index Terms*—security, surveillance systems, privacy, quick response (QR) codes, 2-D barcodes.**

I. Introduction

Nowadays, video surveillance systems are increasingly prevalent and in many modern cities across the world, cameras can be found in almost all public areas such as streets, airports, train/bus stations, and markets [1]. Although these surveillance cameras are deployed to enhance security, they can also inadvertently pose severe risks for information security. For example, private data, such as passwords, can be revealed and compromised when users entering these on a keyboard are recorded in surveillance video. Users can choose to avoid some physical actions and behaviors in public settings. However, in an increasingly electronic and connected world, it is unviable to entirely avoid the communication of private information in public settings. Passwords are a key example of personal information that users are often forced to enter into devices in public settings in order to gain access. Intelligently designed attacks can effectively obtain the password from a short recorded video of a person typing in the password, even when the keyboard is not directly visible [2], [3]. By combining movement tracking with text analysis techniques for sentence reconstruction and error correction, one can also use keyboard video to eavesdrop on grammatically structured text in an automated fashion [4].

Users have few feasible options to ensure the privacy of text that they enter on a keyboard in public settings. As an extreme measure, users can choose to completely conceal themselves physically, as was done by Edward Snowden in an interview, when he covered himself and his laptop computer under a blanket to enter his password [5]. However, such extreme measures are neither socially acceptable nor always feasible (not everyone carries a large enough blanket or sheet with them!). For smartphone and tablet devices, the increasing deployment of biometric authentication modes partly mitigates the problem of keyboard based entry of passwords in public settings. However, the problem remains prevalent for laptop computers, kiosks, and other devices where a keyboard remains the predominant mode of input.

We propose a method to address the aforementioned challenges by using a novel extension of QR codes that we refer to as dual modulated QR (DMQR) codes. DMQR codes use synchronization marker patterns identical to QR codes and also an identical geometry of data carrying modules, square non-overlapping tiles that, in conventional QR codes, carry individual bits of data based on whether they are black or white. By reusing these elements, DMQR codes inherit QR codes fast localization capabilities and also the ability to compensate for variations in capture geometry. Different from conventional QR codes, however, DMQR codes use two different modulation schemes for embedding of two data streams, which we refer to as primary and secondary data. The primary data is embedded via intensity modulation compatible with conventional QR code readers. Specifically, based on each bit in the primary data, each module is either white or includes a black ellipse that reduces the module's brightness. The embedded primary data also includes error protection according to the QR code standard, which makes the primary data *decodable with conventional QR code decoders without requiring any modification.* For each module that contains an elliptical dot, secondary data is additionally embedded by modulating the orientation of the elliptical dot according to the values of the secondary data,

Manuscript received January 14, 2020; revised July 10, 2020 and September 9, 2020; accepted October 11, 2020. I. Barron's work was supported by Consejo Nacional de Ciencia y Tecnología (CONACYT), Mexico and by the University of Rochester. H.J. Yeh's work was supported by US National Science Foundation grant #1559970. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Husrev Sencar. (*Corresponding author: Gaurav Sharma.*)

I. Barron, H. J. Yeh, K. Dinesh, and G. Sharma are with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, 14627-0231, USA (e-mail: {ibarron., hyeh4@u., kdinesh@ur., gaurav.sharma@}rochester.edu).

which also includes error protection to enable error recovery. Because the orientations of the small elliptical dots can be reliably inferred only when displayed barcode images are captured from close distances (typically under 12 inches), *the secondary data offers privacy from surveillance cameras that are at relatively larger distances, a property that we refer to as "proximal privacy."* The level of proximal privacy can itself be adapted by varying the eccentricity of the ellipses used.

The proximal privacy allows users in public settings to securely communicate private data to a camera-equipped device from their smartphone by using the secondary data channel of DMQR codes displayed on their smartphones. Figure 1 illustrates this for one of the key application scenarios. Figure 1 (a) illustrates the current situation: a user's password is revealed in captured surveillance video when they enter it on a keyboard in a public setting, seriously compromising their private information. Figure 1 (b) illustrates the alternative enabled by the proposed DMQR codes. The user logs on by displaying a DMQR code on their mobile device that carries the password (obtained from a secure password vault on the smartphone) as the secondary data. The DMQR code is captured from a close distance by the webcam of the device that the user wishes to log on to and the secondary data corresponding to the password is readily recovered from the captured image allowing the user to log on. The surveillance cameras on the other hand can only capture the DMQR codes from a larger distance and therefore the password information cannot be recovered from the captured surveillance images/video.

We note that one could also envisage directly using QR codes for communication of private information in public settings. However, in these situations, the relative robustness of the QR codes becomes a liability, leaving open the possibility that the privacy will be compromised in captured images. As we describe in Section IV, appropriate protocols used in combination with DMQR codes can also allow for the proximal privacy protection of the secondary data to be extended to the primary data.

This paper is organized as follows. Section II describes the proposed DMQR codes, covering both the encoding and decoding for primary and secondary data. Section III first outlines the set up used for the experimental evaluation of the proposed DMQR codes and then presents results from both simulations and tests with actual deployments. Section IV includes a discussion that covers security aspects of DMQR codes for our target application as well as design and implementation choices for our experiments and application scenarios. Section V highlights prior work that relates to DMQR codes and contrasts DMQR codes with alternative technologies for private proximal communication. Finally, Section VI concludes the paper with a summary of our main findings. Additional implementation details and assessment of the proposed DMQRC codes in print are presented in Appendices A and B, respectively.

II. Dual Modulated QR Codes

Dual modulated QR codes use intensity and orientation modulation to embed primary and secondary data, respectively.

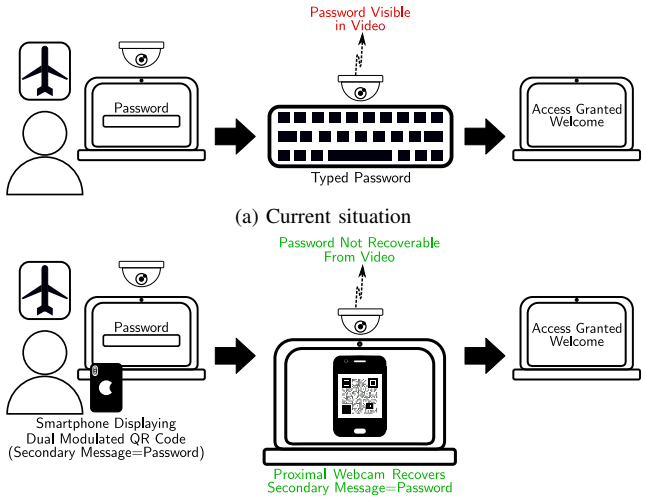


Fig. 1. Device log-on in a public video surveillance setting as a motivating application for the proposed DMQR codes. (a) current situation, video captures and compromises user password, and (b) approach where password is carried as secondary data in the proposed DMQR code and allows successful recovery and log-on from the close-distance webcam capture, but is not recoverable from video.

Figure 2 provides an overview of the encoder and decoder for DMQR codes. The DMQR codes use synchronization and alignment patterns identical to those for conventional QR codes and also the same geometry of modules that carry individual bits of data. Within each module, primary data is embedded via intensity modulation, which is compatible with conventional QR code detection. Specifically, depending on the bit to be embedded in the module, it is either left white, or an elliptical black dot is placed within the module. Additionally, for each module containing an elliptical dot, secondary data is embedded by orientation modulation; that is, by using different orientations for the elliptical dots. The secondary data is recovered at the decoder by estimating the orientations of the elliptical dots. Next we detail the individual operations within the encoder and decoder.

A. DMQR Code Encoder

The top half of Fig. 2 depicts the encoding process for the DMQR code. Both the primary and the secondary messages are first encoded with error correction codes and the resulting (encoded) primary and secondary data are then modulated into the DMQR code. We describe the process in turn for the primary and secondary messages.

