Enhancing QR Code System Security by Verifying the Scanner's Gripping Hand Biometric

Ruxin Wang
Louisiana State University
Baton Rouge, United States
rwang31@lsu.edu

Long Huang
Louisiana State University
Baton Rouge, United States
lhuan45@lsu.edu

Kaitlyn Madden
Louisiana State University
Baton Rouge, United States
kmadde5@lsu.edu

Chen Wang
Louisiana State University
Baton Rouge, United States
chenwang1@lsu.edu

ABSTRACT

Because of the great convenience and being not readable to humans, Quick Response (QR) codes are increasingly being utilized to offer a variety of security applications to mobile users, such as online payments, website logins, and private data sharing. To facilitate these security applications, QR codes usually contain sensitive information, such as bank account details, credit card numbers, and personal/organizational/device data, or they are specifically designed to work with cloud servers to provide security services. However, there is currently no existing solution to verify the identity of the smartphone user who scans a QR code from a Kiosk or another phone's screen. Verifying the scanner's identity is essential to ensure that financial transactions go to the correct recipient and that sensitive data is securely shared to its intended destination. This work aims to equip QR code providers with the ability to verify human scanners' identities, facilitating authorization and auditing. When a phone is held close to scan a QR code, we utilize the front camera of the code provider (a Kiosk or phone) to simultaneously verify the scanner's hand. Instead of requiring the scanner to present a stretched palm to obtain traditional hand geometries, we find that the geometry of an individual's hand, when it grips a phone, is also identifiable. We thus design a vision-based approach to extract gripping hand biometrics. We leverage the QR code's screen to cast light onto the scanner's gripping hand, ensuring adequate illumination even in low-light conditions. We then use a hand tracking tool, MediaPipe, to detect and localize the hand and develop a transformer-based algorithm to verify four types of gripping hand biometric features extracted from the hand image, including hand contour, skeleton, color, and surface. We further capture the subtle hand joint movements for liveness validation, because the user needs to click touchscreen buttons to start QR code scanning. Extensive experiments, including a long-term study spanning over 32 months, show that the system achieves 98.3% accuracy in verifying the user and mitigating 2D and 3D replay attacks. Compared to the widely used facial recognition, this approach addresses the recent struggles of identifying faces behind masks and the public concerns about privacy erosion.

CCS CONCEPTS

• **Security and privacy**; • **Human-centered computing** → **Human computer interaction (HCI)**;

KEYWORDS

QR code security, hand biometric, authentication

ACM Reference Format:
Ruxin Wang, Long Huang, Kaitlyn Madden, and Chen Wang. 2024. Enhancing QR Code System Security by Verifying the Scanner's Gripping Hand Biometric. In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '24), May 27–30, 2024, Seoul, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3643833.3656128

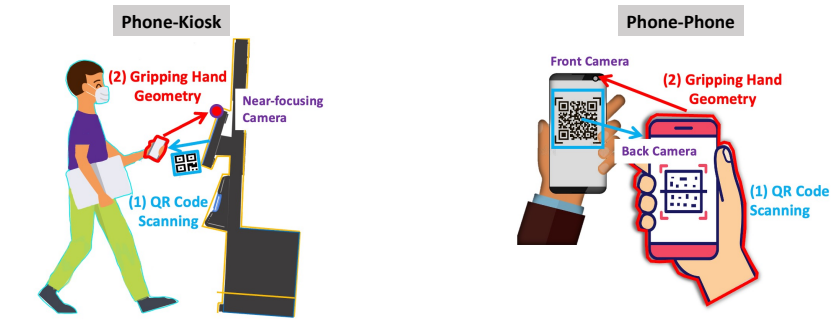
1 INTRODUCTION

These years, people are well familiar with the iconic black and white square design of Quick Response (QR) code [15]. This is an efficient way to transmit machine-readable information. Any regular display and camera can be used as the QR code provider and scanner. Due to its ease of use and compatibility with a wide range of contexts, QR codes have been increasingly used by mobile device users for many security-related applications, including transport ticketing, website logins, contact information exchange and money transfers. To initiate the process, a smartphone user needs to open a QR code scanning App and point the back camera to the target QR code, which triggers an instant response. In 2021, 75.8 million smartphone users in the United States use a QR code scanner on their smartphones, and the number is expected to reach 99.5 million by the year 2025 [23]. In China and India, QR code payment has begun to replace traditional methods such as cash and credit card payments on a large scale [16, 30].

The broad acceptance of QR codes in security services raises concerns about their inherent security vulnerabilities. One notable concern is that because QR codes are not human-readable, they could be exploited to trick users into scanning fraudulent codes that direct them to malicious websites [32]. Prior studies have addressed this issue by verifying the QR code's authenticity to protect the scanners [4, 9, 21, 22, 27, 49]. However, none of prior studies have

WiSec '24, May 27–30, 2024, Seoul, Republic of Korea

ACM ISBN 979-8-4007-0582-3/24/05.
https://doi.org/10.1145/3643833.3656128



(a) Kiosk system (e.g., ticketing, self-check in/out) (b) QR code scanning between phones

Figure 1: Enhanced QR code system security by verifying the scanner's identity (illustrated with two typical scenarios).

considered the scenario where the scanners themselves could be adversarial and the QR code provider needs protection. It is crucial to ensure that the person scanning the QR code is a permitted user and not an adversary who exploits a lost or compromised device [45]. We also find that an adversary can use a camera-relay approach to remotely scan a money-transfer QR code and secretly take the money away.

This work addresses the threats from QR code scanners to protect the provider, no matter whether the QR code is generated locally or by the cloud. We consider two common scenarios as illustrated in Figure 1. In the **Phone-Kiosk scenario**, a smartphone user scans a QR code displayed by a Kiosk or another user's smartphone. Kiosk Systems have gained increasing deployment in public places for self-services, including street vendors, self-checking in/out, auto ticketing, and food ordering [11, 12, 18, 40]. These Kiosk services are all identity-dependent, requiring human-Kiosk interaction for authentication before service. Specifically, the Kiosk displays a QR code on its screen to initiate the interaction. The user then scans it with a phone and completes the authentication on the phone's screen (e.g., enter a PIN or password). In the **Phone-Phone scenario**, scanning QR codes between two smartphones is typically needed for sharing contact details and conducting transactions, such as using PayPal [37], Zelle [19] and Venmo [48]. Verifying the identity of the scanner not only prevents unauthorized access but also guarantees non-repudiation and enables auditing.

To address threats from QR code scanners, recent Kiosks have begun replacing QR code-based authentication with facial recognition technologies [46, 50]. There is still no solution to secure the QR code scanning between smartphone users. Though it is possible for the provider's smartphone to take a face shot for identification, we find that this process requires the scanner's active participation and is hard to do simultaneously with QR code scanning. More importantly, facial recognition is subject to the following concerns: 1) Face ID is extremely private personal information, and the deployment of facial recognition in public places may lead to privacy erosion, societal apprehension, racial bias, and potential misuse of power. As a result, many states have prohibited its use [6]; 2) It is hard to verify users who wear masks. A QR code provider may request the scanner to present a security token via the network to prove the identity. But this method only verifies the scanning device, but not the human user, which is vulnerable to device loss.

This work targets the potential suspicious QR code scanners and protects the providers in both the Phone-Kiosk and the Phone-Phone scenarios. We find that when users hold a smartphone to scan a QR code, their gripping hand geometry are identifiable and readily available for authentication. We thus propose a low-effort vision-based authentication system to secure the QR code scanning process by verifying the scanner's hand. It requires minimal user effort and is contact-free. As shown in Figure 1, when a user holds the smartphone towards a Kiosk or another smartphone's screen to scan a QR code, the provider's device takes a photo shot of the back of the human scanner's gripping hand for authentication. Based on that, the provider can authorize the scanner to further access the intended link, facilitating auditing and non-repudiation. The QR code screen also serves as a consistent light source to illuminate the hand in different light conditions. We develop a Transformer-based authentication system to verify the scanner's hand. The system detects and segments the hand in the image and extract four types of gripping hand biometric features after image size/color normalization and background removal. The obtained features are fed into our Transformer model for verification. For the QR code-based authentication that involves a trusted intermediary in the cloud, the scanner's identity has been claimed by the intermediary using the scanner device's security token. Our system thus verifies the gripping hand against the claimed user identity. For the peer-to-peer QR communications, our system verifies the scanner against a group of registered users. Further, the system extracts hand joint-level motions of touchscreen button clicks for liveness validation.

