Feature Article

# Quashing Quishing Attacks Using Self-Authenticating Dual-Modulated QR Codes

Irving Barron[a], *University of Rochester, Rochester, NY, 14627-0231, USA*
Gaurav Sharma[a], *University of Rochester, Rochester, NY, 14627-0231, USA*

*Abstract—Quishing, i.e., QR code based phishing, is an increasingly prevalent security attack used to deceive individuals into providing sensitive private information. To combat quishing, we propose self-authenticating dual-modulated QR (SDMQR) codes that transparently retrofit existing QR codes with cryptographic self-authenticating capability leveraging the recently developed dual-modulated QR (DMQR) codes.* ***Index Terms: QR codes, quishing, authentication, digital signatures***

Quick response (QR) codes are quite prevalent nowadays and they are used for a variety of applications, most commonly to conveniently connect to online resources on user's smartphone devices. Unfortunately, bad actors are also capitalizing on the popularity of QR codes and exploiting them to commit crimes. QR-code-based phishing, also known as quishing[b], is increasingly prevalent[c,d]. In a common quishing ploy, users are tricked into providing their credentials by connecting them, via a QR code, to a fake website masquerading as the user's bank website. Bad actors also particularly target email users with quishing attacks because these attacks can circumvent email security systems[e]. Furthermore, quishing attacks are not exclusive to the digital world. As reported by the Federal Bureau of Investigation (FBI) and the Federal Trade Commission (FTC)[f,g], bad actors are also tampering with legitimate physical QR codes by printing spurious QR codes to cover the genuine ones (for instance, those used to collect parking payments[h]). Quishing is also not the only QR-code-based attack being used in practice. Bad actors are also using QR codes to install malware, as reported by the Better Business Bureau (BBB)[i]. Additionally, there is concern regarding QR codes carrying information to execute SQL or HTML injection attacks[j].

To defend against bad actors and their QR-code-based attacks, users have several alternatives. First, users can examine the data within a QR code (e.g., inspect a URL before opening it), a practice recommended by the FBI, FTC, and BBB. Unfortunately, under the pressure of urgency (commonly induced by scammers) and given the limited real-estate on smartphone screens, it is challenging for users to properly scrutinize the data carried by a QR code. Additionally, in the case of quishing, scammers use link redirection services (e.g., link shortening and online QR code generator websites) to further obfuscate the malicious URL, making the scrutiny process even more challenging. The FBI offers an alternative recommendation[k] that consumers use antivirus software equipped with a QR code reader that can verify the safety of a link before opening it. This is, however, rarely done as most users prefer to access QR codes

XXXX-XXX © 2024 IEEE
Digital Object Identifier 10.1109/XXX.0000.0000000

[a]Both authors contributed equally to this work. The authorship order was determined alphabetically.

[b]https://www.techtarget.com/searchsecurity/feature/Quishing-on-the-rise-How-to-prevent-QR-code-phishing

[c]https://abnormalsecurity.com/blog/data-shows-c-suite-receives-42x-more-qr-code-attacks

[d]https://keepnetlabs.com/blog/understanding-quishing

[e]https://www.inky.com/en/blog/fresh-phish-malicious-qr-codes-are-quickly-retrieving-employee-credentials

[f]https://www.ic3.gov/PSA/2022/PSA220118

[g]https://consumer.ftc.gov/consumer-alerts/2023/12/scammers-hide-harmful-links-qr-codes-steal-your-information

[h]See footnote g.

[i]https://www.bbb.org/article/news-releases/27342-bbb-scam-alert-fraudulent-qr-codes-continue-to-be-used-in-a-variety-of-scams

[j]https://www.fbi.gov/contact-us/field-offices/elpaso/news/fbi-tech-tuesday-building-a-digital-defense-against-qr-code-scams

[k]See footnote j.

directly using their smartphone cameras. As another alternative, users may consider specialized barcodes. For instance, Denso, the company behind the invention of QR codes, offers a "secret-function-equipped QR code (SQRC)" for the secure transmission of private data[l,m]. This technology limits access to the private information contained within the SQRC code to authorized devices only by using a cryptographic key. However, SQRC codes do not ensure the security of the encoded data and fail to integrate seamlessly with existing QR code workflows. As an extreme solution, users can completely avoid using QR codes. By avoiding QR codes, users sacrifice the convenience that these barcodes offer and are faced with the challenge of finding alternative methods for tasks where QR codes are commonly used. Because QR-code-based applications are widely deployed and used on many diverse devices, a whole-scale system re-design to add security features is also extremely challenging as it would disrupt existing workflows and applications for a large numbers of users.