1) Primary Message Encoding and Embedding: The primary message embedding procedure for our proposed DMQR codes is analogous to and compatible with conventional QR code encoding. In particular, we make use of synchronization patterns, alignment patterns, and the geometry of data carrying modules, identical to those in traditional QR codes. In Fig. 2, in the image labeled "Intensity Modulated QR Code" (and other similar images), the standard QR code synchronization patterns can be seen as the three distinctive identical patterns in the top-left, top-right, and bottom right corners comprising

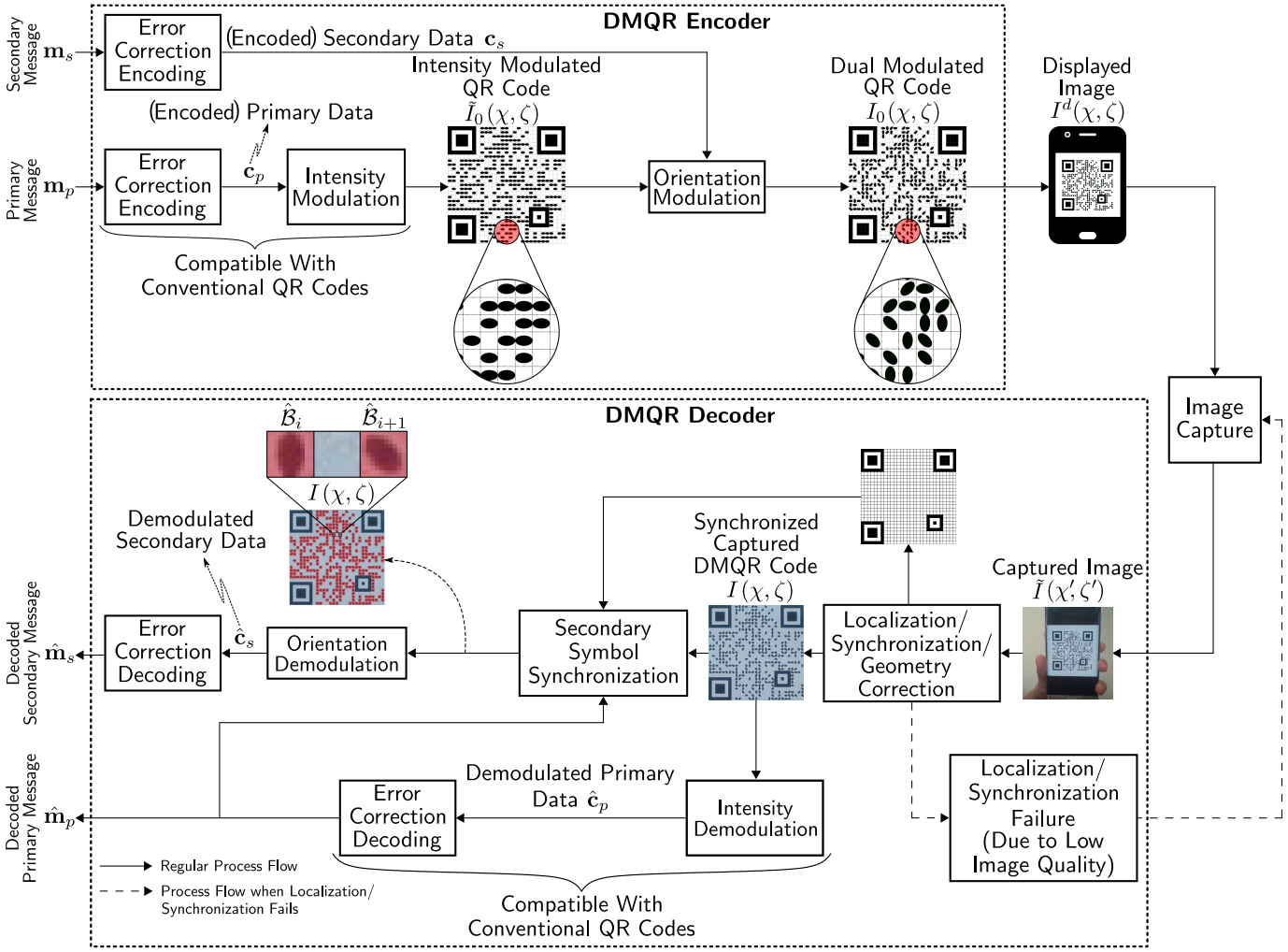


Fig. 2. End-to-End overview of DMQR code encoding and decoding. For details of the individual steps involved in the encoding and decoding, see description in Sections II-A and II-B, respectively.

of a hollow outer square with an inner solid square and the alignment pattern can be seen as the similar but smaller pattern located a little inside on the bottom right side. Together the synchronization and alignment patterns define the geometry of the modules used for the data embedding which are shown as the squares in the same image, bounded by dotted lines. The primary message $\mathbf{m}_p$ first goes through an error correction encoding as per the QR code standard [6] that yields the (encoded) primary data $\mathbf{c}_p$, with added redundancy for enabling error recovery. The primary data $\mathbf{c}_p$ is then embedded in the barcode pattern using intensity modulation. Specifically, bits of data from $\mathbf{c}_p$ are embedded in individual data carrying modules of the QR code; based on the (0/1) value of the bit being embedded either a black elliptical dot is placed in the module or it is left white (empty). As with conventional QR codes [6], prior to this embedding the bits are XOR-ed with one of eight possible binary mask patterns to reduce runs of identical consecutive values and to ensure roughly equal numbers of zeros and ones. Bits to identify the mask and the error correction encoding are also included in the data embedded in the modules so that the decoder can use the same mask to undo the XOR operation and also perform appropriate error correction decoding. Conceptually, this embedding procedure can be understood as producing an intensity modulated QR code represented by $\tilde{I}_0(\chi,\zeta)$, where (χ,ζ) represent the coordinate axes, aligned with the module grid. We note that thus far, the procedure for creating a barcode from the primary message is identical in all respects to that used for creating conventional QR codes, except that conventional QR codes make the entire module black or white whereas the DMQR codes "reduce the magnitude of intensity modulation" by replacing the black modules with black elliptical dots within the modules. Because of the commonalities with conventional QR codes, the DMQR codes inherit several of the beneficial properties of classic QR codes, viz., robust localization and synchronization, perspective distortion correction, and various levels of selectable error correction capability. Also, as long as the intensity modulation magnitude is appropriate, the embedded primary message can be directly decoded with a conventional QR code decoder. Importantly, the size of the embedded elliptical dot in comparison to the module size determines the magnitude of intensity modulation, a parameter that we shall discuss subsequently. For our subsequent discussion, we represent the spatial support of the i^{th} module containing a black elliptical dot as $\mathcal{B}_i$, for

$1 \leq i \leq N_B$, with N_B representing number of modules containing black elliptical dots.

2) Secondary Message Encoding and Embedding: The procedure for embedding the secondary message involves modulating the orientation of the black elliptical dots resulting from the primary data modulation within their corresponding N_B modules. First, the secondary message $\mathbf{m}_s$ undergoes error correction encoding, which adds redundancy to allow for error recovery, and yields (encoded) secondary data $\mathbf{c}_s$. The secondary data $\mathbf{c}_s$ is represented as a string of N_B M-ary symbols $\mathbf{c}_{s,1}, \mathbf{c}_{s,2}, \ldots, \mathbf{c}_{s,N_B}$, where $\mathbf{c}_{s,i} \in \{0, 1, \ldots (M-1)\}$. Based on the value taken by the M-ary symbol $\mathbf{c}_{s,i}$, the major-axis orientation angle $\phi_{\mathbf{c}_{s,i}}$ for the elliptical dot in the i^{th} module $\mathcal{B}_i$ is determined as,

$$\phi_{\mathbf{c}_{s,i}} = \frac{\pi \mathbf{c}_{s,i}}{M}. \tag{1}$$

Specifically, if Δ denotes the length of either side of the square modules and (χ_0^i, ζ_0^i) denote the coordinates of the center of the i^{th} module $\mathcal{B}_i$, the region satisfying the constraint

$$\begin{aligned} &\frac{((\chi - \chi_0^i)\cos\phi_{\mathbf{c}_{s,i}} + (\zeta - \zeta_0^i)\sin\phi_{\mathbf{c}_{s,i}})^2}{\Delta^2} + \\ &\frac{((\chi - \chi_0^i)\sin\phi_{\mathbf{c}_{s,i}} - (\zeta - \zeta_0^i)\cos\phi_{\mathbf{c}_{s,i}})^2}{(\epsilon\Delta)^2} \leq 1, \end{aligned} \tag{2}$$

is rendered as the black elliptical dot within the extent of the module, which is defined by $(\chi_0^i - \Delta/2) \leq \chi \leq (\chi_0^i + \Delta/2)$ and $(\zeta_0^i - \Delta/2) \leq \zeta \leq (\zeta_0^i + \Delta/2)$. Note that the length of the major axis of the ellipse matches the width/height Δ of the modules and is oriented along the angle $\phi_{\mathbf{c}_{s,i}}$. The eccentricity parameter ϵ, representing the ratio of the lengths of the minor and major axes, determines the eccentricity[1] of the elliptical dots, which in turn determines how discernible the changes in orientation are. The process yields the DMQR code $I_0(\chi, \zeta)$ that carries both the primary data $\mathbf{c}_p$ and the secondary data $\mathbf{c}_s$ embedded, respectively, with intensity and orientation modulation. Note that for conceptual explanation, we have described the primary and secondary message embedding processes sequentially as introducing black elliptical dots within modules and modulating their orientations. In actual practice, however, the encoding and modulation process for the primary message only needs to identify the N_B modules in which the elliptical dots are to be inserted and the dots can then be directly placed within the identified modules with their orientations determined by the (encoded) secondary data.

As an example, Fig. 3 illustrates $M = 4$-ary orientation modulation, where $\mathbf{c}_{s,i} \in \{0, 1, 2, 3\}$ and the major axis for the elliptical dots take the four possible orientations $\phi_{\mathbf{c}_{s,i}} \in \{0, \pi/4, \pi/2, 3\pi/4\}$. The DMQR code $I_0(\chi, \zeta)$ in Fig. 2 illustrates the complete DMQR code created using this $M = 4$-ary orientation modulation along with a blow-up of a small region that illustrates the dots and their orientations more clearly.

[1]The eccentricity of the ellipse is directly obtainable from ϵ as $\sqrt{1 - \epsilon^2}$.