Our contributions are summarized as follows:

- We propose a vision-based authentication system to protect the QR code provider in two scenarios, Phone-Kiosk and Phone-Phone. It enables Kiosks to verify a QR code scanner by integrating the traditional security token with the novel gripping hand biometric. It also facilitates auditing and non-repudiation for both QR code scanning scenarios.
- We find that the geometry of each individual's hand when it grips a smartphone is unique and identifiable. This form of biometric information is simpler to collect compared to traditional hand geometry biometrics, eliminating the need for the scanner's active participation.
- We leverage the existing hand-tracking interface, *MediaPipe*, to detect and obtain the hand skeleton in practical scenarios and extract four types of gripping hand geometry features. We further develop a transformer algorithm to achieve robust

Figure 2: The new hand geometry biometric.

hand verification under practical noises and facilitate the potential long-term use.

- Extensive experiments, including a long-term study, demonstrate the reliability of our system in both scenarios. Even when the user replaces a phone, we show that our system does not require additional training data from the new device. Moreover, our system achieves promising performance even under varying image backgrounds, lighting conditions and phone-holding angles.

2 BACKGROUND AND SYSTEM MODELS

2.1 Extending Hand Geometry To Gripping Hand Biometric

Hand geometry is a well-known biometric utilized by many practical user authentications. In particular, a person's hand geometry can be uniquely described by the hand shape features, including hand contour, palm width, and finger lengths/widths/thicknesses. Vision sensors are mainly used to acquire this biometric data [1, 17, 20, 28, 51]. However, the current hand biometric is only limited to the shape of a stretched hand. The user has to press the hand against a surface or hang in the air [51] to put the fingers and the palm on one plane, keep fingers straight and show the minimum self-occlusion for biometric collection. It requires active user participation and is thus hard to use in many situations.

This work aims to secure QR code scanning and protect the code providers. Because a user needs to hold a smartphone close to the provider's display to scan the QR code, we propose a new hand biometric that is readily available for the provider device to obtain during the QR code scanning process. Specifically, we extend the hand biometric concept from the stretched hand geometry to the more practical gripping hand geometry, which can be captured by any ordinary camera. The intuition is that when a user grips a device, the hand geometry characteristics are unchanged. The difference is that compared to a stretched hand, extracting biometric features from a gripping hand is more challenging, as self-occlusion causes most parts (e.g., fingers) not to be seen in a camera. In addition to the physiological features, the gripping hand shape (e.g., gripping hand contour and knuckle/joint positions) also reflects the user's individual gripping behavior. This is because people tend to adjust their hand pose according to the device's dimension/shape to grip it comfortably and tightly, which is a reflection of Anthropometry. Furthermore, the hand surface patterns (e.g., valleys, tendons and veins) and color distributions are also individually unique.

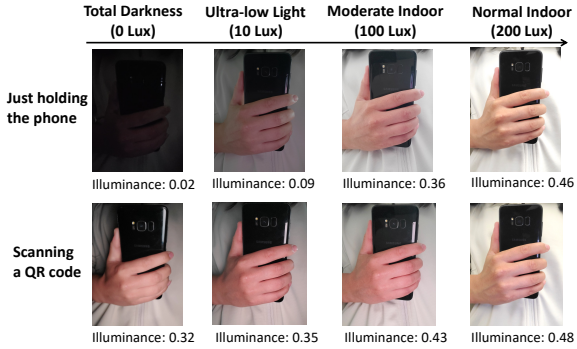


Figure 3: Using QR code display to illuminate the scanner's hand and obtain relatively consistent gripping hand images. (For illustration, light intensity is not normalized.)

Therefore, the gripping hand biometric is an integration of the user's hand geometry and gripping behavior. Figure 2 illustrates how differently people grip a smartphone. The gripping hand shapes are distinctive among all these users. Specifically, the differences are in the contours of the gripping hand, the positions and spacing of fingers, knuckle positions, skin colors, tendons, vein patterns, and the distance from each detected finger "tip" to the center of the palm. The geometric relationships between the hand and the handheld device are also unique for each participant. Therefore, capturing a user's hand geometry with an ordinary camera for verification is feasible and practical.

2.2 Gripping Hand Verification Using Existing Hand Tracking Interface and Limitations

Given the availability of existing hand tracking interfaces, we initially explore the use of these hand tracking interfaces for verifying a user's gripping hand. This task requires the capability to cope with hands that are not stretched and are self-occluded. After comprehensive searching, we find *MediaPipe Hands* [52] meets the requirement, which is a commercial interface developed by Google to track hands in real time. It consists of two parts: The palm detection model recognizes palms from a whole image; The hand landmark model performs key point localization and marks 21 landmarks (i.e., 3D coordinates of 1 wrist point and 20 finger joints) based on a pre-trained general hand model. Figure 5(A) illustrates the hand skeleton obtained by *MediaPipe* when the user's hand grips a smartphone. The self-occluded joints are predicted based on a general 3D hand model. The reconstructed hand skeleton shows individually unique palm widths and finger lengths. With the hand skeleton data returned by *MediaPipe*, we develop three methods to distinguish users' gripping hands, two of which based on the Euclidean distance between hands and inter-joint distances, respectively. The third method is plotting hand joints/skeletons as images for recognition. We find that the hand skeleton/joints obtained by *MediaPipe* exhibit individually unique information, but they are insufficient to provide reliable authentication. In particular, using images as input outperforms Euclidean distance, and using images of hand joints and skeletons alone shows a verification accuracy of 77.1% and 78.9%. When they are integrated together in an input image, the hand verification accuracy can be improved to 88.2%.

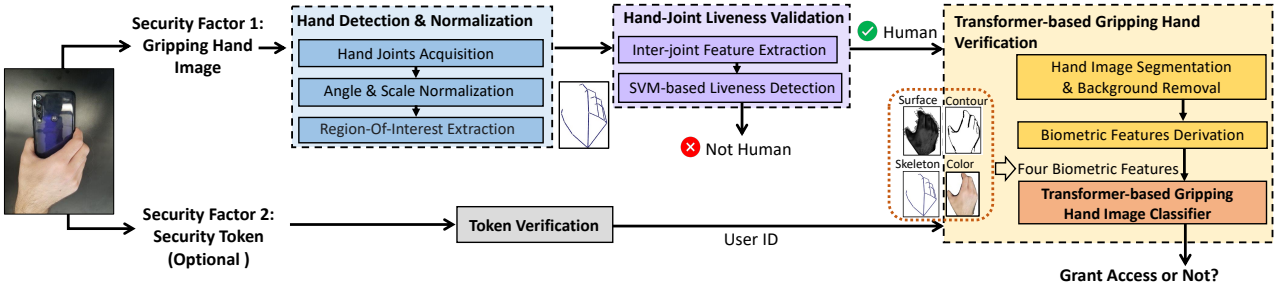


Figure 4: Flow of the proposed QR code scanner authentication.

Limitations. The low verification accuracy reflects the limitations of using the hand tracking interface (i.e., *Mediapipe*) data for gripping hand verification. This is because *MediaPipe* is designed for providing hand tracking and gesture recognition rather than verification, which fails to capture enough biometric information. In particular, it detects a hand based on a palm instead of a complete hand, which enhances the detection performance, especially with complex backgrounds. However, this design fails to provide sufficient joint/skeleton precision and resolution for distinguishing the minute hand differences among people. Besides, the 3D joint coordinates of occluded fingers, though recovered by pre-trained 3D hand models, are still unreliable for verification. More importantly, the hand joints and skeleton represent just a part of the biometric features. The hand contour, color, and surface patterns (e.g., skin textures, tendons, and veins) also carry individually unique information and are identifiable.

