ABC: Enabling Smartphone Authentication with Built-in Camera

Zhongjie Ba*, Sixu Piao*, Xinwen Fu†, Dimitrios Koutsonikolas*, Aziz Mohaisen† and Kui Ren*,
*University at Buffalo, State University of New York
Email: {zba2, sixupiao, dimitrio, kuiren}@buffalo.edu
†University of Central Florida
Email: {xinwenfu, mohaisen}@cs.ucf.edu

Abstract—**Reliably identifying and authenticating smartphones is critical in our daily life since they are increasingly being used to manage sensitive data such as private messages and financial data. Recent researches on hardware fingerprinting show that each smartphone, regardless of the manufacturer or make, possesses a variety of hardware fingerprints that are unique, robust, and physically unclonable. There is a growing interest in designing and implementing hardware-rooted smartphone authentication which authenticates smartphones through verifying the hardware fingerprints of their built-in sensors. Unfortunately, previous fingerprinting methods either involve large registration overhead or suffer from fingerprint forgery attacks, rendering them infeasible in authentication systems.**

In this paper, we propose ABC, a real-time smartphone Authentication protocol utilizing the photo-response non-uniformity (PRNU) of the Built-in Camera. In contrast to previous works that require tens of images to build reliable PRNU features for conventional cameras, we are the first to observe that one image alone can uniquely identify a smartphone due to the unique PRNU of a smartphone image sensor. This new discovery makes the use of PRNU practical for smartphone authentication. While most existing hardware fingerprints are vulnerable against forgery attacks, ABC defeats forgery attacks by verifying a smartphone's PRNU identity through a challenge response protocol using a visible light communication channel. A user captures two time-variant QR codes and sends the two images to a server, which verifies the identity by fingerprint and image content matching. The time-variant QR codes can also defeat replay attacks. Our experiments with 16,000 images over 40 smartphones show that ABC can efficiently authenticate user devices with an error rate less than 0.5%.

I. Introduction

Authentication systems that identify individuals by "something the user has" are playing an increasingly important role in defeating identity theft. According to breach level index [30], 9.2 billion data records have been lost since 2013, including plaintext passwords and fingerprints. Such leakage makes knowledge-based authentication severely broken and poses particular threats, such as device-based impersonation attacks [12], to biometrics-based authentication. Therefore, there is a vast amount of works studying and implementing Multi-Factor Authentication systems which verify device's identity along with user's. Providing enhanced security without degrading user experience calls for secure and practical smartphone identification methods.

In the literature, one prevalent methodology to identify smartphones is to differentiate the fingerprints of their built-in sensors. Sensor fingerprint is a systematic distortion of sensor reading incurred by manufacturing imperfection. Such distortion remains constant for each individual hardware and exhibits strong diversity among different devices. It has been proved that the fingerprints of motion sensors, WiFi chipsets and speakers [20], [37], [6], [8] are respectively strong enough to differentiate smartphones. However, most of existing methods fail to meet two security requirements: *Fingerprint Leakage Resilience* and *Fingerprint Forgery Resilience* [4]. Although it is infeasible to steal a sensor in a smartphone, the signals generated by that sensor, in most cases, are available to the public. An adversary who has collected those signals might extract the victim's hardware fingerprint and synthesize forged signals [8], [14]. This vulnerability to the fingerprint forgery attack makes them infeasible in practice. It remains open to find usable and secure smartphone fingerprinting method that can provide physical layer proof of device's identity.

The Photo-Response Non-Uniformity (PRNU) [33] of an image sensor has been used as a physical layer fingerprint identifying conventional digital cameras in digital forensics. Given a query image taken by a camera of interest, the camera can be identified through correlating the query image's noise residue against candidate devices' reference fingerprints. In this paper, we explore using the PRNU of an image sensor on a smartphone to authenticate a user's device to defeat various frauds and attacks.