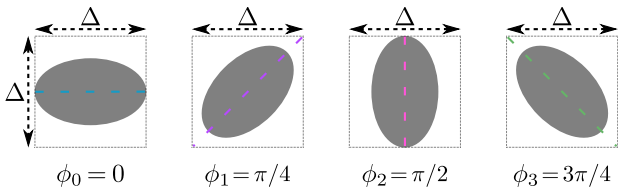


Fig. 3. Example illustrating $M = 4$-ary orientation modulation for the secondary data. The orientation of the ellipse in a module is determined by the corresponding two bits of secondary data as indicated by (1).

B. DMQR Code Decoder

As illustrated in Fig. 2, the decoder operates on a captured image $\tilde{I}(\chi', \zeta')$ from a displayed version $I^d(\chi, \zeta)$ of the DMQR code. Standard algorithms developed for QR codes can identify the locations of the synchronization and alignment patterns within the captured image $\tilde{I}(\chi', \zeta')$. From these locations perspective distortion introduced in the images due to deviations from an ideal fronto-parallel geometry can be estimated and corrected and synchronization can be established to identify the locations of the individual data carrying modules. With synchronization established[2], for notational simplicity we reuse the coordinate notation and represent the synchronized captured DMQR code image as $I(\chi, \zeta)$.

The embedded primary and secondary messages are decoded from the synchronized captured DMQR code image $I(\chi, \zeta)$ using the procedure shown in the bottom half of Fig. 2. Next we explain the process in turn for the primary and secondary messages.

1) Primary Message Decoding: The primary message is decoded using a conventional QR code decoder. The reader performs intensity demodulation, i.e., estimates a $(0/1)$ bit value for each data carrying module based on whether the module is white or dark, i.e., contains a black elliptical dot. By XORing with the binary pattern used for the encoding, an estimate $\hat{\mathbf{c}}_p$ of the encoded primary data is first obtained. An error correction decoder corresponding to the error correction code used at the encoder decodes the message $\hat{\mathbf{c}}_p$ to recover an estimate of the primary message, $\hat{\mathbf{m}}_p$.

2) Secondary Message Decoding: Additional secondary symbol synchronization is performed first by estimating which modules contained black elliptical dots. Specifically, using the decoded primary message $\hat{\mathbf{m}}_p$, the decoder can estimate the locations of the dot carrying modules by replicating the process used at the encoder for the primary message to identify the modules $\hat{\mathcal{B}}_i$, $i = 1, 2, \ldots N_B$ that (purportedly) carry the oriented black elliptical dots. Once secondary symbol synchronization is achieved, for simplicity of implementation, we use a version of the efficient heuristic orientation demodulation scheme proposed in [7]. The $I(\chi, \zeta)$ is binarized[3] to obtain $\bar{I}(\chi, \zeta)$ and one-dimensional image moments are computed along the M orientation modulation angles. Specifically, for

[2]If the localization/synchronization fails, for instance due to low quality of the captured image, a fresh image can be captured, as is typical in QR code decoding software applications that keep acquiring and processing frames sequentially while "hunting" for the localization/synchronization patterns.

[3]At proximal distances, for which the secondary message can be reliably decoded, we observe very similar performance when decoding from either the binarized image or the captured grayscale image.

$j = 0, 1, \ldots (M-1)$, the image moment along the orientation angle $\phi_j = \pi j/M$ is computed as

$$\mu_j^i = \frac{\sum_{\chi,\zeta \in \hat{\mathcal{B}}_i} \bar{I}(\chi,\zeta)(x_j - \bar{x}_j)^2}{\sum_{\chi,\zeta \in \hat{\mathcal{B}}_i} \bar{I}(\chi,\zeta)}, \tag{3}$$

where

$$x_j = \chi \cos\phi_j + \zeta \sin\phi_j, \tag{4}$$

corresponds to the abscissa for the coordinate system obtained by rotating the coordinates (χ, ζ) counter-clock-wise by the angle ϕ_j, and

$$\bar{x}_j = \frac{\sum_{\chi,\zeta \in \hat{\mathcal{B}}_i} \bar{I}(\chi,\zeta) x_j}{\sum_{\chi,\zeta \in \hat{\mathcal{B}}_i} \bar{I}(\chi,\zeta)}, \tag{5}$$

is the coordinate of the abscissa for the center of mass of the binarized image $\bar{I}(\chi, \zeta)$ in the rotated coordinate space. The angle ϕ_j that maximizes μ_j^i determines an estimated symbol $\hat{\mathbf{c}}_{s,i}$, which can be mathematically expressed as

$$\hat{\mathbf{c}}_{s,i} = \underset{j}{\operatorname{argmax}}\, \mu_j^i. \tag{6}$$

Fig. 4 illustrates the angles ϕ_j and the rotated abscissa axes x_j for $j = 0, 1, 2, 3$ for $M = 4$-ary orientation demodulation.

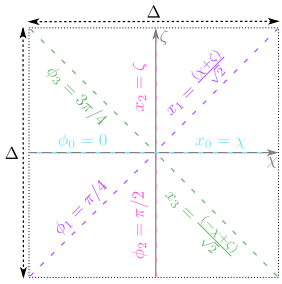


Fig. 4. Example illustrating the four different axis orientations along which one-dimensional image moments are computed within a module for demodulation of the secondary data embedded with $M = 4$-ary orientation modulation.

Finally, an error correction decoder corresponding to the code used to encode the secondary message decodes the demodulated secondary data $\hat{\mathbf{c}}_s$ to obtain an estimate $\hat{\mathbf{m}}_s$ of the secondary message.

C. DMQR Codes Parameters: M and ϵ

Apart from parameters inherited from the underlying QR code construction (e.g. module size, error correction level, etc.), DMQR codes have two additional parameters, M, the number of orientations used and ϵ, which controls the eccentricity of the elliptical dots used. The number of orientations M impacts the reliability with which individual secondary modulated symbols can be detected, as is standard in most multi-level modulation schemes for digital communications. The eccentricity parameter ϵ, which ranges between 0 and 1, impacts the modulation for both the primary and secondary data as illustrated in Fig. 5. For a value of $\epsilon = 1$, the ellipse becomes a circle and no meaningful orientation modulation is possible. As the value of ϵ is reduced, the eccentricity of the ellipse increases making the orientation modulation more apparent, however, as fraction of the module that is black reduces, which makes it more challenging to distinguish between modules with an ellipse and those without. Thus, the choice of ϵ trades-off detectability of the primary and secondary data modulations. Larger values of ϵ are favorable for the primary data modulation because they produce larger changes in the intensity between the white modules and the modules with the black elliptical dots, however, these larger values are less favorable for the secondary data modulation because the dots become more directionally symmetric making the orientation modulation harder to decipher. Vice versa, smaller values of ϵ are unfavorable to the primary data modulation because they produce smaller changes in the intensity between the white modules and the modules with the black elliptical dots but these smaller values are more favorable for the secondary data modulation because the dots are less directionally symmetric making the orientation modulation easier to decipher. Thus, the appropriate choice of ϵ is a critical choice for DMQR codes.

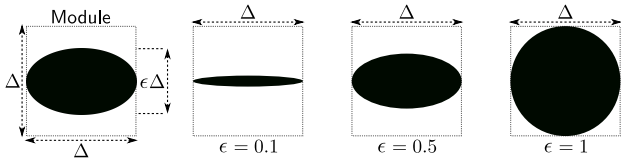


Fig. 5. The elliptical dots in DMQR codes for different values of the eccentricity parameter ϵ.

III. Experiments and Results

Motivated by our primary target application that seeks to use DMQR barcodes to communicate sensitive data in public places, we consider an experimental setting where a laptop webcam captures a DMQR code displayed on a smartphone. We first conducted simulations to determine an appropriate operating choice of the number of orientations M and the range for the eccentricity parameter ϵ. Then, based on the simulation results, we fixed M and selected a smaller range of values for the eccentricity parameter ϵ to conduct experiments with actual devices. Fig. 6 illustrates the set-up used for both the simulations and for experiments with actual display/capture devices. We first outline the common set-up elements and evaluation metrics used for both, before providing details and results for each.