2.3 Challenges & Our Solutions

To verify a user's identity using gripping hand images, we need to address three critical challenges: 1) identifying representative and reliable biometric features in the gripping hand images, 2) enabling the system to work under different light conditions, phone-holding angles, and distances and QR code scanning scenarios. 3) making the system robust over the long term, even when the user changes a phone, has slight hand geometry changes or shows inconsistent phone-holding behaviors.

Four Types of Biometric Features. We identify four types of biometric features from the gripping hand image: (1) **Skeleton:** the hand bone structure, including the hand skeleton, finger lengths and widths and the gripping hand shape, (2) **Contour:** the gripping hand contour, (3) **Surface:** the hand surface patterns, including the knuckle positions, valleys, tendons and veins, and (4) **Color:** the hand color distribution features. Our study demonstrates the effectiveness of each feature in verifying the users' hands. Moreover, by integrating the four types of biometric features, we achieve the highest verification performance. The detailed feature contribution study is illustrated in Section 4.3.

Image Backgrounds & Light Conditions. In the QR code scanning process, the image backgrounds are mainly the human scanner's clothes, which have diverse color and texture patterns. We find that *MediaPipe* can successfully detect the gripping hand in the captured images with different backgrounds. In ultra-low light scenarios, *MediaPipe* sometimes fails to detect the hand. But by leveraging the QR code display screen as a consistent light source to illuminate the scanner's hand, we ensure proper illumination in all our tested light conditions, including complete darkness (0 Lux). We capture the consistent gripping hand images as shown in Figure 3. In comparison, the image captured without a QR code display suffers significantly from varying light conditions. Furthermore, we find the screen lights enable the hand image to exhibit relatively consistent light and color features.

Different Angles & Long-term Reliability. We develop a Transformer-based model to learn the spatial patterns and relations of four types of biometric features, making the system robust and efficient in user verification regardless of camera angles, distances, slight hand geometry changes over time, and even when the user changes a phone. It is noted that while Convolutional Neural Networks (CNN) are also effective in learning spatial features from 2D images, they struggle to recognize images of the same user presented at different angles and scales. Differently, the attention mechanism in the transformer model allows it to focus on different parts of an image independently of its position in the input. This means that even if the appearance of an image class changes slightly over time, the model can still identify its core features by paying attention to the most relevant parts.

2.4 System Overview

The proposed authentication system protects the QR code provider by verifying the scanner's identity. Its architecture is shown in Figure 4. When a user holds a smartphone close to a Kiosk or another phone to scan a QR code, the system allows the provider device to take the image of the user's gripping hand as input. Depending on the QR code application scenarios, the system can verify the scanner to authorize further access to the link provided by the QR code and perform auditing. The system also offers users the option to provide their tokens, assisting in user ID derivation. During the verification process, the system retrieves the user's ID from a trusted cloud-based intermediary. The system can be deployed in the cloud or locally on the provider's device.

The core of the system consists of three components: (1) *Hand Detection & Normalization:* It leverages an existing hand tracking interface (i.e., MediaPipe) to detect the hand skeleton and 21 joints (as 3D coordinates) in the image. A bounding box can then be obtained based on the hand skeleton, which is used to extract the Region-Of-Interest, ensuring only the hand part is included. The hand joints can also be used to determine rotation angles for hand orientation calibration. (2) *Hand-Joint Liveness Validation:* It uses the hand joint coordinates to calculate the differences between

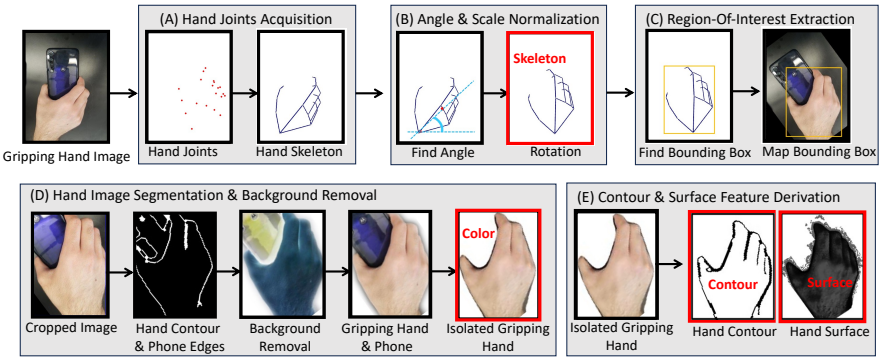


Figure 5: Illustration of the process of image processing and four types biometric feature extraction.

every two joint coordinates over a very short period (e.g., 0 to 3 seconds) as liveness features. The unique inter-joint features are then analyzed by a binary Support Vector Machine (SVM)-based classifier. (3) *Transformer-based Gripping Hand Verification:* It continues to process the hand images to verify the user's identity. Specifically, we first apply a series of image processing schemes to segment the gripping hand, remove the background, and derive four types of biometric features (i.e., hand contour, skeleton, color, and surface). It is noted that to leverage these features efficiently, we format all features as images. The four types of features are then normalized before being input into the transformer-based classifier. The transformer-based classifier recognizes the individually unique biometric patterns of gripping hands for verification.

2.5 Attack Models

The proposed system is not intended to serve all QR code scanning scenarios, but only for the QR code provider devices that need scanner verification. The system is expected to verify the scanner's identity in two QR code authentication cases: 1) The QR code itself contains sensitive information accessible to a group of users, and the scanner's identity is not directly provided. Our system needs to identify the scanner within the group (a classification problem with a limited number of classes); 2) A security token is provided by the scanner device to claim an identity, and our system verifies the gripping hand against the identity for two-factor authentication. The attacker may steal or have access to the target user's smartphone, and the legitimate user is not present at the attack scene. The long-focus camera attack when the user is present at the scene is beyond the scope of this work, which can be addressed by prior screen privacy works [7, 24, 25]. In particular, we consider **three major types of attacks:**

Zero-effort Attack: The attacker attempts to scan the QR code without presenting any biometrics. In particular, we consider a *camera-relay attack*, which takes a photo of the QR code and scans the relayed QR code somewhere else without exposing the hand.

Impersonation Attack: The attackers attempt to use their own hand to cheat our authentication system in person. Based on whether the attacker has the knowledge of the legitimate user's gripping hand shape to reproduce the biometric, we further consider a) *random impersonation attack* and b) *imitation attack*.

2D and 3D Replay Attack: According to the attacker's ability to replay the user's biometric, we consider three replay attacks: a) *2D image replay* attacker obtaining the gripping hand image of the user can print it out and present it to the system; b) *3D silicone hand replay* attacker purchases a silicone hand and shapes it according to the user's hand grip pose; c) *3D-printed hand replay* attacker leverages the latest 3D scanning & printing techniques [47] to reproduce the user's gripping hand.

3 APPROACH DESIGN

3.1 Hand Detection & Normalization

We utilize the hand tracking interface, MediaPipe, to detect and localize the hand in the captured image, which returns 21 hand joint 3D coordinates. Based on this, we normalize the hand's orientation and scale, and extract the Region-Of-Interest in the image with a bounding box. This step addresses the varying orientations and distances of the gripping hand relative to the camera in practical scanning scenarios and prepares the normalized hand image for the next step: deriving four types of gripping hand biometric features.

Hand Joints Acquisition. As illustrated in Figure 5(A), after capturing the gripping hand image from the front camera, we leverage MediaPipe's hand landmark model to obtain the 3D coordinates of 21 hand joints, where the depth information is estimated by referring a general 3D hand model. We then connect the hand joints (ignore depth) to recover the hand skeleton structure.