To address the aforementioned challenges, **here, we propose and demonstrate a method for transparently retrofitting QR codes in existing applications with self-authenticating security capability** by leveraging our recently developed dual-modulated QR (DMQR) codes [1], [2]. Using the proposed method, QR codes already in use in various applications can be replaced by DMQR codes **that establish their own veracity through cryptography, without impacting the already existing functionality**. We use the moniker self-authenticating dual-modulated QR (SDMQR) codes to refer to the particular instance of DMQR codes obtained using the proposed method. Like DMQR codes, SDMQR codes carry a primary and a secondary message. The primary and secondary messages in SDMQR codes, however, are related. Specifically, the secondary message is a (digital) signature of the primary message where this signature is created by using the primary message and a cryptographic key. After successfully reading an SDMQR code, the recovered primary message and a cryptographic key can be used to verify the recovered signature establishing the authenticity of the primary message in the SDMQR code.

SDMQR codes are particularly attractive to fight quishing attacks. Figure 1 depicts an example that demonstrates how SDMQR codes can be used to prevent quishing. In this setting, the primary message is a URL (universal resource locator) link to a bank's login webpage, and the secondary message is the corresponding signature computed using the signatory's private key. Upon reading the SDMQR code, the primary and secondary messages can be recovered and, using the available public key for the signatory, the signature can be checked to determine the authenticity of the URL. When the signature verification is successful, as depicted in Fig. 1 (a), the SDMQR code reader shows this to the user with a green check mark, indicating it is safe to follow the associated link. On the other hand, if the signature verification fails, as shown in Fig. 1 (b), the SDMQR code reader puts a red cross, warning the user that the content of the barcode has not been authenticated and the link may be unsafe to follow. In this setting, we also propose a cryptographic protocol for using a centralized signatory for the SDMQR codes, where individual service providers pre-register their URLs and obtain signatures for use in the SDMQR codes. The proposed protocol is advantageous because only a single public key (or a few public keys) then needs to be available in the smartphone camera app for directly verifying the authenticity of URLs when the camera app reads

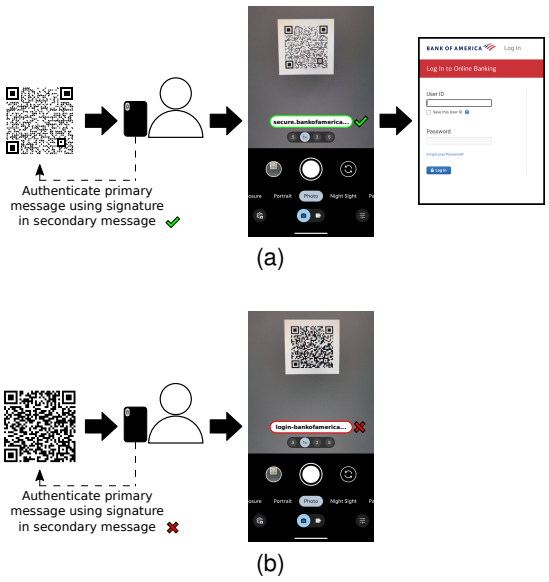



FIGURE 1: Illustration demonstrating how SDMQR codes can prevent quishing: (a) for a genuine SDMQR code the primary message is authenticated via the signature embedded in the secondary message and the user gets a green check mark indicating it is safe to proceed to the linked site, and (b) for other QR codes, the authentication fails and the user is warned by a red cross mark.

[l] https://www.denso-wave.com/en/system/qr/product/sqrc.html

[m] http://denso-adc.com/learning-center/what-is-sqrc

SDMQR codes.

## Self-Authenticating Dual-Modulated QR Codes

The proposed methodology for SDMQR codes is illustrated in Figure 2, where, for concrete description, we focus on the specific embodiment of SDMQR codes to fight quishing attacks. With minor modifications, the description extends straightforwardly to alternative realizations, some of which are described later. Our description assumes the availability of public key cryptography infrastructure [3, Chap. 8], under which, the signatory attesting to the authenticity of the SDMQR codes has created a private (secret) key **d** and a corresponding public key **Q**. Using the private key **d**, the signatory can compute a digital signature **s** for an arbitrary message **m** (represented as a sequence of bits). A signature verification algorithm can then use the public key **Q** to determine whether a presented signature $\hat{\mathbf{s}}$ verifies a presented message $\hat{\mathbf{m}}$. Assuming that the cryptosystem is not compromised, a valid verification is obtained only when the signature $\hat{\mathbf{s}}$ is generated from the message $\hat{\mathbf{m}}$ using the private key **d**, establishing the authenticity of the message under the signatory's authority.