A. Set-up/Evaluation Metrics

DMQR codes were created using the encoding process described in Section II-A with the primary message $\mathbf{m}_p$ as a URL and the secondary message as a secure password obtained from a strong password generator website [8]. Values $M = 2, 4, 8$ were used for embedding the secondary data. Error correction encoding of the primary and secondary messages was performed using, respectively, the level M error correction [6] (a Reed-Solomon code capable of correcting approx. 15% errors) for conventional QR codes and a rate $1/2$ convolutional code [9]. We note that to maintain compatibility

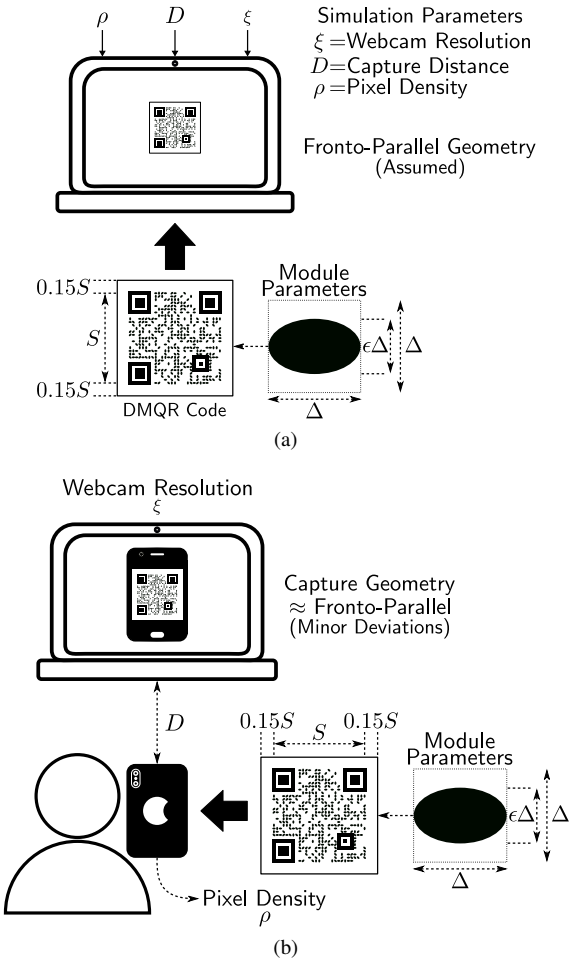


Fig. 6. Configuration for DMQR code experiments: (a) via simulations and (b) with actual devices.

with conventional QR code decoders, the error correction code for the primary data also needs to be compatible with the QR code standard [6], however, error correction for secondary data is not similarly constrained and can be chosen independently from among the many available options. Convolutional codes were chosen in our implementation because they are readily available and the lack of a block-length constraint makes it easy to handle the variable length of the secondary data without having to switch codes. To facilitate synchronization, a uniform white border 15% of the size S of the DMQR code along each dimension was added. For accomplishing localization and synchronization, and for decoding the primary data, we used ZXing [10], an open source QR code decoder, that handles the intensity demodulation and error correction decoding of $\hat{\mathbf{c}}_p$. Additional implementation details can be found in Appendix A.

To characterize performance, for both primary and secondary data we used two evaluation metrics: bit error rate (BER) which characterizes the performance at the modulation subsystem level and decoding success rate (DSR) for the messages, which quantifies the performance for the overall system. Specifically, the BER for the primary and secondary data indicates the percentage of erroneous bits, i.e., the percentage of bits in which $\mathbf{c}_p$ differs with $\hat{\mathbf{c}}_p$ and $\mathbf{c}_s$ differs with $\hat{\mathbf{c}}_s$, respectively. Similarly, the DSR for the primary and secondary messages indicates the percentage of barcodes from which the messages $\mathbf{m}_p$ and $\mathbf{m}_s$, respectively, are successfully recovered after error correction decoding.

B. Simulations

The simulations allow us to effectively explore the system performance for different choices of M and the aforementioned trade-off between primary and secondary data modulation robustness (see Section II-C) over a wide range of values for the eccentricity parameter ϵ, without being overly burdensome in terms of time requirements, and without being impacted by sources of experimental variation (though the simulations do include noise). Specifically, through the simulations, we investigated the performance of DMQR codes for $M = 2, 4, 8$ orientations and values of $\epsilon = 0.1, 0.2, \ldots, 1.0$.

Fig. 6(a) shows the simulation set-up: the DMQR code is displayed on a screen with pixel density ρ and captured with a webcam with resolution ξ from a distance D in an ideal fronto-parallel geometry. To conduct the simulations, we use the methodology described in [11], for which required software models for the display and the capture camera along with associated signal processing components are available as a toolkit [12]. Parameter settings appropriate for our display scenario were obtained by choosing one of the default screen options in [12], which corresponds to an Apple LCD monitor with a pixel density of $\rho = 326$ pixels per inch (PPI) that is comparable to an iPhone 7 display, representing the low end (and therefore more challenging) resolutions likely to be encountered in future applications of DMQR codes. The camera parameters were chosen to correspond to a fixed focus with a sensor resolution $\xi = 1280 \times 720$ pixels, which are also representative of the low end of what would be seen in current laptop webcams. Capture distances of $D = 3, 6, 9, 12, 15, 18$ inches were simulated. To realistically incorporate the effect of hardware compression that is often integrated within camera capture pipelines, the simulated captured images were stored as JPEG files with a quality factor of 90. Other simulation parameters, including image capture noise, were kept at the default values in [12]. The simulated captured images of DMQR codes on simulated displays were then used as input for our DMQR decoder implementation that was described previously and the BER statistics were computed for both the primary and secondary data over 250 Monte-Carlo simulations. For situations where the decoder failed to localize/synchronize, a BER of $1/2$ corresponding to random guessing was assumed.

Fig. 7 presents the simulation results. For $M = 2$, Figs. 7(a) presents percent BERs as a function of the eccentricity parameter ϵ and the capture distance D for the primary and secondary data overlaid as the purple and green 3D surfaces, respectively. Corresponding plots for $M = 4$, and 8 are shown in Figs. 7 (b) and (c), respectively. The plots provide a quantitative representation of the trade-off between the primary and secondary data recoverability as a function of the eccentricity parameter ϵ, which was qualitatively alluded to in Section II-C. Furthermore, important for our primary motivating application, the plot also indicates how this trade-off changes with the capture distance D. Because the recovery

TABLE I
DISPLAY AND CAPTURE DEVICES USED FOR THE EXPERIMENTS IN SECTION III-C

Capture Devices	Operating System and Capture Software	Camera Resolution (pixels)	JPEG Quality Factor	Display Devices	Pixel Density (PPI)
HP Pavilion Power 15-cb045wm	Linux Webcamoid	1280 × 720	75	Xiaomi Mi 8	402
Lenovo Yoga 730	Windows 10 Camera App	1280 × 720	90	Samsung Galaxy S6 Edge+	518
Dell Inspiron 13-7368	Windows 10 Camera App	1280 × 720	90	Apple iPhone XR	326
iPad 6th Generation	iOS Camera App	1280 × 960	100		

of the primary data is a pre-requisite for secondary symbol synchronization, we first examine the plots from the perspective of the primary data. The plots shows that across the range of capture distances simulated, the primary data is recoverable with rather low error rates for values of the eccentricity parameter ϵ in the range between 0.5 to 0.9. Furthermore, in the simulations, the secondary data exhibits the desirable proximal privacy property: over a significant portion of the aforementioned range of ϵ the BERs for the secondary data are quite low for small capture distances D but increase quickly with increasing capture distances, approaching the random guessing threshold of $1/2$ (50%) at around $D = 18$ in., indicating that no meaningful information about the secondary message is recoverable beyond this distance. Examining the variation in the secondary data BERs across the three different choices of M, we observe that $M = 2$ and 4 offer a reasonable operating envelope in terms of the proximal capture distance D and the eccentricity parameter ϵ, whereas for $M = 8$, the operating envelope is rather small.

C. Experiments with Actual Devices

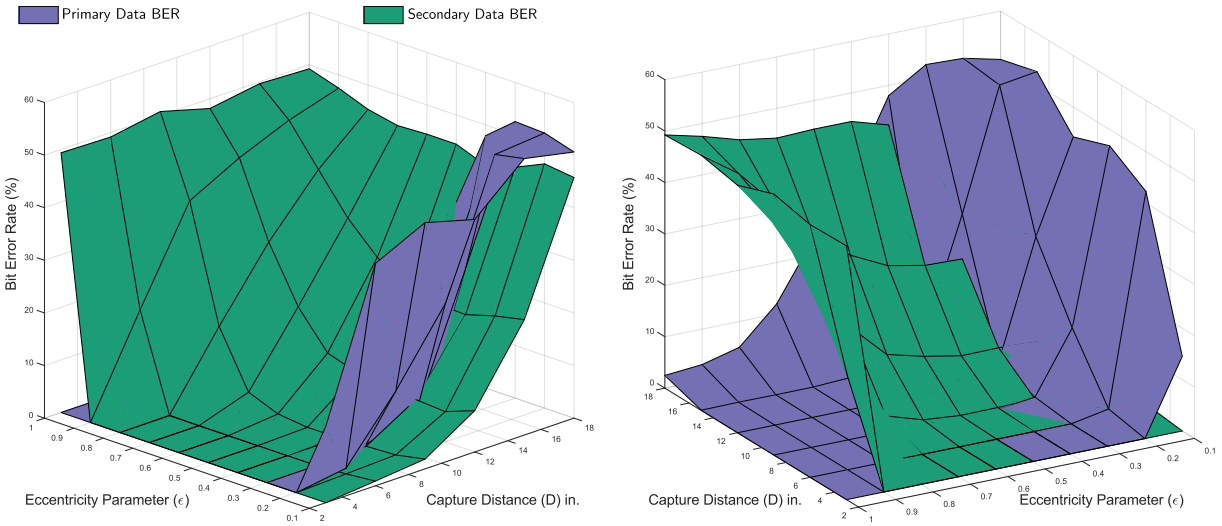
The set-up shown in Fig. 6(b) was used for the experiments. A smartphone and a laptop webcam were used for the display and capture, respectively, of the DMQR codes; other aspects mirror the simulation set-up. The pixel density ρ and camera resolution ξ are determined by the actual devices used and the capture distance D is physically varied instead of being simulated. In particular, we selected the capture distances $D = 6, 9, 12$ inches. Based on the simulation results, a subset of the DMQR codes, representing the effective operational range of the parameter space, was selected. Specifically, we chose $M = 4$-ary modulation[4] and eccentricity parameter values of $\epsilon = 0.5, 0.6, 0.7, 0.8, 0.9$. Furthermore, as a baseline against which DMQR code performance can be compared, we also tested a traditional QR code carrying the strong password that is embedded as the secondary message in the DMQR codes. A module size and (primary message) error correction level identical to that used in the DMQR codes was used for the conventional QR code.