Angle & Scale Normalization. We utilize the derived hand skeleton to determine the alignment angle and normalization scale. As shown in Figure 5(B), we locate the hand center and connect it with the wrist point derived by *MediaPipe* to obtain the inclination angle. This angle is then used to adjust the hand image.

Region-Of-Interest Extraction. We further process the image as illustrated in Figure 5(C) to extract the Region-Of-Interest. Specifically, we identify four corner points based on the hand skeleton, which are used to form a bounding box. We then map the bounding box to the normalized hand image, and remove the parts outside of the bounding box. The cropped image can be found in Figure 5(D).

3.2 Extracting Gripping Hand Features

We further process the cropped images with hand image segmentation and background removal. The detailed steps are shown in Figure 5(D) and we finally obtain the isolated gripping hand image.

Hand Image Segmentation. To capture the gripping hand features, we need to segment the gripping hand and smartphone from the background, including the hand's contour and the edges of the gripping smartphone. It is achieved by applying the Canny

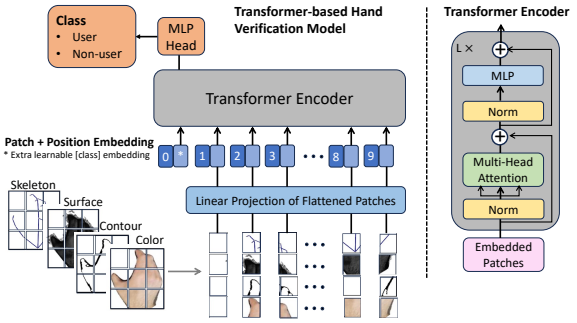


Figure 6: Transformer-based hand verification model.

Edge Detector [5] at the cropped hand image. Specifically, we apply the gradient magnitude thresholding and the lower bound cut-off suppression to eliminate spurious responses to the edge detection algorithm. A double threshold-based method is then used to determine the possible edges of the interested region. To further improve the hand detection result, we perform dilation to widen the boundaries of the hand and smartphone. Then, we erode the boundaries for a clear hand contour and smartphone edges, further expressed in a mask (binary image).

Background Removal. We continue to remove the background pixels of no interest and keep the foreground regions to highlight the hand and smartphone in a clear view. Specifically, we perform the flood fill to the above mask and perform dilation and erosion to highlight the hand-phone area. Then, this mask is used to generate a 3-channel alpha mask, which is used to blend with the original image to segment the hand-phone area from the background. Finally, we restore the original colors for the detected hand and smartphone and finish gripping hand extraction using the OpenCV K-means Clustering library [35] and obtain the isolated gripping hand image.

Biometric Features Derivation. We derive four types of biometric features as images: (1) **Skeleton:** we derive the hand skeleton, as shown in Figure 5(B), to express the unique hand bone structure; (2) **Contour:** we derive the gripping hand contour, as shown in Figure 5(E), which represents the shape of the hand when the user holds a phone; (3) **Surface:** we apply histogram equalization to the grayscale hand image to extract hand surface patterns and augment contrast ratio to enhance the visibility of valleys, tendons, and veins on the hand, as shown in Figure 5(E); (4) **Color:** we utilize the hand image processing output, as shown in Figure 5(D), to describe hand color. We resize all biometric feature images to the input dimensions of the transformer-based model.

3.3 Transformer-based Hand Image Classifier

Motivated by a prior work [10], we develop a transformer-based model to analyze the four biometric feature images (i.e., contour, skeleton, surface and color). The model is designed as a transformer-based binary classifier for verification against a claimed user ID. The model training requires feature images of the 'user' and a 'non-user' dataset consisting of data from multiple non-users. A total of four feature models are trained. During verification, each trained feature model compares the biometric feature against the user's profile and outputs two confidence scores. The final decision is made by integrating the output scores from all four biometric feature models.

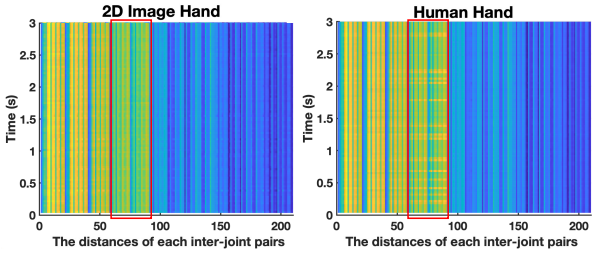


Figure 7: Using inter-joint motions for liveness validation.

The architecture of our transformer-based hand verification model is shown in Figure 6, where the input is a 2D biometric feature matrix with a dimension of $32 \times 32 \times 3$, and the output is the class probability distribution. The input is first reshaped into a sequence (i.e., nine patches) of flatten 2D patches. As the transformer uses a constant latent vector through all of its layers, we flatten the patches and map them to the 20×20×3 dimension using a trainable linear projection. The output of this projection is referred as the patch embeddings. We prepend a learnable embedding to the sequence of embedded patches, whose state at the output of the transformer encoder serves as the image representation. Position embeddings are added to the patch embeddings to retain positional information. The transformer encoder consists of alternating layers of multiheaded self-attention and MLP blocks. Layernorm (LN) is applied before every block, and residual connections are applied after every block. The multiheaded self-attention enables the transformer to process all image patches in parallel, making itself capable of learning both the biometric features embedded in each of the image patches and the relationships between different patches. The high-level biometric features derived by the transformer encoder are then passed through a flatten layer to be converted into a 1D vector. A dropout layer is then added to prevent the network from memorizing specific features of the training data, thus preventing over-fitting, after which one fully connected dense layer is deployed to shrink the size of the vector. The same architecture (One Dropout followed by one Dense layer) is repeated two more times. Finally, after the last dense layer, we obtain a score vector consisting of two real values (i.e., *user* and *non-user*). The final decision is made by integrating the outputs of the four feature models, which result in eight scores (two confidence scores for each model). The system then searches for the highest score within the integrated score vector to make the final decision.

3.4 Liveness Validation via Hand-Joint Motions

Given the proposed system is a vision-based method, it faces risks from common replay attacks, including 2D image spoofing and 3D replay attacks. Capturing the target user's gripping hand with a mobile or hidden camera to initiate 2D replay attacks isn't difficult. Furthermore, advancements in depth cameras and 3D scanning and printing technologies make 3D replays feasible. This section outlines a defense method against potential 2D and 3D replay attacks by analyzing hand joint-level motions.

Hand-Joint Motions. We notice that to scan a QR code, a user always needs to open the phone's camera or tap the smartphone screen to find a QR code app [14]. We thus propose capturing the minute hand joint-level motions for liveness validation, as it is very hard for attackers to mimic or reproduce such tiny and complex

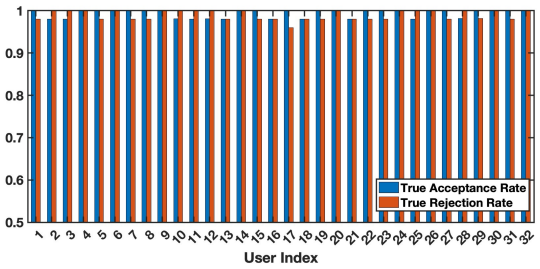


Figure 8: Performance of verifying 32 participants.

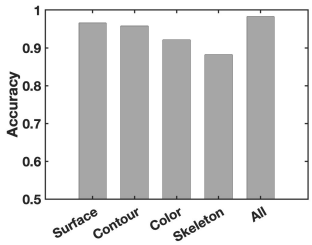


Figure 9: Biometric features significance study.