There are two grand challenges of using PRNU to identify and authenticate smartphones. First, eliminating the large registration overhead. For conventional digital cameras, normally at least 50 images are required to derive a usable reference fingerprint. Such a large registration overhead is often prohibitive for a practical smartphone authentication protocol. Second, defending against impersonation attack. The PRNU-based fingerprinting method is also vulnerable to fingerprint forgery attacks [27], [36], [24], [38]. To impersonate a victim device, an adversary could estimate the victim smartphone's fingerprint from public images and embed the obtained fingerprint into an image captured by her own device. Existing forgery

Network and Distributed Systems Security (NDSS) Symposium 2018
18-21 February 2018, San Diego, CA, USA
ISBN 1-1891562-49-5
http://dx.doi.org/10.14722/ndss.2018.23099
www.ndss-symposium.org

detection mechanisms suffer from either poor reliability [27] or huge transmission and storage overhead [36].

We performed extensive experiments to understand the characteristics of PRNU of smartphone cameras in order to address these challenges in using PRNU to identify and authenticate smartphones. A key observation is that, compared with conventional digital cameras, a smartphone's image sensor is tens of times smaller. With the same level of manufacturing imperfection, the reduction in the image sensor's dimension amplifies the pixels' dimensional non-uniformity and generates a much stronger PRNU. Our experimental results reveal that the PRNU of smartphone cameras is so strong that one image alone can uniquely identify a smartphone camera. Based on this observation, we propose directly using the PRNU estimated from the noise residue of an image taken by a smartphone as the reference fingerprint. This will significantly reduce the registration overhead of such an authentication system.

Given the unique PRNU of smartphone cameras, we propose ABC, a PRNU-based smartphone authentication protocol that can also defeat various attacks. ABC involves a registration phase and an authentication phase. During the registration phase, the user uploads a freshly captured image to the verifier/server. From this image, the verifier estimates a reference fingerprint for the user's smartphone. In the authentication phase, the verifier challenges the user to photograph and upload two time-variant QR codes, in each of which an abstract of the ongoing transaction, a random number and a time stamp are encoded. Each QR code image is also embedded with a semi-fragile probe signal that can survive photographing but not fingerprint removal. The user then puts her smartphone parallel to the screen and takes pictures of those two QR codes. She verifies the messages in the QR codes and uploads the images to the verifier. Upon receiving the images captured by the user, the verifier authenticates the user's device through the following procedure: 1) Detect the existence of the two time-variant QR codes and the target smartphone's fingerprint. Replay attacks and man in the middle attacks can be defeated by the two QR codes. 2) Detect fingerprint forgery by measuring the similarity between the two received QR code images. This is based on our observation that two images forged by the adversary contain both the fingerprint of the victim device and the fingerprint of the adversary's device, and incur a significantly higher similarity value. 3) Detect fingerprint removal through checking the strength of the probe signal embedded in each received image in case that the adversary removes the PRNU of her own device from a forged image.

Our major contributions are summarized as follows:

1 To the best of our knowledge, we are the first to explore the PRNU-based smartphone fingerprinting on a large scale. We are the first to observe that one image alone can uniquely identify a smartphone due to their unique PRNU. We conducted extensive experiments by collecting images taken by smartphones through Amazon Mechanical Turk and can achieve a total error rate below 0.5% in differentiating smartphone cameras. This new discovery makes the use of PRNU practical for smartphone authentication.

2 We propose a real-time smartphone authentication protocol that can provide reliable authentication and defeat various attacks. It has the following salient features: 1) ABC achieves secure physical layer smartphone authentication with a registration overhead of merely one photoshot. 2) Our experiments on 4,000 forged images demonstrate that ABC can detect the fingerprint forgery attack with a total error rate less than 0.47%. 3) The usability of the proposed protocol is preserved since the requirement for taking photos is familiar and convenient to smartphone users.

The rest of this paper is organized as follows. Section II reviews the current PRNU-based digital camera fingerprinting method. Section III formulates the problem to be addressed in this paper. Section IV presents our smartphone authentication protocol. Section V analyzes the security feature of the proposed protocol. Section VI conducts the performance evaluation. Section VII reviews the related existing work on hardware fingerprinting. Section VIII concludes this research.