The experiments used three different smartphones for displaying the DMQR/QR codes and to capture the images of these barcodes for decoding, we use three laptop webcams and a tablet's front facing camera. The specifications for the capture/display devices are listed in Table I. The captured images were then used for decoding the DMQR/QR codes using our decoder implementation described earlier.

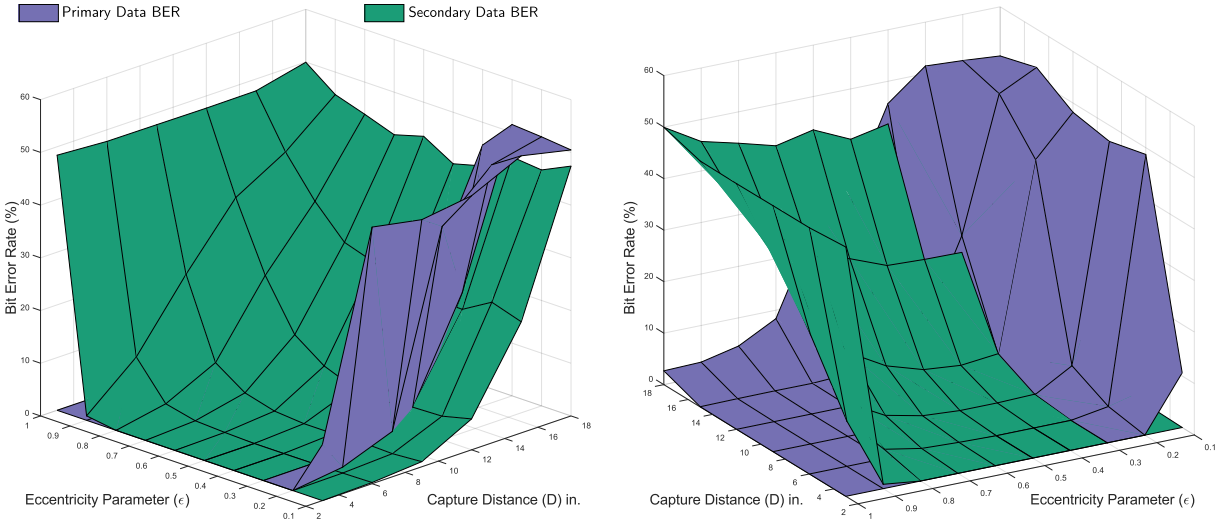
Fig. 8 shows the average (across the device combinations) BER for the primary and secondary data as a function of the eccentricity parameter ϵ, where the results are presented in subfigures (a), (b), and (c) corresponding to capture distances $D = 3, 6,$ and 12 inches. The variance of the secondary data BER is shown as error bars in these plots in order to also depict the extent of variation seen in the results across the devices. Across the range of parameter values, the average BER for the primary data is close to 0 (the average BER for the baseline conventional QR codes is also close to 0 throughout and omitted from the plots because it would completely overlap the primary data BER). Thus, the primary data can be robustly recovered for these choice of parameters and capture distances. For the secondary data, the BERs exhibit expected trends desirable for proximal privacy: BERs rise with increase in the eccentricity parameter ϵ and with the capture distance D. Choices of $\epsilon = 0.5$ and 0.6 allow recovery of the secondary data with fairly low error rates at 6 and 9 in. capture distances. For $\epsilon = 0.9$, the secondary data BER is noticeably higher, even at a close capture distance of $D = 6$ inches. On the other hand, the variance of the secondary data BER is low at 6 in. for all values of ϵ, with the exception of $\epsilon = 0.9$, and increases with the capture distance.

In addition to the BER, the DSR was also evaluated in the experiments both for the DMQR codes and conventional QR codes. Figure 9 shows a plot of the DSRs for primary and secondary data for DMQR codes and for the baseline conventional QR codes (with bars indicating standard deviation). For the DMQR codes, the same values of the eccentricity parameter ϵ and capture distances D that were used for the BER characterization are represented in these results, whereas for the conventional QR codes, additionally, results from capture distances of $D = 18, 24, 30,$ and 36 inches have been included because their performance did not show any degradation over distances under 12 inches. The results validate two important points: the secondary message exhibits the desired proximal privacy property and the choice of the eccentricity parameter ϵ allows a control of the degree of proximal privacy. For example, a choice of the eccentricity parameter $\epsilon = 0.7$ limits reliable

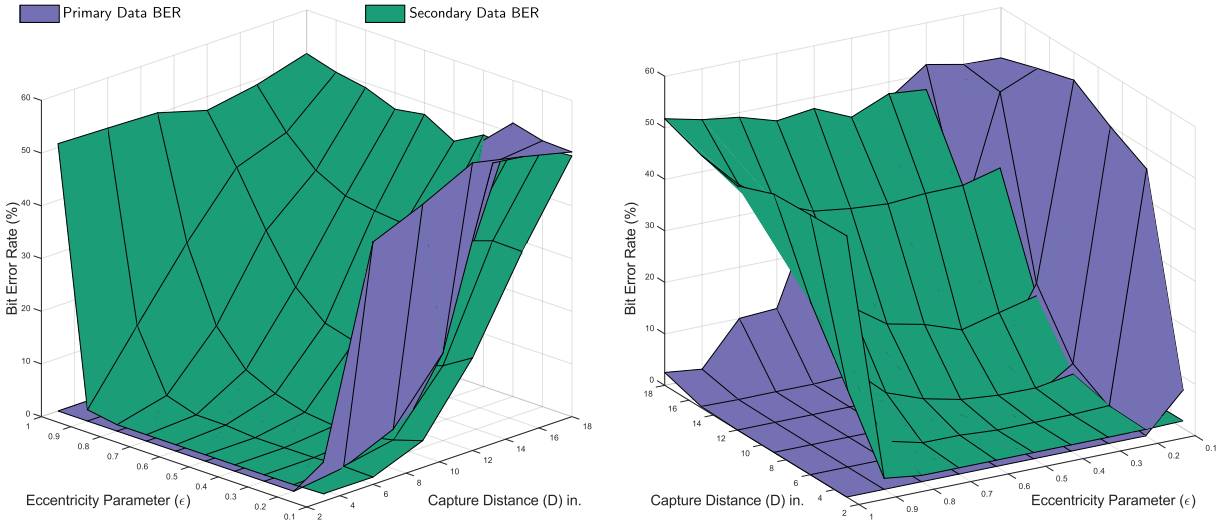
[4]Experiments on actual devices validated the finding from the simulations that $M = 8$-ary modulation did not work effectively.