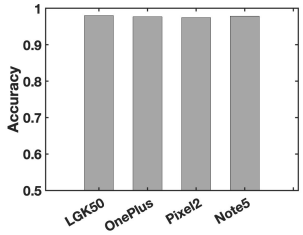


Figure 10: Testing on phones excluded from training data.

motions with a 2D image or a 3D fake hand in practice. Specifically, we compute the Euclidean Distance between every hand joint pair obtained by the hand tracking interface (e.g., MediaPipe). For each frame, we derive a total of 210 inter-joint distances. During a QR code scan (typically 0 to 3 seconds), we track the changes in each inter-joint distance and display the time series for all 210 distances in Figure 7. We find that the inter-joint distances related to the thumb finger present the largest variances. The reason is that when performing tapping, the thumb finger is inevitably included in such action. Moreover, the other finger joints also move slightly, because these fingers need to counteract the force imposed by the thumb. We select 33 inter-joint distance features (circled in the red box) to describe the unique joint-level motions of the user's hand. It is noted that we choose to use a Support Vector Machine (SVM)-based binary classifier for distinguishing a live hand from 2D images and 3D fake hands. This choice is due to its efficiency in analyzing joint motion features, eliminating the need for deep learning algorithms like Convolutional Neural Network (CNN).

4 PERFORMANCE EVALUATION

4.1 Data collection & Experimental Setup

We recruit 32 participants, comprising 26 males and 6 females, with ages ranging from 19 to 61 and collect data with six different phone models (i.e., Samsung S8, LG K50, Motorola Moto G8, OnePlus 7 Pro, Google Pixel2, and Samsung Note5). The IRB approval has been obtained. Before starting, we explain to each participant the purpose of this research and the data we needed to collect. During data collection, participants are instructed to hold the phone with their dominant hands. We assumed that the dominant hand for all participants is the right hand, making the scenario more challenging since it allows easier distinction between the right and left hands. Each participant is asked to re-grab the phone scanner 100 times, which includes behavioral inconsistency (e.g., different gripping positions) and involves varying distances and viewing angles from the gripping hand to the provider phone's camera. This section comprehensively evaluates the system's performance for both phone-kiosk and phone-phone scenarios when the QR code provider is placed at a 90-degree angle (e.g., McDonald's Vending Machine). This setup requires the phone scanner to scan the QR code vertically. All images are captured by the Samsung S8's rear camera with a maximum picture resolution of 4032×3024. The experiments are conducted in a typical indoor office setting with regular ceiling lights as the default environment. In total, 9700 RGB images are collected. We use 50% of the participants' data for training, while the rest for testing. For the long-term study being over 32 months, only 50% of the participants' first-day data are used for training, and other days are only used for testing.

4.2 User Verification Performance

We first evaluate our system's performance to verify each user by their hands. Figure 8 presents the True Acceptance Rate (TAR) to measure the proportion of actual positives (i.e., legitimate users) that are correctly identified by our system, and the True Rejection Rate (TRR) to measure the proportion of actual negatives (i.e., non-users) that are rejected by our system in the verification of 32 users. We observe that our system performs well in verifying all the users, with a 99.3% TAR and a 98.9% TRR on average. Moreover, 66% of users obtain a 100% TAR, and 50% of users achieve a 100% TRR. The results indicate that the system performs well in verifying the user against the claimed ID and rejecting the non-users.

4.3 Biometric Feature Significance

To further understand the gripping hand biometric, we study the significance of each biometric feature for verification. Figure 9 presents the biometric feature contributions to user verification. The hand skeleton feature obtains the lowest accuracy of 88.2%, which explains why the current hand tracking interfaces can not be directly used for gripping hand verification. In comparison, the hand contour feature alone obtains a 95.8% accuracy. The results reflect the potential of using a depth camera to obtain the user's hand shape for verification without capturing the unwanted private areas. The hand surface patterns features achieve slightly higher performance, whose accuracy is 96.6%. Furthermore, using hand color alone obtains 92.1% accuracy. When integrating all four features together, our system achieves the highest accuracy of 98.3%. The results demonstrate that using each feature alone is feasible but not sufficient. Only by utilizing them together can we obtain the best performance.

4.4 If The User Changes A Phone

We investigate whether there is a need to collect training data from new phone models if the user replaces the old phone. The intuition is that different phone models might impact the way the user grips the phone. Ten participants and five different phone models which cover common phone dimensions for both Android and iPhone are included in this study. We choose one phone (i.e., Motorola Moto G8 ($6.35'' \times 2.98'' \times 0.35''$)) for user profile training and the other four phones (i.e., Samsung Note5 ($6.03'' \times 3.00'' \times 0.30''$), LG K50 ($6.35'' \times 3.03'' \times 0.34$"), Google Pixel2 ($5.74'' \times 2.74'' \times 0.31''$), and OnePlus 7 Pro ($6.40'' \times 2.99'' \times 0.35''$)) as new devices the user switches

Figure 11: Illustration of the 3D-printed hand replay attack.

to, which are used exclusively for testing. Figure 10 illustrates the verification performance when testing four phone models. We find that regardless of the size of testing phone models, our system achieves over 97% accuracy. Particularly, when the testing phone model has similar dimensions (e.g., Moto and LG) to the trained phone model, our system achieves its best performance with 98% accuracy. Even with significant variances in model dimensions (e.g., Moto and Pixel), our system still achieves 97.5% accuracy in user verification. The results show that the extracted features are independent of the phone model, eliminating the need to collect new data and update the model even though users change phones.

4.5 Performance Under Attack

We recruit five participants to act as attackers and launch three types of attacks as described in Section 2.5, where three participants are involved in the 3D replay attack. Each attacker is instructed to perform each attack 20 times, and their gripping hand images are collected for analysis. Below are the detailed attack setups:

Zero-effort Attack: The attackers take a photo of the QR code and scan the relayed QR code in a different location without exposing their hands.

Impersonation Attack: *1) Random impersonation*, the attackers are instructed to grab the smartphone using their own unique styles. The gripping hand images collected during this process are then used to impersonate each of the 32 legitimate users. *2) Imitation attack*, the attackers first observe gripping hand images and videos of legitimate users. Afterward, the attackers try to imitate the gripping styles of legitimate users using their own hands.

2D/3D Replay Attack: *1) 2D image replay attack*, we print out the hand images of the 32 legitimate users on A4 papers and then present these A4 papers to the verification system. *2) 3D silicone hand replay*, we utilize three fake silicone hands to mimic the gripping styles of the 32 legitimate users. *3) 3D-printed hand replay*, we use a commodity 3D scanner, Revopoint 3D Scanner-POP2 [38], to obtain 3D hand models with the gripping styles of three legitimate users from the training dataset. These scanned 3D hand models are then imported into a commodity 3D printer, Creality 3D Printer CR-10 V3 [8] to produce the 3D-printed hands for attack. The 3D-printed fake hands are expected to exhibit a $1:1$ matching with the legitimate users' hands as shown in Figure 11.

Results of Zero-effort and Impersonation. Table 1 presents the verification performance of our system under zero-effort and impersonation attacks. Our system requires a hand image input to initialize the authentication process. And if no hand image is provided, the authentication request is considered invalid and cannot be processed. Therefore, the zero-effort attack, which fails to provide a hand image to the system, achieves a 0% attack success rate. Moreover, we find that our system effectively prevents impersonation attacks. The random impersonation attack only gets a 0.31% success rate, as the attacker has no knowledge of the legitimate users' hand biometrics and gripping pose. In the gripping hand imitation attack, the attacker obtains a 0.42% attack success rate. Though performing slightly better than random impersonations, it remains challenging for imitation attackers to mimic the legitimate users' gripping hand biometrics accurately, which requires reproducing the hand shape, finger widths/lengths, joint positions and skin patterns simultaneously.

Results of 2D/3D Replay. Table 1 shows the verification performance of our system against 2D/3D replay attacks. If not considering liveness validation and only checking biometric features, our system has a 3.34% chance of allowing 2D image attacks to pass through the system. And using the 3D-printed hands, the attacker increases the success rate to 7.02%. This is because the 3D-printed hands can reproduce the user's hand shape, hand geometry, and skin patterns. Compared to impersonation attacks, the success rate of 2D image spoofing and 3D-printed hand replay attacks are much higher. The 3D silicone hand replay attack, though imitating the user's gripping pose, struggles to reproduce the user's gripping hand biometric, resulting in a 0.42% attack success rate. These findings indicate that the 2D and 3D replay attacks, facilitated by low-cost commodity hardware, are capable of deceiving vision-based authentication systems. Furthermore, there is still large room for adversaries to increase the attack success rate further by utilizing more advanced 3D scanning and printing with higher resolutions. Therefore, relying solely on image recognition methods becomes inadequate in addressing the emerging 2D and 3D replay attacks.