The process for creating an SDMQR code is depicted in Fig. 2 (a). The primary message $\mathbf{m}_p$ is (a sequence of bits representing) a URL that connects users to an online resource. The secondary message $\mathbf{m}_s$ is a cryptographic signature of the primary message $\mathbf{m}_p$ computed using a signatory's private key **d**. The messages $\mathbf{m}_p$ and $\mathbf{m}_s$ are then embedded as the primary and secondary messages for a DMQR code by using a DMQR encoder [1], [2], producing an SDMQR code. While readers are referred to DMQR publications [1], [2] for details of DMQR codes, a quick overview is provided by the illustration in Fig. 2 (b) where we show a zoomed-in region highlighting spatial detail of the SDMQR code. The synchronization/alignment patterns and the underlying geometry of the square modules used to carry the data is identical to that of traditional QR codes[n] but, whereas each module is either completely white or completely black in traditional QR codes, in the SDMQR (and in DMQR) codes, each module is either white or carries an elliptical dot of varying orientation. The modules with elliptical dots are darker than those without, and this variation in intensity

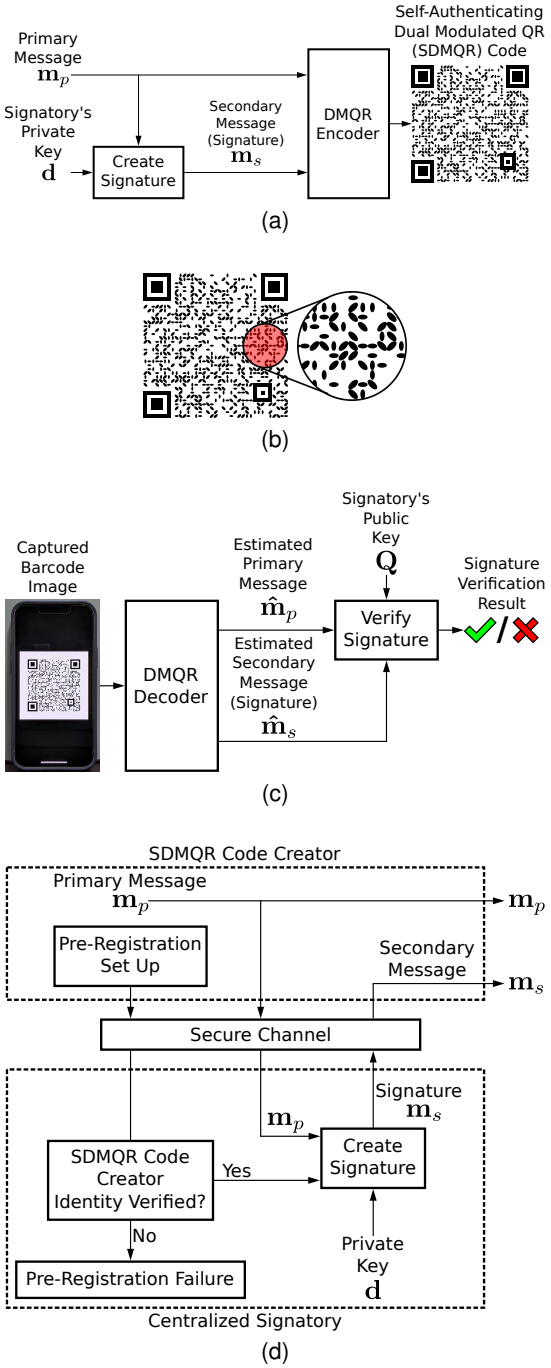



FIGURE 2: Self-authenticating dual-modulated QR (SDMQR) codes: (a) SDMQR code creation, (b) an SDMQR code with spatial detail of the embedding modules highlighted, (c) SDMQR code authentication, and (d) pre-registration with a centralized signatory. See text for additional details.

[n] An excellent video tutorial [4] describes how QR codes are created and highlights QR code security concerns that are effectively addressed by the approach proposed here.

carries the primary message and the variations in orientation of the elliptical dots carry the secondary message. Error correction coding is incorporated for both the primary and secondary message to allow recovery of the messages despite some errors in individual modules. The errors may be inadvertently caused by various sources or may result from "beautification" where a small spatial region of the code is deliberately cannibalized and replaced by a logo or another image. For the primary message, the error correction coding follows the QR code standard [5]. Details of an exemplary implementation of the error correction coding for the secondary message can be found in a prior publication [2], where the performance is also characterized in the presence of beautification. Importantly, the modulation of intensity and the error correction coding for the primary message maintain compatibility and the primary message can therefore be read with standard QR code readers, which can also be verified by reading the SDMQR code in the figure with most modern smartphone cameras.

The authentication process for SDMQR codes is demonstrated in Fig. 2 (c). The captured image of a barcode to be authenticated is input to a DMQR decoder [2]. If the decoding succeeds for both the primary and the secondary message, the decoder returns estimates $\hat{\mathbf{m}}_p$ and $\hat{\mathbf{m}}_s$, of the primary message and the secondary message for further verification, otherwise the code is deemed inauthentic. When decoding succeeds, the public key $\mathbf{Q}$ of the signatory is used in a digital signature verification algorithm [3, Chap. 11] to check whether the recovered secondary message $\hat{\mathbf{m}}_s$ is a signature verifying the recovered primary message $\hat{\mathbf{m}}_p$. If the verification succeeds, the barcode is deemed to be an authentic/valid SDMQR code (indicated by the green check mark in the figure), otherwise it is deemed inauthentic/invalid (indicated by the red cross mark in the figure). Note that when the barcode being read is a traditional QR code, invariably, the DMQR decoder will fail in the process of error correction decoding for estimating the secondary message and when the barcode being read is a DMQR code that is not an SDMQR code, the signature verification will fail. When verification succeeds, users can be assured[o] of the authenticity of the SDMQR code and by choosing an appropriate signatory that only signs valid URLs, users can be assured that they will not fall prey to quishing if they only follow validated links.