(a) $M = 2$

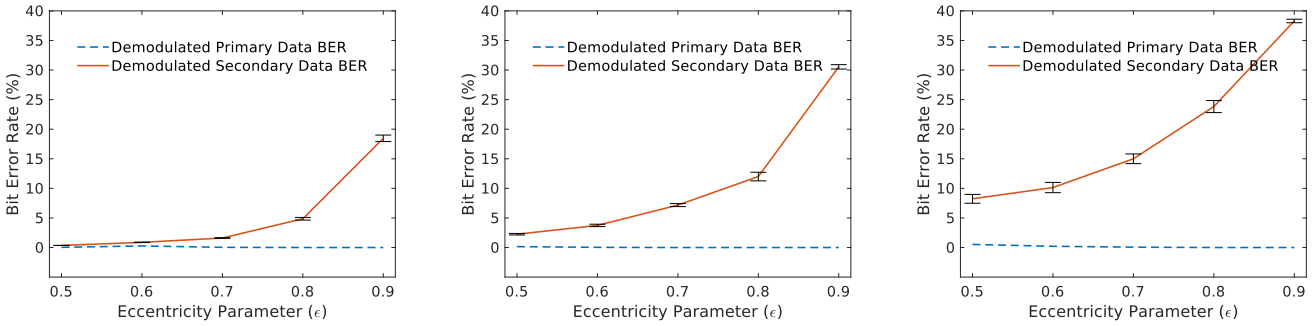


(b) $M = 4$



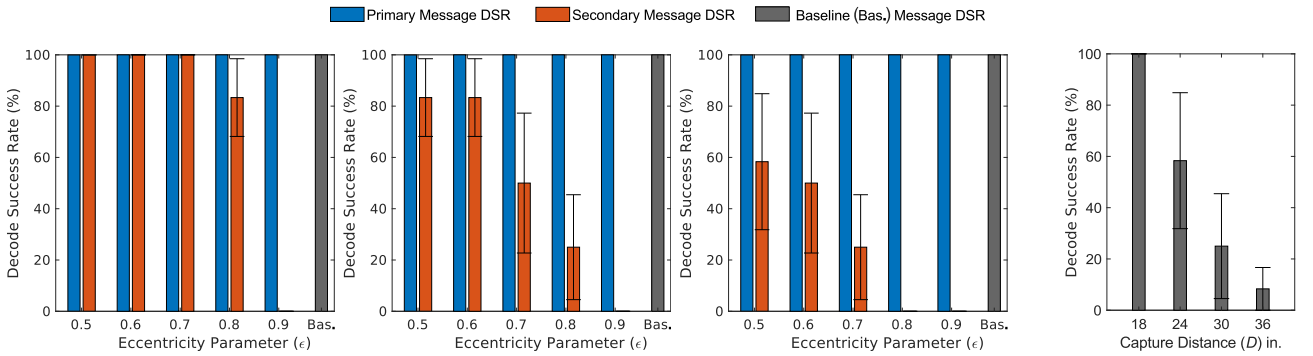
(c) $M = 8$

Fig. 7. Bit error rate (BER) as a function of the ellipse eccentricity parameter ϵ and capture distance D for the DMQR code simulations, for (a) $M = 2$, (b) $M = 4$ and (c) $M = 8$. For clarity, two views per 3D plot are presented. The surfaces show the trade-off between the primary and secondary data BER; as the value of ϵ increases, from 0, the secondary data BER increases while the primary data BER decreases. However, the plots also demonstrate that it is possible to achieve low BER for both primary and secondary data for proximal distances with appropriate choice of ϵ.



(a) $D = 6$ inches (b) $D = 9$ inches (c) $D = 12$ inches

Fig. 8. Average percentage bit error rate (BER) of dual modulated QR codes as a function of the ellipse eccentricity parameter ϵ at three different capture distances.



(a) $D = 6$ inches (b) $D = 9$ inches (c) $D = 12$ inches (d) Baseline at Large Distances

Fig. 9. Average percentage decoding success rate (DSR) for traditional QR codes and for the proposed dual modulated QR codes as a function of the eccentricity parameter ϵ and the capture distance D.

recovery of the secondary message in the DMQR codes to a distance of 6 inches. On the other hand, the primary message is decodable from large capture distances and the conventional QR code can be reliably decoded from images captured at 18 inches and in some cases even at 3 feet away. This robustness becomes a disadvantage when one wishes to keep sensitive data private from cameras further than one's immediate proximity.

IV. Discussion

Our discussion and presentation in this paper was motivated by a specific application scenario for DMQR codes and, for concrete description, we presented a particular design and experimental implementation. From a security perspective, it is also helpful to examine the threat model that the proposed design addresses to better understand its strengths, weaknesses, and potential enhancements. Additionally, alternative design/implementation choices and application scenarios are also worthy of consideration in future work. We address both these aspects in the discussion in this section.

A. Security: Assumptions, Limitations, Enhancements

In the motivating application for DMQR codes described in the introduction, the user of the DMQR codes relies on the proximal privacy to communicate private information securely in a public setting. From a security perspective, others present in the public setting are potential adversaries that seek to discover the private information being communicated. The proximal privacy of DMQR codes relies on the fact that the interaction of the primary user is happening within their personal space and the adversaries are assumed to be at a further distance. The most common adversarial threats in public places are other smartphone users and surveillance cameras. Because, cameras in contemporaneous consumer devices offer similar resolutions, our experiments are well matched with the former setting. As we showed in Section III, at distances greater than or equal to 18 inches (which is considered the boundary of the primary user's intimate space [13, pp. 117]), casual smartphone users would encounter error rates for secondary data decoding close to the 50% random guessing threshold at which there is no leakage of the secondary message information. Surveillance cameras, which are mounted on ceilings, would not only have a longer capture distance, but also significant geometric distortions due to the non fronto-parallel geometry and the wide field of view that such systems are designed to cover. Thus DMQR codes address the most common threats; they are, however, not immune against determined adversaries using specialized equipment. The relative distances of the primary and the adversarial users with respect to the screen displaying the DMQR codes characterizes the proximal privacy and security of DMQR codes. For instance, when the distance from the smartphone displaying the DMQR code to the primary user camera is at 6 inches and the adversarial user is at the 20 feet distance typical for distant vision acuity tests, the

adversarial user needs to overcome a $40\times$ resolution penalty to attain parity with the primary user, not accounting for geometry changes, which are also likely to adversely impact the adversary. Cameras with optical zoom capability can allow the adversary to overcome the distance and geometry disadvantage, particularly when used with a tripod for stabilization. However, the adversary is not entirely unconstrained in using such equipment because it will be clearly apparent to the primary user unless distances are significantly larger than 20 feet, and also likely to attract attention of others in public places (and also recorded on surveillance cameras). As we discuss next, use of DMQR codes with appropriate protocols can offer security enhancements.

The experiments showed that DMQR codes provide significantly better proximal privacy for secondary data compared with classic QR codes. Importantly, by adjusting the eccentricity parameter ϵ, the capture distances at which the secondary message is recoverable can be adjusted, allowing the proximal privacy to be adapted to specific application needs. For sensitive applications, where it is desirable to really minimize the possibility of the secondary data being compromised, one can also use a protocol where the displayed DMQR code starts with the eccentricity parameter $\epsilon = 1$, which is then decreased progressively just until the secondary data is decoded by the target capture camera. In this setting, the primary message channel provides a validated communication interface with robust synchronization and localization in the context of which protection of the secondary message is ensured subject to the constraint of completing the desired transfer. The proximal privacy of the secondary message can also be extended to the primary message by protocols, which use DMQR codes. Fig. 10 illustrates one such protocol, where the secondary message is used as a key/passphrase to encrypt the primary data before the embedding in a DMQR code and, after a successful decoding, the recovered secondary message is used to decrypt the primary message. Thus, if the secondary message is not correctly decoded, the primary message cannot be recovered from its encrypted version.

B. Design/Implementation Choices and Application Scenarios

DMQR codes take advantage of the high pixel density in modern smartphone displays, which allow the orientation modulation in the ellipses to be faithfully rendered. Yet, DMQR codes are flexible and need not be restricted to a particular shape/pattern for the intensity modulation nor limited to specific type of orientations for the orientation modulation. In general, the patterns used to embed the primary data via intensity modulation need to cover enough area to make apparent the difference between empty and modules with patterns while, for the secondary data embedding, the orientation of the patterns has to be significantly different such that they can be distinguished from each other.

ZXing was chosen as the decoder in our experiments because it was easy to integrate with the other elements in the DMQR decoder. The compatibility of the primary message embedding in DMQR codes with traditional QR codes also allows

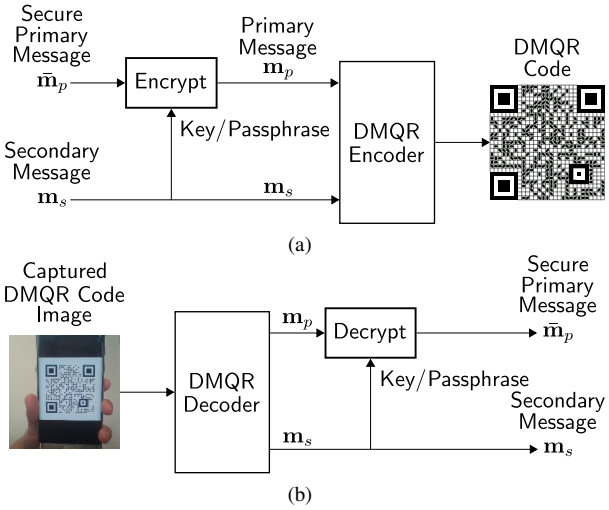


Fig. 10. Protocol for extending the proximal privacy feature of the secondary data to the primary data.

alternative decoders to be used. In fact, based on limited tests, we identified that the decoding performance of ZXing is worse than that of the native QR code decoders in iOS [14], [15] and Android [16], [17] or other open source barcode decoders, e.g. ZBar [18], [19]. The emerging support for QR code localization and decoding in both native camera applications and in the camera application programming interfaces (APIs) for common smartphone operating systems [20]–[22] also makes it easier to realize the overall system proposed in this paper for secure communication of private data in public settings using DMQR codes. We also note that many sites already require smartphone/tablet based two-factor authentication [23], [24] for logging on. In these settings, the authentication code from the two factor authentication app on the smartphone/tablet can also be conveniently incorporated within our proposed DMQR code improving convenience and user experience. Similarly, conventional QR codes are often used for device paring/Wi-Fi connectivity, for instance, for allowing a laptop or tablet device to connect with a mobile hot-spot hosted on a smartphone's data connection. In these settings, the proximal privacy and security provided by DMQR codes also provides improved security over the conventional barcodes.

V. Related Work

Work related to the proposed DMQR codes exists in three contexts: (a) alternative methods for embedding multiple messages in (monochrome) QR codes, (b) orientation modulation for data embedding, and (c) other approaches for achieving proximal privacy. In this section, we discuss relevant prior work in each of these three categories.