Replay Resilience via Liveness Validation. We now present the effectiveness of our liveness validation to defend against replay attacks, which is shown in Table 1. We find that our system effectively prevents 100% replay attacks based on liveness validation. In particular, the success rate of the 2D image replay attack decreases from 3.34% to 0%, and that of the 3D-printed hand replay attack decreases from 7.02% to 0%. This is because the replayed 2D and 3D hands are hard to accurately reproduce the minute inter-joint motions of live hands, though they may forge the shape, geometry, and skin patterns of the user's hand. These results confirm the enhanced security of our system design, which comprises a biometric features matching process and a liveness validation module.

4.6 Study of Practical Impact Factors

To study the performance in practical scenarios, we further evaluate our system with ten participants under different impact factors.

Impact Of Light Conditions. To ensure the system's usability in both indoor and outdoor environments, we evaluate the impact of lighting conditions on gripping hand images. We examine the

Table 1: Performance under various attacks.

	Attack Success Rate
Zero-effort Attack	0%
Random Impersonation Attack	0.31%
Gripping Hand Imitation Attack	0.42%
2D Image Spoofing Attack	3.34%
3D Silicone Hand Replay Attack	0.42%
3D-printed Hand Replay Attack	7.02%
2D Image Spoofing Attack (W/ liveness detect)	0%
3D Hand Replay Attack (W/ liveness detect)	0%

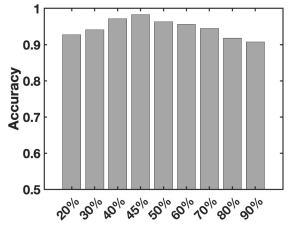


Figure 12: Illuminance level.

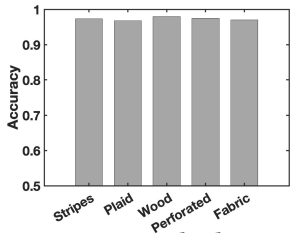


Figure 13: Image background.

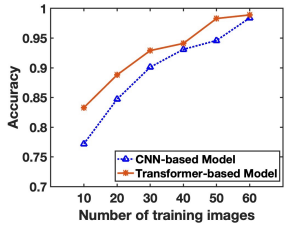


Figure 14: Training data size.

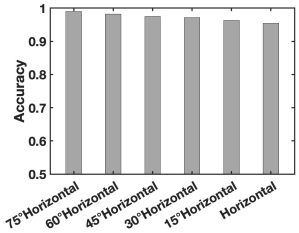


Figure 15: Phone-hold angle.

gripping hand images under different illuminance levels by calculating their average brightness [43] and estimating their illuminance between 0 and 1 using a brightness map [44]. For instance, the original illuminance of a gripping hand image collected indoors is mostly between 0.4 and 0.5, which is the only training data. To simulate various light conditions, we use the *mageEnhancer.Brightness()* method in OpenCV's PIL Library to modify the illuminance of the images. The resulting images are used for testing and fed into the authentication system as input. Figure 12 shows the user verification performance with nine different illuminance levels ranging from 0.2 to 0.9. Our system demonstrates good performance across these illuminance levels. The highest accuracy, 98.3%, is achieved with the illuminance of 0.45, which is a typical indoor light condition. Additionally, we observe a slight degradation in performance as the illuminance level increases or decreases. For instance, at illuminance levels of 0.3 and 0.4, the accuracy is 94.1% and 97.1%, respectively. When the illuminance level rises to 0.6 and 0.8, the accuracy is 95.6% and 91.8%, respectively. The brightness of a hand image can affect its contrast and color saturation, introducing potential noise during hand detection and feature extraction. However, lighting condition variations primarily affect the extraction of hand surface and color features, which minimally impacts our system due to its reliance on all four types of features. The Transformer-based model's attention mechanism ensures focus on the most critical parts, effectively minimizing the influence of lighting variations. Moreover, leveraging the QR code display as a consistent light source aids in acquiring stable hand biometric features, further enhancing our system's reliability.

Impact Of Image Backgrounds. In practical scenarios, when scanning a QR code on a Kiosk or another phone, the provider's camera may capture gripping hand images with varying backgrounds, such as the scanner's clothing or some scenery. Such background variety can make it difficult to segment the gripping hand and verify the user's identity. Therefore, we evaluate our system's performance in identifying hand images across different backgrounds. Five backgrounds with different colors and textures are included: clothes with black and white stripes, clothes with blue plaid, wood with natural white oak color, perforated sheet in yellow color, and fabric wallpaper with light gray color. We use images with black background

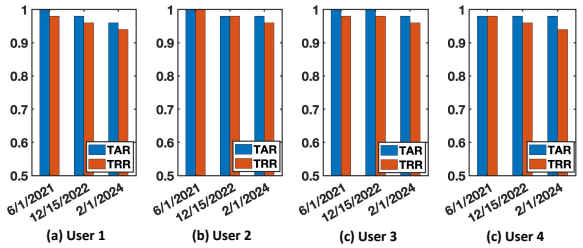


Figure 16: Long-term study performance with 4 participants (only 50% data on 6/1/2021 is used for training).

(refer Figure 11) for training and the images with five different backgrounds for testing. As shown in Figure 13, our system performs well for all backgrounds. In particular, *Wood* background performs the best with 98% accuracy, while the *Fabric* background has the lowest accuracy 97%, which is still high. The results demonstrate that our system can effectively verify users' identities regardless of the backgrounds in captured images.

Impact Of Training Data Size. We conduct a study to investigate the minimum training efforts required by varying the number of hand images from each user for training. Moreover, since Convolutional Neural Network (CNN) has also been well utilized for image recognition, we further compare the user verification performance using CNN and our Transformer-based model. As illustrated in Figure 14, we observe the Transformer-based model always outperforms the CNN model. Furthermore, when using more than 30 hand images for training, our system achieves an accuracy of over 92.9%. When using 40 and 50 images, the system achieves 94.1% and 98.3% accuracy, respectively. The results show that in practical deployment, our system only requires as few as 40 to 50 hand images for training to achieve a good performance.

4.7 Long-term Study

To better understand how the system could perform in the long term, we collect data from four participants over a 32-month period on 6/1/2021, 12/15/2022 and 2/1/2024. We use 50% of the data on 6/1/2021 for training and the rest of the 6/1/2021 data, all of the data on 2/15/2022 and 2/1/2024 for testing. The results are illustrated in Figure 16. We observe slight TAR and TRR degradation over time. Particularly, after 18 months, our system still recognizes the users' gripping hands with an average of 98.5% TAR and 97% TRR. The slight performance drop is attributed to the environmental factors and the participants' normal changes in weight, body fat ratio, and skin color in the long term. Moreover, after 32 months without model updating, our system can still recognize the user's identity with 97.5% TAR and 95% TRR. The results confirm the consistency of gripping hand biometrics and the robustness of our system in providing daily verification services over a long time.

5 PHONE-TO-PHONE: DIFFERENT ANGLES

The above results present our system performance in the Phone-Kiosk and Phone-Phone QR code scanning scenarios when the scanner holds the smartphone vertically. We notice that in the Phone-Phone scenario, some QR code providers hold their devices horizontally. Then the scanner has to hold the phone horizontally too or with small angles in order to scan the QR code. This section

investigates the gripping hand geometries associated with different phone-holding angles, especially when the testing hand images show different view angles from the training images.