Alternative signatories can be chosen for guaranteeing the authenticity of SDMQR codes. An obvious choice is for resource providers to sign their own SDMQR codes. In this setting, the provider of an online resource, such as a bank, signs their own SDMQR codes using the provider's private key. In this setting, the SDMQR code reader must have access to public keys for all possible signatories for whom verification could be required and the signatory's identity is also required at the SDMQR code reader. While this is not infeasible, it does pose a significant burden for the SDMQR code reader. We propose a cryptographic protocol to eliminate this burden, by using a few authorized central signatories, or a single one, and a pre-registration process that is illustrated in Fig. 2 (d). In the pre-registration process, for creating an SDMQR code, an online resource provider generates their primary message $\mathbf{m}_p$ and sends it to the central signatory via a secure channel on which the identity of the resource provider has been verified (using existing public key cryptography infrastructure). The central signatory generates the signature $\mathbf{m}_s$ for authenticating the message $\mathbf{m}_p$ using the signatory's private key $\mathbf{d}$ and sends back the signature $\mathbf{m}_s$ to the online resource provider, who then generates the SDMQR code using $\mathbf{m}_p$ and $\mathbf{m}_s$ as the primary and secondary messages as in Fig. 2 (a). The authenticity of the generated SDMQR code can then be established by using the public key $\mathbf{Q}$ for the signatory. The SDMQR code reader only needs the public keys of the authorized central signatories, which is much easier to handle than the situation described earlier where public keys for each resource provider are required. If there are few possible central signatories, the SDMQR code reader can simply declare a scanned barcode to be authentic if the retrieved signature $\hat{\mathbf{m}}_s$ authenticates the retrieved primary message $\hat{\mathbf{m}}_p$ with any of the central signatories public keys, or a few bits in the secondary message can be used to identify the signatory. The centralized signatory option is particularly attractive because Apple and Google are the two providers of mobile operating systems for the overwhelming majority of smartphone and tablet devices and they also provide native camera apps with built in QR code readers. Thus, with Apple and Google serving as the two central signatories for SDMQR codes and the aforementioned pre-registration process, public keys registered to these companies can be used to conveniently verify SDMQR codes with only the public keys for these two companies required at the SDMQR code reader. Ideally, the SDMQR code reading and verification capability can be incorporated within the native camera barcode readers allowing for seamless authentication within the camera app as illustrated in Fig. 1.

[o]As before, we assume that the cryptosystem is not compromised.

## Experiments and Results

We develop a specific realization of SDMQR codes geared toward countering quishing attacks and report experimental results for two scenarios, one where the SDMQR code is displayed on a screen and the other where it is printed on paper. For our digital signatures, we chose the elliptic-curve-cryptography-based Edwards-curve Digital Signature Algorithm (EdDSA) that is part of the National Institute of Standards and Technology (NIST) Digital Signature Standard (DSS) [6] and provides an advantage over the widely used RSA [7] public key cryptosystem by requiring a shorter signature for a given security strength. Specifically, using the Edwards25519 elliptic curve [8], EdDSA with 256-bit keys and 512-bit signatures achieves security comparable to that offered by RSA-3072 with 3072-bit keys and 3072-bit signatures [9]. The OpenSSL[p] open-source cryptography toolkit was used to generate a private-public key pair ($\mathbf{d}$,$\mathbf{Q}$) and for the signature generation and verification functions. A bank URL was used as the primary message $\mathbf{m}_p$ and the corresponding EdDSA signature was obtained using the private key $\mathbf{d}$. The specific URL, the private and public keys used for the experiments, and the signature are provided in Supplementary Materials, where other parameters specific to the implementation are also summarized.

An SDMQR code was obtained using the process of Fig. 2 (a), where the $\mathbf{m}_p$ and $\mathbf{m}_s$ were used as the primary and secondary messages for the DMQR encoder [2]. The actual resulting SDMQR code is used in the illustrations in Fig. 2 (a) and (b). We note that although $\mathbf{m}_p$ and $\mathbf{m}_s$ are fixed for the tests presented in this section, the results are representative [2] because the QR code generation process includes XORing (exclusive OR-ing) of the data bits with a pseudo-random mask pattern that is designed to randomize individual data carrying modules [5].