Other approaches have also been explored to allow multiple messages to be communicated via printed monochrome barcodes. In [25], a construction using two physical layers with a small gap between them is proposed that provides two alternative decoded messages when the QR code image is captured from two different viewing directions (left/right). The approach requires a specialized construction, where the upper layer is either transparent or black or white at each module

and the upper and lower layers are co-optimized taking into account the standard QR code error correction approach to provide the two decodings. In [26] a two-level barcode is proposed, where the black modules in a conventional QR code are replaced with alternative texture patterns that are used both for authentication of the document bearing the printed QR code and for communicating data privately. Given the focus on printed document authentication and validation, the approach in [26] uses fine texture patterns that cannot be readily replicated and consequently requires both a high resolution print and a high resolution document scanner based capture of the printed QR code for validation and data recovery. The approach cannot be used with smartphone displays and webcam capture, where the resolution is significantly lower for both the smartphone display and webcam; and additionally, there is also some perspective distortion introduced by the variations in capture geometry from an ideal fronto-parallel geometry. Our tests with the codes provided in [26] confirmed this hypothesis. The proposed DMQR codes on the other hand use oriented elliptical dots for the secondary data, which are easier to resolve at the lower resolutions of interest in our applications and also significantly simplify detection based on image moments, instead of requiring the more expensive correlation detector used in [26]. For printed barcodes, interesting future directions of work would be to explore combinations of the proposed orientation modulation approach with those in [25] and [26] by using multiple layers and/or adding texture to the elliptical dots.

Orientation modulation has also been previously used in several applications involving printed documents. Its use was first proposed for watermarking of documents printed with conventional clustered dot halftones in [27] and the methodology was subsequently adapted to high density printed color barcodes [7] and image barcodes [28]. The application setting in this paper focuses on mobile display applications, which introduces significant differences from the prior work. Specifically, instead of a high resolution scanner used in the prior works, in the present work, images for decoding the barcodes are captured with a camera and therefore have lower resolution and are subject to perspective geometric distortion in the capture process. The approach proposed here overcomes the challenges by leveraging the robust synchronization and perspective geometry correction built into the design of QR codes and by using explicitly defined elliptical dots in the orientation modulation instead of the methodology used in [7], [27], [28], where an asymmetry was introduced in the halftone threshold function to accomplish the orientation modulation. Alternative forms of orientation modulation have also been proposed for dispersed dot halftone printing in monochrome [29] and color settings [30], [31]. The operational rates that we achieve with the 4-ary DMQR codes used in our experiments (simulations and with actual devices) are adequate for our target application, which does not require particularly high capacity. While beyond the immediate scope and focus of the current work, for other application settings, it may be of interest to analyze the theoretical capacity achievable with DMQR codes by using a framework analogous to the one that has been used for the related orthogonal halftone orientation modulation channel [32].

Alternative technologies can also be used for proximal communication. In particular, near field communication (NFC) [33] technology is specifically designed for close range communication between compatible devices using radio waves. However, NFC technology is still not standard in all mobile devices (laptops, smartphones, etc.) and represents an added expense compared with QR (and DMQR) codes that require only displays and cameras, which are already ubiquitous on almost all mobile devices. Furthermore, NFC users can be unaware of which devices are in the NFC range, whereas for QR codes, the user is aware of the"field of view" within which the code can be seen [34]. Due to these advantages, QR code based systems are already being extensively used in several point of payment systems [35]–[37]. The proposed DMQR codes further enhance the functionality of QR codes, providing not only the ability to convey additional data but also proximal privacy that, as already noted, is particularly advantageous when communicating private information in public settings. We note that Denso, the company that created QR codes, also provides for a "secret-function-equipped QR code (SQRC)" that can be used for secure communication of private data [38], [39]. However, unlike DMQR codes, SQRC codes require specialized hardware for decoding.

VI. Conclusion

DMQR codes introduced in this paper enable "proximal privacy" for communication of private data in public settings using smartphone displays. In addition to a primary message, whose encoding can be completely compatible with conventional monochrome QR barcode decoders, DMQR codes also carry a secondary message that is only decodable from images of the DMQR codes captured from a close distance. The proximal privacy of DMQR codes can be adapted to meet application requirements through appropriate choice of the eccentricity parameter ϵ that controls the orientation modulation for the secondary data. Experiments with DMQR codes displayed on smartphone displays and captured using webcams demonstrate that the DMQR codes achieve their objectives. The secondary data exhibits low bit error rates for capture distances under 9 to 6 inches (depending on ϵ) which can be readily overcome using error correction, whereas for larger capture distances over 12 inches, the secondary message is not recoverable from captured images. Using appropriate protocol constructions, we also highlight how proximal privacy can be extended to the primary message.

VII. Acknowledgments

We thank the anonymous reviewers for their comments and suggestions, which have significantly improved this article. We are also grateful to the Center for Integrated Research Computing, University of Rochester, for providing access to computational resources for this work.

Appendix A
Implementation Details

For the DMQR codes, the URL https://labsites.rochester.edu/gsharma/ was used as the primary message $\mathbf{m}_p$

and the secure passwords `g5r[GRw}^Gu*kk^c:Q)s`, `g5r[GRw}^Gu*kk^c:Q)s/:Sys7q?cgvA5rCM:!s%`, and `wUXk__}K7ygV!(5^b\.N8y7,wpK-aT}E-RnQ3>k8G}XtfT4yrbKJdGtu(g5r)` served as the secondary messages $\mathbf{m}_s$ for $M = 2, 4$ and 8, respectively. The intensity modulated $\mathbf{c}_p$ yields an encoded $\tilde{I}_0(\chi, \zeta)$ with 29 modules in the horizontal and vertical directions; thus, $\tilde{I}_0(\chi, \zeta)$ has $29 \times 29 = 841$ modules from which $N_B = 330$ (number of modules with ellipses). Error correction for the secondary data utilized a rate $1/2$ maximum free distance binary convolutional code from [40, Table 10.1, pp. 160], which was first presented in [41]. The specific code used the generator polynomials specified in octal by $[457, 755]$, which has a constraint length of 9 bits (corresponding to $2^{9-1} = 256$ states in the decoder trellis) and a free distance of 12 bits.

For displaying the barcodes, we used the full width resolution of a smartphone in portrait orientation. We considered the resolution of an iPhone 7 (1334×750) because current smartphones have similar or higher pixel count. Modules of size 20×20 pixels ($\Delta = 20$) are utilized and thus the DMQR code image $I_0(\chi, \zeta)$ is 580×580 pixels ($S = 580$), given that $29 \times 20 = 580$. Furthermore, to facilitate synchronization (as in [42]), we added a white uniform border into $I_0(\chi, \zeta)$ resulting in a 15% increase in each dimension such that $I_0(\chi, \zeta)$ ends up with a size of 754×754 pixels. $I_0(\chi, \zeta)$ is 4 pixels larger than the width resolution of the iPhone 7 (750 pixels) because we wanted to have an integer value for Δ to avoid rendering artifacts. We also assume that the $I_0(\chi, \zeta)$ is created in the smartphone and thus $I_0(\chi, \zeta)$ is stored as a PNG file.

Appendix B
DMQR Codes For Print

Given our primary motivating application of securely communicating private data in public settings, the experiments in Section III used DMQR codes displayed on smartphone and captured using webcams. DMQR codes can, however, also be used in print.

We conducted limited experiments to demonstrate and validate DMQR codes in print. Specifically, we printed (on a Xerox WorkCentre 7845 printer) DMQR codes generated with eccentricity parameters $\epsilon = 0.5, 0.6, 0.7$ in three different sizes, 1.2×1.2 cm, 1.5×1.5 cm and 2×2 cm. The printed codes were then captured from two different distances (3 and 6 in) with a smartphone (Xiaomi Mi 8 with 2268×4032 pixel native camera resolution, stored at JPEG quality factor 90) and the BERs and DSRs for primary and secondary data/messages were evaluated following the methodology already outlined in Section III. Other parameters, were identical to those for the experiments in Section III. Table II summarizes the BER and DSR performance metrics for the printed DMQR codes across the different printed sizes and for the three choices of the eccentricity parameter ϵ.

The results indicate that (in our limited experiments) no errors were seen for the primary data and the primary message was decoded with a 100% success rate. For the secondary data, for barcodes generated with the eccentricity parameter $\epsilon = 0.5$ the bit error rates were low and the message was recoverable using the convolutional error correction codes, except for the smallest 1.2×1.2 cm printed version captured from 6 inches. In contrast, for the eccentricity parameter $\epsilon = 0.7$ the bit error rates for secondary data were high and except for the largest 2×2 cm printed version captured from the close 3 inches distance, the secondary message was successfully decoded. On the other hand, the printed DMQR codes with $\epsilon = 0.6$ exhibited low bit error rates at 3 inches and the secondary message was recovered while at 6 inches only the secondary message of the largest DMQR code was decoded correctly. The experiments confirm that our DMQR codes can also be successfully used in print. For print applications, DMQR codes provide the clear benefit of increasing the data rate (with appropriately chosen parameters). While they also provide proximal privacy, compelling applications for this proximal privacy remain to be determined for printed DMQR codes.