Due to the differences in how the smartphone's gravity affects the hand (e.g., held horizontally versus vertically), there are slight variations in the gripping hand contour and skeleton, but most biometric features of the gripping hand are unchanged. For example, in Section 4.4, we show that a user's gripping hand biometric features are still similar when switching to a different phone model. We thus propose a novel training strategy that leverages the Transformer's generative learning capabilities. Specifically, we make the non-user training set (defined in Section 3.3) to include hand images from different angles, while the "user" training set (defined in Section 3.3) to include hand images from only one angle. The Transformer-based model is trained with this dataset to analyze the non-users hands from different angles and learn the ability to recognize the user's hand images from untrained camera angles. This training strategy, together with the generative AI model, can effectively reduce the user effort to enroll with hand images of different view angles.

We recruit ten participants for evaluation, and each participant scans the QR code at different angles 100 times, ranging from 90° to 0° horizontal. We evaluate the system's performance by training the Transformer-based model with "user" hand images at 90° horizontal only (i.e., vertical) and "non-user" hand images from 90° to 0° horizontal, and then testing with "user" hand images at angles from 75° to 0° horizontal. The results are shown in Figure 15. Our system performs well in all testing scanning angles, even when they are not included in the training set. Specifically, our system achieves 99%, 98.2%, 97.5%, 97.2% and 96.3% accuracy when the smartphone scanning angle is 75°, 60°, 45°, 30° and 15° Horizontal, respectively. The accuracy increases as the scanning angle becomes closer to one of the training data (i.e., 90° Horizontal) since the extracted hand features are similar. Moreover, when tested with 0° horizontal, our system achieves an accuracy of 95.4%, and only the user's vertical data is used for training. The results indicate that our transformer-based system can accurately verify a user's hand from a broad range of untrained angles, demonstrating unobtrusive authentication in practical QR code scanning scenarios.

6 RELATED WORK

Some studies have focused on extracting biometric information from the user's hand for authentication. For example, it has been shown that the geometry of the back of a human hand is uniquely identifiable, which can be obtained by a top-hung camera when the hand is placed on a flat surface with the palm facing downwards [39, 41]. The finger lengths, widths, angles, and distances between them are extracted as biometric features. There are also some works on scanning the palm face of the hand by placing an image sensor under the flat surface [29, 34], and both the hand geometry and the palm print patterns are used for authentication. More recent works focus on extracting hand biometrics with contact-free data acquisition methods [26, 33] or using low-cost mobile device cameras to lower the hardware cost [13]. And there is a closed related work which explores extracting hand geometry features from smartphone camera captured 2D hand gesture images to verify user's identity [51]. However, these methods are intrusive by requiring high installation overhead or active user participation, such as asking the user to press the palm against a scanner or take a stretched palm photo. They are thus hard to be deployed widely.

The latest development of Kiosks shows an increasing trend to adopt biometrics for authentication. For example, Sawetsutipun*et al.* [42] explore integrating the facial image verification technique with government Kiosks, which allows citizens to access their welfare data and receive services through a Kiosk instead of going to a government office. To further enhance the security, AlRousan*et al.* [3] propose a multi-factor authentication approach, that asks users to input a One-Time Password (OTP) received by their smartphone together with their face and fingerprint. Pichetjamroen*et al.* [36] develop a two-factor authentication system for Kiosks, whose authentication process remains to be contactless. In particular, the users need to both verify the face and show a QR code received by their smartphone to the Kiosk. Different from the above methods, which require user efforts to provide additional biometric inputs, our authentication system acquires the gripping hand biometrics simultaneously with the QR code scanning when the user holds the smartphone close to the Kiosk.

As QR codes can open links automatically, attackers can use them to redirect users to forged websites and thus improve the success rate of their phishing campaigns [32]. Also, attackers can use sneaky techniques to initialize a QR session, clone the QR code, and redirect the victims to a phishing page, ultimately allowing attackers to steal access. Several security solutions are discussed to help defend against the above attacks [2, 31, 53]. We find that these works mainly focus on defending against engineering attacks on QR codes. Admittedly, these attacks should be taken seriously and defended against in daily life to protect scanners, but equally important is the security of QR code providers when the scanner's identity could be potentially suspicious. Thus, this work fills the security gap caused by the threats from QR code scanners and provides enhanced security to the providers.

7 CONCLUSION

This work proposes an efficient and replay-resistant smartphone user verification system for QR code scanning scenarios. The proposed system uses an ordinary camera to take a photo of the back of the user's hand for verification when it grips a smartphone. We leverage the existing hand-tracking interface for hand detection and normalization, and develop a liveness validation method based on examining hand joint motions. Furthermore, we develop image processing schemes to extract gripping hand biometric features and a transformer-based algorithm to verify the user. The transformer design effectively addresses the many practical impact factors, including light conditions, camera view angles, background colors, and the change of phones. We successfully demonstrate the efficiency of our system to verify users' gripping hands with extensive experiments, including a 32-month long term, which indicates that the geometry of the gripping hand can serve as a robust and sustainable biometric factor for verifying the user's identity.

8 ACKNOWLEDGMENTS

This work is partially supported by LABoR LEQSF(2020-23)-RD-A-11 and NSF CNS-2155131.

REFERENCES

[1] Irfan Ahmad and Manzoor Khan. 2017. *Hand recognition using palm and hand geometry features.* LAP LAMBERT Academic Publishing.

[2] Mohammed S Al-Zahrani, Heider AM Wahsheh, and Fawaz W Alsaade. 2021. Secure Real-Time Artificial Intelligence System against Malicious QR Code Links. *Security and Communication Networks* 2021 (2021).

[3] Mohammad AlRousan and Benedetto Intrigila. 2020. Multi-Factor Authentication for e-Government Services using a Smartphone Application and Biometric Identity Verification. *Journal of Computer Science* (2020).

[4] Zee Media Bureau. Nov 13, 2021. QR Code Scam: User lost Rs 50,000 by scanning a code; here's how to stay safe. https://zeenews.india.com/technology/qr-code-scam-a-user-lost-rs-50000-by-scanning-a-code-here-s-how-to-stay-safe-2410046.html

[5] John Canny. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), 679–698.

[6] Electronic Privacy Information Center.epic.org. 2019. EPIC - State Facial Recognition Policy. https://epic.org/state-policy/facialrecognition/

[7] Chun-Yu Chen, Bo-Yao Lin, Junding Wang, and Kang G Shin. 2019. Keep others from peeking at your mobile device screen!. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.

[8] Creality. 2020. Creality3D CR-10 V3 3D Printer - Creality 3D. https://creality3d.shop/products/creality3d-cr-10-v3-3d-printer

[9] Nathan Daniels. Jan 17, 2022. QR Code Fraud: What is it and How Can You Protect Yourself. https://vpnoverview.com/internet-safety/cybercrime/qr-code-fraud/

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[11] Grandviewresearch.com. 2020. Market Analysis Report. https://www.grandviewresearch.com/industry-analysis/interactive-kiosk-market

[12] Nikunj Gundaniya. April 5, 2022. Everything you need to know about QR Code Payments. https://www.digipay.guru/blog/everything-you-need-to-know-about-qr-code-payments/

[13] Ahmad Hassanat, Mouhammd Al-Awadi, Eman Btoush, Amani Al-Btoush, Esra'a Alhasanat, and Ghada Altarawneh. 2015. New mobile phone and webcam hand images databases for personal authentication and identification. *Procedia Manufacturing* 3 (2015), 4060–4067.

[14] HelloTechHow. 2022. How to Scan a QR Code on an iPhone or Android. https://www.hellotech.com/guide/for/how-to-scan-qr-code-iphone-android

[15] iClassPro. 2021. How Do QR Codes Work with the Check-In Kiosk? https://support.iclasspro.com/hc/en-us/articles/360024943634-How-Do-QR-Codes-Work-with-the-Check-In-Kiosk-

[16] Harrison Jacobs. Oct 14, 2019. One photo shows that China is already in a cashless future. https://www.businessinsider.com/alipay-wechat-pay-china-mobile-payments-street-vendors-musicians-2018-5

[17] Md Khaliluzzaman, Md Mahiuddin, and Md Monirul Islam. 2018. Hand geometry based person verification system. In *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*. IEEE, 1–6.