II. Background

In this section, we first introduce the generic Photo Response Non-Uniformity (PRNU) based camera fingerprinting technique, which establishes a link between digital images and the corresponding cameras. We then introduce the fingerprint forgery attack against this fingerprinting technique and analyze existing countermeasures.

A. PRNU-based Camera Fingerprinting

PRNU [33], [23] is caused by an image sensor's non-uniform sensitivity to light. It introduces a multiplicative factor to the actual optical view. Denote the real sensor output as $\mathbf{I}$ and the actual optical view as $\mathbf{I}^{(0)}$. Any image captured by a digital camera can be represented as Equation (1) [6],

$$\mathbf{I} = \mathbf{I}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \Theta, \tag{1}$$

where $\mathbf{K}$ is the camera's PRNU, and Θ represents other noise components such as shot noise and read-out noise.

Since PRNU behaves like a white Gaussian noise variable with a variance between 3 to 5 [33], [10], it can be extracted using a denoising filter. The extracted noise residue $\mathbf{W}_{(i)}$ can be represented as Equation (2) [11],

$$\mathbf{W}_{(i)} = \mathbf{I}_{(i)}\mathbf{K} + \Xi_{(i)}, \tag{2}$$

where $\Xi_{(i)}$ is a random noise component combining Θ and other minor components.

For conventional digital cameras, the noise residue of its captured image is so noisy that it can not be directly used as a fingerprint. Therefore, an averaging process is used to reduce random components ($\Xi_{(i)}$) and to enhance PRNU ($\mathbf{K}$) [7]. It suppresses random noise components through averaging the noise residues of multiple images taken by the same camera. The obtained fingerprint can be represented as Equation (3),

$$\hat{\mathbf{K}} = \frac{\sum_{i=1}^{N} \mathbf{W}_{(i)}\mathbf{I}_{(i)}}{\sum_{i=1}^{N} (\mathbf{I}_{(i)})^2} = \mathbf{K} + \Delta, \tag{3}$$

where Δ is the difference between the estimated fingerprint $\hat{\mathbf{K}}$ and the real fingerprint $\mathbf{K}$.

The quality of the estimated fingerprint is defined as $q = corr(\mathbf{K}, \hat{\mathbf{K}})$ [27], which is the similarity between the estimated fingerprint and the real fingerprint. For each individual device, q is positively correlated to the number of images used in the averaging process. The most commonly used similarity metric is Peak to Correlation Energy (PCE) [26].

To determine if a query image is taken by a camera of interest, existing **fingerprint detection** strategies correlate the image's noise residue against that camera's reference fingerprint extracted from **at least 50 images**. Following this strategy, Goljan et al. [28] has proved camera fingerprint's *accuracy* and *user capacity* on over one million images taken by 6896 individual cameras. They show that camera fingerprint can achieve a false rejection rate less than 2.38% at false acceptance rate below 0.002% in differentiating conventional digital cameras.

B. Fingerprint Forgery Attack and Countermeasures

With a PRNU fingerprint $\hat{\mathbf{K}}$ estimated from a victim's public images, an adversary could fabricate forged images using Equation (4),

$$\mathbf{J}' = \mathbf{J}(1 + \alpha\hat{\mathbf{K}}), \tag{4}$$

where $\mathbf{J}$ is a foreign image and α controls the strength of the injected fingerprint. With an appropriate α, the fabricated image could easily pass various fingerprint detection schemes.