References

[1] J. Leggate, "Big brother watching you? these are the world's most heavily surveilled cities," Aug. 2019, accessed Sept. 27, 2019. [Online]. Available: https://www.foxbusiness.com/technology/cities-under-most-surveillance-world

[2] Y. Xu, J. Heinly, A. M. White, F. Monrose, and J.-M. Frahm, "Seeing double: Reconstructing obscured typed input from repeated compromising reflections," in *Proc. ACM SIGSAC Conf. Comput. & Commun. Secur.*, 2013, pp. 1063–1074.

[3] D. Shukla and V. V. Phoha, "Stealing passwords by observing hands movement," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 12, pp. 3086–3101, Dec. 2019.

[4] D. Balzarotti, M. Cova, and G. Vigna, "ClearShot: Eavesdropping on keyboard input from video," in *IEEE Symp. Secur. and Privacy*, May 2008, pp. 170–183.

[5] J. Gilmore, "The FRONTLINE interviews: Glenn Greenwald," Feb. 2014, accessed Sept. 27, 2019. [Online]. Available: https://www.pbs.org/wgbh/pages/frontline/government-elections-politics/united-states-of-secrets/the-frontline-interview-glenn-greenwald/

[6] ISO/IEC, "ISO/IEC 18004:2015: information technology — automatic identification and data capture techniques — QR code bar code symbology specification," Feb. 2015, accessed May 18, 2020. [Online]. Available: https://www.iso.org/standard/62021.html

[7] O. Bulan and G. Sharma, "High capacity color barcodes: Per channel data encoding via orientation modulation in elliptical dot arrays," *IEEE Trans. Image Proc.*, vol. 20, no. 5, pp. 1337–1350, May 2011.

[8] "Secure password generator," accessed Oct. 17, 2019. [Online]. Available: https://passwordsgenerator.net/

[9] S. Wilson, *Digital Modulation and Coding*. Upper Saddle River, NJ, USA: Prentice Hall, 1996.

[10] "ZXing C++ port," accessed Oct. 15, 2019. [Online]. Available: https://github.com/glassechidna/zxing-cpp

[11] J. E. Farrell, P. B. Catrysse, and B. A. Wandell, "Digital camera simulation," *Appl. Opt.*, vol. 51, no. 4, pp. A80–A90, Feb. 2012.

[12] "ISETCam," accessed Oct. 19, 2019. [Online]. Available: https://github.com/ISET/isetcam

[13] E. Hall, *The Hidden Dimension*. New York, NY: Doubleday, 1966.

[14] F. Tepper, "The iPhone's camera app can now read QR codes." https://techcrunch.com/2017/06/05/the-iphones-camera-app-can-now-read-qr-codes/, 2017.

[15] R. Holmes, "Why the "hidden" update in Apple's latest iOS changes the rules of marketing," https://www.forbes.com/sites/ryanholmes/2017/10/30/why-the-hidden-update-in-apples-latest-ios-changes-the-rules-of-marketing/#354c27a73d0a, 2017.

[16] N. Arce, "Motorola updates camera and gallery apps, now allows QR code scanning and saving albums to MicroSD," http://www.techtimes.com/articles/80661/20150831, 2015.

[17] "Scanning a QR Code from Samsung phone," accessed Oct. 18, 2019. [Online]. Available: https://www.samsung.com/au/support/mobile-devices/samsung-qr-code-scanner/

TABLE II
PERFORMANCE STATISTICS OF PRINTED DMQR CODES

Eccentricity Parameter ϵ	DMQR Code Size (cm)	Capture Distance (inches)	Primary Data BER (%)	Secondary Data BER (%)	Primary Message DSR (%)	Secondary Message DSR (%)
0.5	1.2×1.2	3	0	1.97	100	100
		6	0	6.36	100	0
	1.5×1.5	3	0	0.76	100	100
		6	0	4.39	100	100
	2×2	3	0	0.15	100	100
		6	0	1.67	100	100
0.6	1.2×1.2	3	0	3.94	100	100
		6	0	13.94	100	0
	1.5×1.5	3	0	0.30	100	100
		6	0	7.88	100	0
	2×2	3	0	0.15	100	100
		6	0	0.91	100	100
0.7	1.2×1.2	3	0	14.70	100	0
		6	0	20.61	100	0
	1.5×1.5	3	0	9.70	100	0
		6	0	11.21	100	0
	2×2	3	0	2.58	100	100
		6	0	4.09	100	0

[18] "ZBar bar code reader," accessed Oct. 18, 2019. [Online]. Available: http://zbar.sourceforge.net/

[19] "ZBar bar code reader," accessed Oct. 18, 2019. [Online]. Available: https://github.com/ZBar/ZBar

[20] Google, "Barcode API overview," accessed Oct. 19, 2019. [Online]. Available: https://developers.google.com/vision/android/barcodes-overview

[21] Apple, "AVMetadataMachineReadableCodeObject - AVFoundation — Apple developer documentation," accessed Oct. 19, 2019. [Online]. Available: https://developer.apple.com/documentation/avfoundation/avmetadatamachinereadablecodeobject

[22] Apple, "CIQRCodeFeature - Core Image — Apple developer documentation," accessed Oct. 19, 2019. [Online]. Available: https://developer.apple.com/documentation/coreimage/ciqrcodefeature

[23] Duo, "Guide to two-factor authentication," accessed Nov. 1, 2019. [Online]. Available: https://guide.duo.com/

[24] F. Aloul, S. Zahidi, and W. El-Hajj, "Two factor authentication using mobile phones," in *IEEE/ACS Intl. Conf. Comp. Sys. and Applic.*, May 2009, pp. 641–644.

[25] T. Yuan, Y. Wang, K. Xu, R. R. Martin, and S. Hu, "Two-layer QR codes," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4413–4428, Sept. 2019.

[26] I. Tkachenko, W. Puech, C. Destruel, O. Strauss, J. Gaudin, and C. Guichard, "Two-level QR code for private message sharing and document authentication," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 3, pp. 571–583, Mar. 2016.

[27] O. Bulan, G. Sharma, and V. Monga, "Orientation modulation for data hiding in clustered-dot halftone prints," *IEEE Trans. Image Proc.*, vol. 19, no. 8, pp. 2070–2084, Aug. 2010.

[28] O. Bulan, G. Sharma, and B. Oztan, "High capacity image barcodes using color separability," in *Proc. SPIE: Color Imaging XVI: Displaying, Processing, Hardcopy, and Applications*, vol. 7866, Jan. 2011, pp. 7866–N,1–9.

[29] J. Guo, C. Su, Y. Liu, H. Lee, and J. Lee, "Oriented modulation for watermarking in direct binary search halftone images," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4117–4127, Sept. 2012.

[30] V. Kitanovski and M. Pedersen, "Orientation modulation for data hiding in chrominance channels of direct binary search halftone prints," *J. Imaging Sci. and Tech.*, vol. 60, no. 5, pp. 50 407–1, 2016.

[31] V. Kitanovski and M. Pedersen, "Detection of orientation-modulation embedded data in color printed natural images," *J. Imaging*, vol. 4, no. 4, p. 56, 2018.

[32] O. Bulan, V. Monga, and G. Sharma, "Capacity analysis for orthogonal halftone orientation modulation channels," *IEEE Trans. Image Proc.*, vol. 21, no. 1, pp. 405–411, Jan. 2012.

[33] ISO/IEC, "ISO/IEC 18092:2013 Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)," 2013. [Online]. Available: http://standards.iso.org/ittf/PubliclyAvailableStandards/c056692_ISO_IEC_18092_2013.zip

[34] G. Sharma, "Image-based data interfaces revisited: Barcodes and watermarks for the mobile and digital worlds," in *8th Intl. Conf. on Comm. Sys. and Networks*, Bangalore, India, Jan. 2016, pp. 1–6.

[35] Walmart, "Walmart Pay," Aug. 2019, accessed Aug. 3, 2019. [Online]. Available: https://www.walmart.com/cp/walmart-pay/3205993

[36] WeChat, "QR code payment," Aug. 2019, accessed Aug. 3, 2019. [Online]. Available: https://pay.weixin.qq.com/wechatpay_guide/qrcode_payment.shtml

[37] Alipay, "Alipay Barcode Payment: Introduction," Aug. 2019, accessed Aug. 3, 2019. [Online]. Available: https://intl.alipay.com/doc/barcode

[38] "SQRC®," accessed Oct. 18, 2019. [Online]. Available: https://www.denso-wave.com/en/system/qr/product/sqrc.html

[39] "What is SQRC (secret-function-equipped QR code)?" accessed Oct. 18, 2019. [Online]. Available: http://denso-adc.com/learning-center/what-is-sqrc

[40] J. M. Cioffi, "Chapter 10: System design with codes," in *Class Reader for Stanford University EE 379B - Digital Communication II: Coding*, February, accessed May 26, 2020. [Online]. Available: https://ee.stanford.edu/~cioffi/doc/book/chap10.pdf

[41] R. Johannesson and K. Zigangirov, *Fundamentals of convolutional coding*. Institute of Electrical and Electronics Engineers, 1999.

[42] K. Dinesh, J. Lu, S. Dhoro, and G. Sharma, "Channel-wise barcodes for color display applications," *J. Electronic Imaging*, no. 3, pp. 033 021–1 – 18, May/Jun. 2019.