For the experiments with SDMQR codes displayed on a screen, a Samsung Galaxy J3 smartphone representative of current entry-level devices was chosen because its relatively low 294 pixels per inch (PPI) screen spatial resolution exemplifies the most challenging application scenario for SDMQR codes [2]. For the experiments with printed SDMQR codes, we utilized a Brother MFC-J1205W printer to produce three different sizes of the same SDMQR code: $1.00 \times 1.00$, $0.75 \times 0.75$ and $0.50 \times 0.50$ inches. For capturing both the displayed-on-screen and printed SDMQR codes, a Samsung Galaxy S6 camera having a resolution of $3984 \times 2988$ pixels was utilized. Images of the SDMQR codes were captured from capture distances of $D = 12, 15$ and 18 inches for the displayed-on-screen version and $D = 9, 12$ inches for the printed version. An approximate fronto-parallel geometry was used for the capture though the geometry was not rigorously controlled and included typical variations that are likely to be encountered in practice. Images from all the captures were stored in JPEG format with a quality factor of 95, the default for the device. Estimates $\hat{\mathbf{m}}_p$ and $\hat{\mathbf{m}}_s$ of the primary and secondary messages, respectively, were obtained using the DMQR code decoder [2]. The signature verification function for EdDSA in the OpenSSL[q] toolkit was then used with the signatory's public key $\mathbf{Q}$ to determine whether the estimated signature $\hat{\mathbf{m}}_s$ verified the authenticity of the estimated primary message $\hat{\mathbf{m}}_p$. Results from the experiments are summarized in Tables 1 and 2 for the different capture distances for the displayed-on-screen and different size printed SDMQR codes, respectively. As shown in Table 1, the signature verification successfully established the authenticity of the displayed-on-screen SDMQR codes for all the capture distances considered. For the printed SDMQR codes, the results in Table 2 indicate that for the $1.00 \times 1.00$ in. and the $0.75 \times 0.75$ in. versions the signature verification successfully established the authenticity of the SDMQR codes for the 9 and 12 in. capture distances considered, whereas it failed to do so for the $0.5 \times 0.5$ in. version. Upon examination we found that the primary message decoded correctly from the captures of the $0.5 \times 0.5$ in. printed barcodes whereas the secondary message did not. Because the orientation modulation used for carrying the secondary message relies on higher spatial resolution, for the smallest $0.5 \times 0.5$ in. printed barcodes we tried an additional capture distance of $D = 6$ in. and found that the signature verification was able to establish the authenticity of the SDMQR code with this closer capture distance which is also more typical of situations where users are trying to capture a small printed code. Table 2 has been augmented to include this result.

We note that our experiments used a relatively low-end smartphone display and also a relatively low-end printer and therefore demonstrate that SDMQR codes are usable with current commodity devices. Smartphone display resolutions typically increase with the progression of time and printers with higher resolutions than our low-end device are readily available for use

[p] https://github.com/openssl/openssl

[q] See footnote p.

in most applications where SDMQR-code-based authentication can be helpful. Therefore, the experiments also demonstrate that SDMQR codes are well suited for immediate practical deployment and can provide a bulwark against quishing attacks in both physical and digital worlds.

TABLE 1: Authentication Results for SDMQR Codes Displayed on a Smartphone

| Capture Distance (in.) | Signature Verification Result |
|---|---|
| 12 | ✓ |
| 15 | ✓ |
| 18 | ✓ |

TABLE 2: Authentication Results for Printed SDMQR Codes

| SDMQR Code Size (in.) | Capture Distance (in.) | Signature Verification Result |
|---|---|---|
| 1.00×1.00 | 9 | ✓ |
| | 12 | ✓ |
| 0.75×0.75 | 9 | ✓ |
| | 12 | ✓ |
| 0.50×0.50 | 6 | ✓ |
| | 9 | × |
| | 12 | × |

## Alternative Realizations

For concreteness, our preceding presentation highlighted a specific realization for SDMQR codes in the context of combating quishing attacks. Several alternative constructions may also be utilized, which may impose their own requirements and/or provide other functionality. First, while our description focused on the situation where the primary data comprised a URL, the SDMQR construction can be used regardless of the nature of the primary data and also in situations where the primary data may be in a proprietary format. For instance, in QR-code-based payment systems at point of sale locations (e.g., Walmart Pay[r] and Kohl's Pay[s], the barcodes may use proprietary encoding for the primary data, intended to be read by a specialized app. SDMQR-code-based authentication can also be retrofitted into such QR codes without impacting their primary functionality. Second, while our realization presented in the last section utilized EdDSA, alternative digital signature algorithms could also be used with concurrent trade-offs.