The state-of-the-art fingerprint forgery detection mechanisms include fragile fingerprint [36] and triangle test [27]. *Fragile fingerprint* explores the component of the PRNU noise that is fragile and removed by the lossy JPG compression. Based on the observation that the majority of images shared online are in JPG format, this mechanism assumes that an adversary derives the fingerprint from public JPG images and such a fingerprint will not contain the fragile fingerprint. If a user is required to submit uncompressed raw images for authentication, a fingerprint forgery attack can be detected through correlating the query image's noise residue against the reference fragile fingerprint of the camera of interest. However, this approach requires 300 raw images to estimate the reference fragile fingerprint, which will incur a huge transmission overhead. Moreover, the robustness of this approach relies on the secrecy of raw images. *Triangle test* is based on the observation that the injected fingerprint $\hat{\mathbf{K}}$ shares additional noise components $\Xi_{(i)}$ with every noise residue $\mathbf{W}_{(i)}$ used by the adversary. These shared $\Xi_{(i)}$s will sharply increase the PCE value between $\hat{\mathbf{K}}$ and all $\mathbf{W}_{(i)}$s. Therefore, it tests all candidate images that might be accessible to the adversary in order to detect forged images. However, due to the popularity of image sharing, it is infeasible for the verifier to collect all candidate images that are accessible to the adversary.

III. Problem Statement

In this section, we first introduce the system model and threat model. We then discuss design goals of the authentication system.

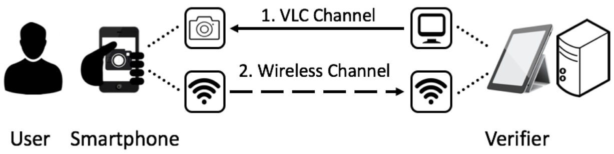


Fig. 1. System model. The verifier authenticate a user's smartphone through tracking the fingerprint of its built-in camera. The verifier first challenges the smartphone to capture and upload the image shown on its interface. Then, the verifier extracts the fingerprint of the received image and correlates it to the reference fingerprint to authenticate the smartphone.

A. System Model

Smartphone authentication is a process of verifying the possession factor (i.e., the smartphone) attached to the claimed identity of a user. Conventionally, the verification of a smartphone is achieved using a secret key controlled by a pre-installed app or an additional hardware (e.g., the secure element in iPhone). In this work, we propose to authenticate a smartphone through tracking its PRNU fingerprint as it requires no additional hardware and is physically unclonable. It is worth mentioning that the proposed ABC can be integrated with conventional cryptographic approaches to provide greater security without degrading the user experience.

Fig.1 shows the system model of ABC. The system involves three entities: *a user*, *her smartphone* and *verifier*. The user performs a transaction or login and needs to be authenticated. The *smartphone* is equipped with a built-in camera and serves as a security token. The user interacts with the verifier's interface and provides the verifier this security token in order to be authenticated. The *verifier* consists of the interface and a server. The server maintains a database of each registered user and her smartphone reference fingerprint.

Without loss of generality, we now use a point of sale (POS) terminal to illustrate the authentication process through PRNU of a smartphone. The verifier (bank) maintains a database that stores each user's account identifier (e.g., card number) and reference PRNU fingerprint. When a user requests to make a payment on the POS terminal, the verifier challenges the user who has to use her smartphone and take pictures of what is shown on the terminal's screen. The user uploads the captured images and her account identifier to the bank. The verifier then extracts the fingerprint of the user's smartphone from the images and correlates it to the reference fingerprint of the account of interest. If the correlation is higher than a threshold, the transaction will be executed. Therefore, the PRNU based authentication relies on "something you have" (i.e. the smartphone) for authentication.

Our PRNU based authentication involves two communication channels: 1) *Visible light communication (VLC) channel* from the verifier's interface to the smartphone's built-in camera. The verifier uses the VLC channel and embeds information into the image taken by the smartphone; 2) *Wireless channel* between the smartphone and the verifier. The smartphone uses the wireless channel to send the captured images to the verifier. The wireless channel may vary depending upon availability.

B. Threat Model

We assume a powerful adversary, who knows everything about the victim user and may sniff and alter the communication between the victim and the verifier, e.g., through deploying a malicious interface. The objective of the adversary is to impersonate a legitimate user and authorize a malicious request. We also assume that the adversary can access any images that the victim captures with her smartphone. Those images may be hard to be kept private anyway, for example, pictures shared through online social networks such as Facebook. However, we assume that the adversary does not physically possess the victim's smartphone.