[18] KIBIS. 2019. Types of kiosks that are assisting us in everyday life. https://medium.com/kibis/types-of-kiosks-that-are-assisting-us-in-everyday-life-ed8337f8bead

[19] Kemas kini terakhir. June 10, 2021. Zelle Pay QR code: Ensuring convenient and paperless transaction. https://www.qrcode-tiger.com/ms/zelle-pay-qr-code#Use-cases_of_using_QR_code_for_Zelle

[20] Marek Klonowski, Marcin Plata, and Piotr Syga. 2018. User authorization based on hand geometry without special equipment. *Pattern Recognition* 73 (2018), 189–201.

[21] Venkat Krishnapur. Fab 4, 2021. What is QR code phishing and how to protect yourself from it. https://indianexpress.com/article/technology/opinion-technology/what-is-qr-code-phishing-and-how-to-protect-yourself-from-it-7174553/

[22] Katharina Krombholz, Peter Frühwirt, Peter Kieseberg, Ioannis Kapsalis, Markus Huber, and Edgar Weippl. 2014. QR code security: A survey of attacks and challenges for usable security. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*. Springer, 79–90.

[23] L.Ceci. Mar 28, 2022. Number of smartphone users in the United States who used a QR code scanner on their mobile devices from 2019 to 2025. https://www.statista.com/statistics/1297768/us-smartphone-users-qr-scanner/

[24] Haibo Lei, Dan Wang, Zijian Pan, Yongpan Zou, and Kaishun Wu. 2021. iScreen: A Pure Software-based Screen Privacy Protection System for Mobile Devices. In *2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*. IEEE, 9–16.

[25] Shiguo Lian, Wei Hu, Xingguang Song, and Zhaoxiang Liu. 2013. Smart privacy-preserving screen based on multiple sensor fusion. *IEEE Transactions on Consumer Electronics* 59, 1 (2013), 136–143.

[26] Rafael M Luque-Baena, David Elizondo, Ezequiel López-Rubio, Esteban J Palomo, and Tim Watson. 2013. Assessment of geometric features for individual identification and verification in biometric hand systems. *Expert systems with applications* 40, 9 (2013), 3580–3594.

[27] Julien Maury. February 21, 2022. QR Codes: A Growing Security Problem. https://www.esecurityplanet.com/threats/qr-code-security-problem/

[28] Neha Mittal, Madasu Hanmandlu, and Shantaram Vasikarla. 2018. An Authentication System based on Hybrid Fusion of Finger-Shapes & Geometry. In *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. IEEE, 1–7.

[29] Hesham Hashim Mohammed, Shatha A Baker, and Ahmed S Nori. 2021. Biometric identity Authentication System Using Hand Geometry Measurements. In *Journal of Physics: Conference Series*, Vol. 1804. IOP Publishing, 012144.

[30] Paul Mozur. Oct 14, 2019. In Urban China, Cash Is Rapidly Becoming Obsolete. https://www.nytimes.com/2017/07/16/business/china-cash-smartphone-payments.html

[31] A Sankara Narayanan. 2012. QR codes and security solutions. *International Journal of Computer Science and Telecommunications* 3, 7 (2012), 69–72.

[32] Kaiti Norton. July 29, 2021. What are Common Types of Social Engineering Attacks. https://www.esecurityplanet.com/threats/social-engineering-attacks/

[33] Laura Gulyás Oldal and András Kovács. 2020. Hand geometry and palmprint-based authentication using image processing. In *2020 IEEE 18th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 125–130.

[34] Michael Goh Kah Ong, Tee Connie, Andrew Teoh Beng Jin, and David Ngo Chek Ling. 2003. A single-sensor hand geometry and palmprint verification system. In *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*. 100–106.

[35] OpenCV. 2020. K-Means Clustering in OpenCV. https://docs.opencv.org/4.x/d1/d5c/tutorial_py_kmeans_opencv.html

[36] Sasakorn Pichetjamroen, Ekkachan Rattanalerdnusorn, Chalee Vorakulpipat, and Achara Pichetjamroen. 2021. Multi-Factor based Face Validation Attendance System with Contactless Design in Training Event. In *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 637–640.

[37] Alison DeNisco Rayome. April 8, 2021. Use PayPal for touch-free purchases in stores. Here's how. https://www.cnet.com/tech/services-and-software/use-paypal-for-touch-free-purchases-in-stores-heres-how/

[38] Revopoint. 2022. POP 2 High-Precision 3D Scanner. https://shop.revopoint3d.com/collections/3d-scanners/products/pop2-3d-scanner?variant=42240758546667

[39] Arun Ross, Anil Jain, and S Pankati. 1999. A prototype hand geometry-based verification system. In *Proceedings of 2nd conference on audio and video based biometric person authentication*. 166–171.

[40] NATALLIA SAKOVICH. 2020. Benefits of Self-Service Kiosks for Businesses and Users. https://www.sam-solutions.com/blog/the-value-of-interactive-kiosks-for-your-business/

[41] Raul Sanchez-Reillo, Carmen Sanchez-Avila, and Ana Gonzalez-Marcos. 2000. Biometric identification through hand geometry measurements. *IEEE Transactions on pattern analysis and machine intelligence* 22, 10 (2000), 1168–1171.

[42] Srirath Sawetsutipun, Thananop Kobchaisawat, and Thanarat H Chalidabhongse. 2020. Facial Image Verification for Government Kiosk System. In *2020 17th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 65–70.

[43] Michael W Schwarz, William B Cowan, and John C Beatty. 1987. An experimental comparison of RGB, YIQ, LAB, HSV, and opponent color models. *ACM Transactions on Graphics (TOG)* 6, 2 (1987), 123–158.

[44] StackOverflow.com. 2021. What are some methods to analyze image brightness using Python? https://stackoverflow.com/questions/3490727/what-are-some-methods-to-analyze-image-brightness-using-python

[45] Ivan Stevanovic. May 12, 2022. Cell Phone Identity Theft Learn To Protect Yourself. https://dataprot.net/guides/cell-phone-identity-theft/

[46] Sur-id. 2021. Contactless Check-In Kiosk. https://www.sur-id.com/?gclid=CjwKCAjwoP6LBhBlEiwAvCcthKrwMIkDtwqv6-lw3rL5OMl2FXsIHw-blZbI1v1fPcHI5Ca-1vfNfRoCTpQQAvD_BwE

[47] Jaime Rios Velasco. 3/28/2019. 3D Scanning, 3D Modeling and 3D Printing a Human Head. https://all3dp.com/2/3d-scanning-3d-modeling-3d-printing-a-human-head-how-to/

[48] Venmo. July 21, 2021. How to Pay with Venmo in Stores. https://help.venmo.com/hc/en-us/articles/360050514333-How-to-Pay-with-Venmo-in-Stores

[49] Timothy Vidas, Emmanuel Owusu, Shuai Wang, Cheng Zeng, Lorrie Faith Cranor, and Nicolas Christin. 2013. QRishing: The susceptibility of smartphone users to QR code phishing attacks. In *International Conference on Financial Cryptography and Data Security*. Springer, 52–69.

[50] Staff Writer. 2020. Facial Recognition Photo Kiosks. https://kioskindustry.org/facial-recognition-photo-kiosks/

[51] Cong Wu, Jing Chen, Kun He, Ziming Zhao, Ruiying Du, and Chen Zhang. 2022. EchoHand: High Accuracy and Presentation Attack Resistant Hand Authentication on Commodity Mobile Devices. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 2931–2945.

[52] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. Mediapipe hands:

On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214* (2020).
[53] Yukun Zhou, Baidong Hu, Yitao Zhang, and Weiming Cai. 2021. Implementation of cryptographic algorithm in dynamic QR code payment system and its performance. *IEEE Access* 9 (2021), 122362–122372.