[r]https://www.walmart.com/cp/walmart-pay/3205993
[s]https://www.kohls.com/feature/kohlspay.jsp

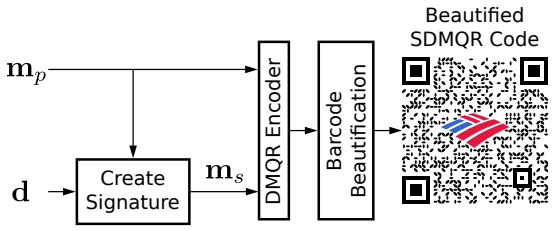


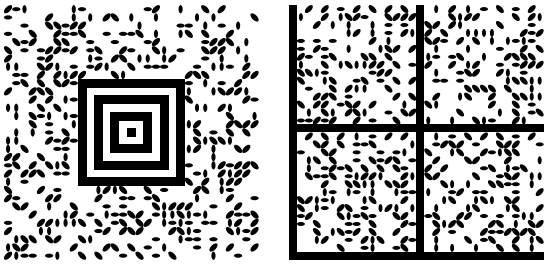

FIGURE 3: Example of beautified SDMQR code where a logo has been inserted in the center of the barcode, replacing data carrying modules. Successful recovery of the primary and secondary messages is still possible due to the error correction capabilities of SDMQR codes.

Figure 3 illustrates a beautified SDMQR code, where after the creation of an SDMQR code, the beautification process replaces the region in the center of the barcode with a logo, intentionally cannibalizing the data carrying modules that are replaced by the logo. Despite this deliberate cannibalization, the beautified SDMQR code can still be successfully decoded and authenticated due to its robust error correction, inherited from standard QR codes and optimized DMQR codes [2].

Finally, we note that the dual-modulation framework is broadly applicable [1], [2] across a wide class of 2D barcodes, including, not only QR codes, but also, Aztec codes [10] that are commonly used for airline boarding passes and Data Matrix [11] codes that are commonly used on courier packages and for the newer generation product labels. Consequently, the approach presented here can is also directly applicable for making these codes self-authenticating by exploiting dual modulation and has the same advantage that the addition of the authentication functionality is completely transparent to the primary use and functionality of these codes. Figure 4 illustrates self-authenticating versions of Aztec and Data Matrix codes with a primary message as the same URL as that for the SDMQR code in Fig. 2.

## Related Work/Discussion

Prior works have extensively explored various QR-code-based attacks and have proposed solutions to combat these threats. Of particular note, are work [12], [13], [14] that, like the proposed approach, make use of digital signatures. The innovations we propose, however, provide significant advantage over the prior



(a) Aztec (b) Data Matrix

FIGURE 4: Examples of self-authenticating Aztec (a) and Data Matrix (b) codes. These barcodes carry the same primary-secondary message pair as the SDMQR codes shown in prior figures, while still being compatible with traditional Aztec and Data Matrix code decoders.

works in two key aspects. First, *by employing dual modulation, we can enable self-authentication transparently and without disrupting existing workflows and applications*, whereas the prior approaches augment the QR code payload itself to include the digital signature. The transparency is crucial for adding the self-authentication security feature in already deployed applications of QR codes (and other 2D barcodes). Most prior approaches for self-authenticating QR codes would require updates to the reader applications to maintain functionality and it is rather challenging to do such a whole-scale synchronized update across all mobile devices on which the readers are deployed. This major barrier to enhancing existing applications with security is completely eliminated by our proposed approach. Second, the practical utility of prior approaches is severely limited by the fact that they rely on cryptographic keys of individual QR code generators for authentication of their QR codes. Authentication schemes based on symmetric key encryption, severely limit functionality (as is the case for most data authentication applications too). On the other hand, for asymmetric key cryptography, authentication based on individual public-private key pairs, poses another crucial challenge that, for authentication on the device, the mobile device needs to have the public keys for all possible signatories and also mechanisms for identifying the signatory for a specific QR code. *Our proposal for a single central signatory, or a few central signatories, using the pre-registration and signature generation protocol highlighted in Fig. 2 (d) presents a radical simplification* where only one key (or a few keys) are required on the mobile device, which is easy to accomplish. We note that using our proposed protocol, mobile devices can determine whether the information is deemed authentic by the signatory, immediately on the mobile device itself, without accessing the internet for the purpose of authentication. The process therefore does not add the delay/latency associated with communication back and forth over a network, which is undesirable. Table 3 compares the functionality provided by the proposed SDMQR codes against key prior approaches for adding security to QR codes, viz., the previously mentioned SQRC codes and the Usable Cryptographic QR codes approach [14]. SQRC codes are not self-authenticating, instead they provide a public data area that can be read with commodity QR code readers and a private data area that requires a dedicated reader. We use a cyan check-mark in the table to indicate this complementary security functionality.