We now use the POS terminal example again and discuss potential attacks in two cases: 1) The adversary is a malicious user who wants to make a payment with a victim's bank account. She knows the victim's account identifier and has pictures taken by the victim device. The adversary may perform the following attacks: *Replay attack* - the adversary replays the previous image tokens from the victim smartphone to the verifier. Such tokens can be obtained through eavesdropping the wireless channel of the victim smartphone from a previous authentication session. *Fingerprint forgery attack* - the adversary uploads a forged image token that is composed of the victim smartphone's fingerprint and the adversary's image. The victim smartphone's fingerprint can be obtained from the victim's public images. 2) The adversary is a malicious merchant who wants to lure a victim to authorize a malicious payment. She controls the POS terminal that processes the victim's transaction. This adversary may further conduct *Man in the middle attack* - The adversary secretly modifies the victim user's ongoing transaction. She controls the terminal to upload a modified payment request to the bank, instead of uploading the payment shown on the screen of the terminal.

C. Design Goals

We envision the following design goals for a robust and usable smartphone authentication system:

Attack resilience: the protocol should only accept fresh images captured by legitimate smartphones. It should be able to detect forged images and the images collected from the victim's previous authentication sessions.

Real-time authentication: the protocol should be able to provide accurate and real-time authentication. Both the fingerprint matching process and the attack detection process should be efficient.

User-friendliness: the protocol should provide simple and convenient interaction processes for both registration and authentication. The involved overhead should be minimal and tolerable for all involved entities.

IV. Proposed System

This section presents our real-time smartphone authentication system. We first investigate the feasibility of using PRNU as a smartphone's unique identity. We then discuss two baseline authentication schemes and their vulnerabilities. Finally, we present our full-fledged authentication protocol that achieves the aforementioned design goals.

A. Smartphone Camera Fingerprinting

Table I [1] shows that although smartphone cameras and digital cameras use similar types of image sensors, a smartphone's image sensor is often tens of times smaller than the image sensor of a traditional digital camera. The reduction in the sensor's dimension significantly degrades the light received by the image sensor, and leads to a worse signal to noise ratio (SNR) in captured images. Since the quality of the extracted fingerprint ($\mathbf{W} = \mathbf{IK} + \Xi$) is mainly determined by the image's noise components, we have to find out whether the existing fingerprint detection strategy is suitable for smartphone cameras.

To investigate the characteristics of a smartphone camera's PRNU, we collected over 16,000 images from 40 individual smartphones and evaluated their noise residues. Our experimental results (Fig. 2) demonstrate a very strong correlation between noise residues from the same smartphone camera. The fingerprint generated by a smartphone camera is much stronger than the fingerprint generated by a traditional digital camera. This is likely caused by the small size of the pixels in a smartphone's image sensor. With the same level of manufacturing imperfection, small pixels exhibit stronger non-uniformity, and hence introduce a "high-quality" fingerprint in a captured image.

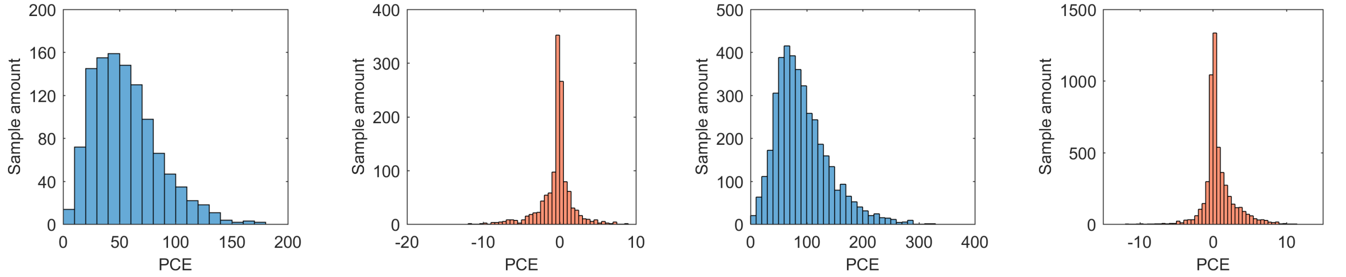
We now demonstrate the strong correlation between images captured by smartphone cameras. Since an authentication is usually carried out in an indoor environment, we look at the scenario where the tested image and the reference image are both indoor images. We note that this is also the **worst-case scenario** since the quality of the fingerprint on a captured image significantly increases with the rise of the intensity of ambient light (will be shown in section VI).

We construct two types of image pairs: 1) matching image pairs, each of which contains two images taken by the same smartphone; 2) non-matching image pairs, each of which contains two images taken by different smartphones. For iPhone 6, we tested 1250 matching image pairs and 1150 non-matching image pairs. For Galaxy Note 5, we tested 4000 matching image pairs and 5300 non-matching image pairs. Fig. 2 shows the distribution of the obtained PCE values. It can be observed that, for both smartphone models, the PCE values of the matching image pairs are significantly higher than the PCE values of non-matching image pairs. By using thresholding to differentiate matching image pairs from non-matching image pairs, we obtained the Receiver operating characteristic (ROC) shown in Fig. 3. Minimizing the total error rate of fingerprint matching based on Fig. 3, we choose 7.4338 as the matching threshold for iPhone 6 and 13.0704 for Galaxy note 5. For iPhone 6, the chosen threshold leads to a false positive rate of 0.08% at a false negative rate of 0.71%. For Galaxy Note 5, the chosen threshold leads to a false positive rate of 0.16% at a false negative rate of 0.94%.

For both smartphone models, the PRNU achieves high accuracy in differentiating image pairs even when the ambient light intensity is low. This suggests that one image alone can be used as a reference fingerprint to uniquely identify a smartphone. The reason why some image pairs are wrongly detected is because the fingerprints on those images are relatively weak. In order to further improve the identification accuracy, the verifier can increase the intensity of ambient light or use a

TABLE I. EXAMPLES OF IMAGE SENSORS FOR DIGITAL AND SMARTPHONE CAMERAS. A SMARTPHONE'S IMAGE SENSOR IS NORMALLY TENS OF TIMES SMALLER THAN A TRADITIONAL DIGITAL CAMERA'S

Digital camera	Sensor size (mm^2)	pixel amount (million)	Smartphone camera	Sensor size (mm^2)	pixel amount (million)
Canon EOS 5D Mark II	36.00×24.00	21.1	Samsung Galaxy S4	4.69 ×3.53	13
Sony A850	35.90×24.00	24.6	Apple iPhone 6	4.89×3.67	8
Nikon D300s	23.60×15.80	12.3	HTC One X	4.54×3.42	8
Pentax Pentax K-30	23.70×15.70	16.3	LG G3	4.69×3.53	13.13
Sigma SD1 Merrill	23.50×15.70	15.36	Nokia Lumia 920	4.80×3.60	8.7



(a) Matching image pairs captured by iPhone 6

(b) Non-matching image pairs captured by iPhone 6

(c) Matching image pairs captured by Galaxy Note 5

(d) Non-matching image pairs captured by Galaxy Note 5

Fig. 2. Similarity statics for images captured by smartphone cameras. PCE measures the correlation between two images' noise residues. For both iPhone 6 and Galaxy Note 5, images taken by the same smartphone (matching image pair) show significantly higher correlation than images captured by different smartphone (non-matching image pair).

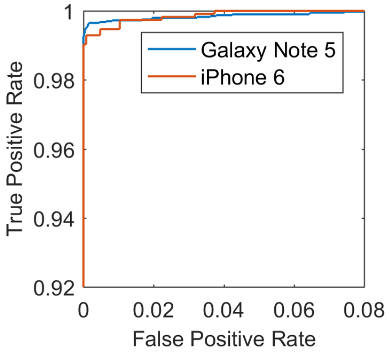


Fig. 3. ROC curve for fingerprint matching. True positive rate measures the percentage of matching images that are correctly identified. False positive rate measures the percentage of non-matching images that are identified as matching ones.