TABLE 3: Comparison of functionality for the proposed SDMQR codes against prior approaches for securing QR codes.

| | SDMQR codes | SQRC | Usable Cryptographic QR codes [14] |
|---|---|---|---|
| Self-Authentication | ✓ | × | ✓ |
| Backward Compatibility | ✓ | ✓ | × |
| Efficient Key Management | ✓ | × | × |
| Commodity Hardware Compatibility | ✓ | ✓ | ✓ |
| Spatial Footprint | ✓ | × | × |
| Partial Data Encryption | × | ✓ | × |

## Applications & Challenges

Users overwhelmingly utilize mobile-device's built-in camera applications for reading QR codes instead of installing additional applications. Incorporating SDMQR code authentication into the camera application can therefore cover a large number of users and services under the SDMQR security umbrella. For phasing in security for QR codes, backward compatibility of SDMQR codes with conventional monochrome QR codes is critical. Approaches that do not maintain backward compatibility and disrupt existing users are practically unviable given the large number of existing mobile devices. When QR codes are replaced with corresponding SDMQR codes, the change is transparent and without disruption to users of devices without the specialized capability to read the secondary message in a SDMQR code and perform authentication, although, they do not get the benefit/protection from authentication. Capability for SDMQR code authentication can

be built into new devices and can also be provided via upgrades of camera and/or QR code reader apps. Over time, SDMQR code security umbrella coverage can be extended, also dissuading criminals from quishing.

SDMQR codes would be immediately beneficial for ovecoming common quishing attacks. For example, when connecting users with their banks and financial institutions, SDMQR codes with self-authentication and adding additional safeguards, such as explicit user verification and a warning to not share personal information before following un-authenticated links, would prevent more users from unwittingly providing login credentials to quishing attackers. Similarly, with SDMQR codes, smartphone based payment systems for publicly accessed services, such as parking, can reduce the chances that users of these services succumb to quishing attacks; particularly with increased deployment and user awareness.

Beyond the immediate quishing scenarios, when scanning barcodes in physically unsecured public locations, the broader adoption of SDMQR (and other self-authenticating dual-modulated 2D barcodes) codes can provide users a clear indication and assurance regarding the veracity of the links/information provided by the scans. For instance, by using SDMQR codes for providing access to Wi-Fi connectivity in public places, "Wi-Fi jacking" vulnerabilities from fake access points can be reduced. Currently, some users are wary of scanning QR codes on restaurant menus, point of sale payment systems, and other locations because of security concerns. SDMQR codes' authentication functionality can improve user confidence and allow more individuals to benefit from their convenience.

Alternative security approaches also partly address the threat of quishing. Widespread adoption of two-factor authentication by financial institutions already provides some protection against quishing. However, social-engineering attacks, SIM (subscriber identity module) swap identity hijacking[t], and systemic vulnerabilities in two-factor authentication messaging systems[u], reiterate the need for a multi-layered approach to security instead of relying on a single lynchpin. The same conclusion applies for other security approaches. Anti-phishing security features in prominent web-browsers are being defeated by client-side cloaking techniques [15]. Browser based approaches also come with added costs with evolution of criminal strategies. To bypass a one-time, upon-receipt, screen of websites linked in email messages for security vulnerabilities, attackers set up an initially innocuous site and then replace it with malicious content after the message has passed screening. Therefore, anti-phishing solutions now re-write links in emails and re-direct via a protection gateway that dynamically analyzes the linked site when the link is accessed, before passing the user onto the site. This adds considerable delay and is particularly undesirable for links embedded in QR codes. Unlike the "back-end protection" browser-based protection, which occurs downstream in the workflow with added costs, SDMQR codes offer proactive "front-end protection" against quishing before the link is even accessed. Importantly, SDMQR code authentication occurs entirely on the device with no added latency and delay for network/service access, which is important for maintaining a positive user experience. Quishing is increasingly being used to bypass email/messsaging anti-phishing solutions, making the SDMQR codes' additional protection layer quite valuable.

The central signatory approach proposed here significantly simplifies public key management for authentication of SDMQR codes on mobile devices, but it does not completely eliminate key management. Deployment of key management infrastructure and availability of authentication keys in camera and other relevant mobile device applications poses a challenge for SDMQR code adoption. The challenge is, however, significantly mitigated by the existing ecosystem for mobile devices and applications. The default camera app for mobile devices comes from the operating system (OS) and/or the device developer; both of these entities already need public key infrastructure and management set up for providing other basic functionality such as verified system and application software updates. The same is also true for other applications where SDMQR code authentication is critical, such as mobile payment systems and banking applications. By leveraging existing public key infrastructure on mobile devices, the need for additional dedicated infrastructure for supporting SDMQR codes can be minimized/eliminated. For broad adoption and interoperability of SDMQR codes, standardization of the digital signature scheme is also desirable. Our example constructions already use algorithms that are part of the NIST DSS [6]. The fact that currently Google and Apple provide the operating systems (OSs) for the vast majority of mobile devices and incorporate APIs for QR code decoding in the operating systems also implies that they should be able to readily add SDMQR code authentication functionality, which will hopefully offer a path to broader practical adoption.

[t] https://www.ic3.gov/PSA/2022/PSA220208

[u] https://www.cisa.gov/resources-tools/resources/mobile-communications-best-practice-guidance

A collaboratively established central authority is also another potential option that may be more open.

## Conclusion

Self-authenticating dual-modulated QR (SDMQR) codes proposed in this paper provide an effective method for retrofitting existing QR-code-based applications with security while maintaining backward compatibility with existing QR code readers. Specifically, a QR code encoding a URL that is already in use in an existing application can be replaced by an SDMQR code that carries the same URL as its primary message and, additionally, a secondary message that is a cryptographic signature of the primary message provided by an authorized signatory. The replacement is transparent to pre-existing QR code readers, which read the SDMQR code to retrieve the same URL that they would obtain from the original QR code. Dedicated SDMQR code readers, can not only retrieve the primary message URL but also the secondary message signature and determine whether or not the authenticity of the URL encoded in the primary message has been established by an authorized signatory. By signaling to users whether or not the authenticity of a barcode being read has been established, SDMQR codes can dissuade users from sharing sensitive personal information on malicious websites and thereby fight quishing. The backward compatibility of SDMQR codes is particularly attractive because existing workflows and applications require no change, particularly if SDMQR-code-based authentication is added to native camera apps that are increasingly used for reading QR codes. The methodology we propose is also directly applicable for other 2D barcodes (such as, Aztec and Data Matrix codes), which significantly expands the existing workflows in which it can be transparently incorporated.

## REFERENCES


1. I. Barron, H. J. Yeh, K. Dinesh, and G. Sharma, "Dual modulated QR codes for proximal privacy and security," *IEEE Trans. Image Process.*, vol. 30, pp. 657–669, 2021.
2. I. R. Barron and G. Sharma, "Optimized modulation and coding for dual modulated QR codes," *IEEE Trans. Image Process.*, vol. 32, pp. 2800–2810, 2023.
3. A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. Florida, USA: CRC Press, 1997.
4. Veritasium, "How do QR codes work? (I built one myself to find out)," YouTube, 30, Sep. 2024. https://www.youtube.com/watch?v=w5ebcowAJD8
5. ISO/IEC, "Information technology – Automatic identification and data capture techniques – QR code bar code symbology specification," 2024, accessed Sept. 28, 2024. https://www.iso.org/standard/83389.html
6. NIST, "Digital signature standard (DSS), FIPS 186-5," Feb. 2023, accessed Mar. 7, 2024. https://csrc.nist.gov/pubs/fips/186-5/final
7. R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
8. NIST, "Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters, NIST SP 800-186," Feb. 2023, accessed Mar. 7, 2024. https://csrc.nist.gov/pubs/sp/800/186/final
9. ——, "Recommendation for key management: Part 1 – general, NIST SP 800-57 Part 1 Rev. 5," May 2020, accessed Mar. 7, 2024. https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final
10. ISO/IEC, "Information technology – Automatic identification and data capture techniques – Aztec code bar code symbology specification," 2024, accessed Sept. 28, 2024. https://www.iso.org/standard/82441.html
11. ——, "Information technology – Automatic identification and data capture techniques – Data Matrix bar code symbology specification," 2024, accessed Sept. 28, 2024. https://www.iso.org/standard/80926.html
12. K. Krombholz, P. Frühwirt, P. Kieseberg, I. Kapsalis, M. Huber, and E. Weippl, "QR code security: A survey of attacks and challenges for usable security," in *Human Aspects of Information Security, Privacy, and Trust*. Cham: Springer International Publishing, 2014, pp. 79–90.
13. R. Focardi, F. L. Luccio, and H. A. M. Wahsheh, "Usable cryptographic QR codes," in *2018 IEEE International Conference on Industrial Technology*, 2018, pp. 1664–1669.
14. R. Focardi, F. L. Luccio, and H. A. Wahsheh, "Usable security for QR code," *Journal of Information Security and Applications*, vol. 48, p. 102369, 2019.
15. P. Zhang, A. Oest, H. Cho, Z. Sun, R. Johnson, B. Wardman, S. Sarker, A. Kapravelos, T. Bao, R. Wang, Y. Shoshitaishvili, A. Doupé, and G.-J. Ahn, "CrawlPhish: large-scale analysis of client-side cloaking techniques in phishing," *IEEE Security & Privacy*, vol. 20, no. 02, pp. 10–21, Mar. 2022.


**Irving Barron** is an assistant professor of instruction in the Department of Electrical and Computer Engi-

neering at the University of Rochester. His current research interests include security and forensics, and computer vision. He is a member of the IEEE. ibarron@ur.rochester.edu

**Gaurav Sharma** is professor in the Departments of Electrical and Computer Engineering and Computer Science at the University of Rochester. His current research interests include security and forensics, computer vision, and color imaging. He is a fellow of the IEEE, of SPIE, and the Society for Imaging Science and Technology. g.sharma@ieee.org