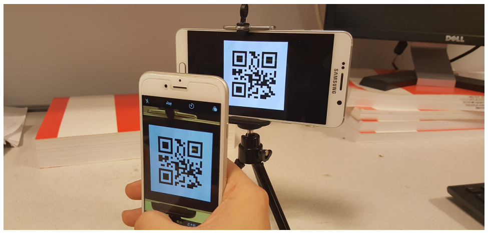


Fig. 4. Use case: a user captures an image shown on the verifier's interface to be authenticated (or registered).

reference fingerprint extracted from a bright image. As will be shown in section VI, if the images are captured in a bright environment (e.g. outdoor), the fingerprint detection strategy can achieve 100% accuracy.

Due to the high-quality fingerprint, smartphone camera fingerprinting differs from the digital camera fingerprinting in the following aspects: *Fingerprint detection strategy* - with a high-quality fingerprint on every captured image, we do not need to acquire a large number of images in order to estimate a reference fingerprint any more. Therefore, for a smartphone camera, we can use only one image's noise residue as the reference fingerprint. *Fingerprint forgery* - use of PRNU for smartphone camera fingerprinting is vulnerable to the fingerprint forgery attack. With a high-quality fingerprint on every image taken by a smartphone camera, the adversary can conduct the fingerprint forgery attack with only one reference image. Since existing forgery detection mechanisms are not practical and unreliable, it is a grand challenge to provide a trustworthy fingerprinting result.

B. Basic Authentication Schemes

Before presenting the full-fledged ABC protocol that achieves all three design goals outlined in Section III-C, we now introduce the framework of the camera fingerprint based smartphone authentication system and two baseline schemes. The first scheme can not distinguish a forged fingerprint from a genuine one. The second scheme can detect forgery attacks, but introduces a huge overhead to the verifier and the user.

1) System Framework: Fig. 4 shows a use case of the two-phase authentication process. *Registration*: the verifier constructs a fingerprint profile for a target smartphone. This phase collects the target smartphone's reference fingerprint, smartphone make and model. The registration process is conducted on the verifier's interface. *Authentication*: the verifier authenticates a smartphone in real time. The verifier challenges the user to upload freshly captured images and uses the fingerprint derived from those images to authenticate the device.

2) Basic Scheme I: This authentication scheme, shown in Fig. 5, can defeat the replay attack and the man in the middle attack. It integrates a challenge response scheme that enforces the user to capture a freshly constructed scene embedded with

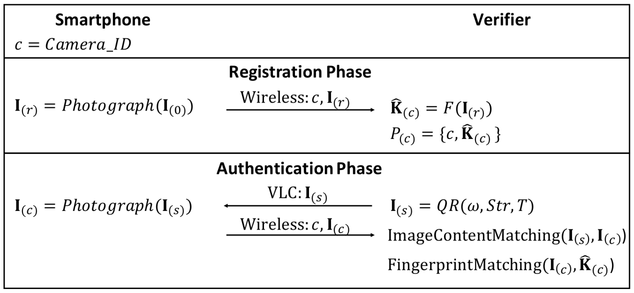

Smartphone
Verifier
$c = Camera\_ID$
Registration Phase
$\mathbf{I}_{(r)} = Photograph(\mathbf{I}_{(0)})$
Wireless: $c, \mathbf{I}_{(r)}$
$\hat{\mathbf{K}}_{(c)} = F(\mathbf{I}_{(r)})$
$P_{(c)} = \{c, \hat{\mathbf{K}}_{(c)}\}$
Authentication Phase
$\mathbf{I}_{(c)} = Photograph(\mathbf{I}_{(s)})$
VLC: $\mathbf{I}_{(s)}$
$\mathbf{I}_{(s)} = QR(\omega, Str, T)$
Wireless: $c, \mathbf{I}_{(c)}$
ImageContentMatching($\mathbf{I}_{(s)}, \mathbf{I}_{(c)}$)
FingerprintMatching($\mathbf{I}_{(c)}, \hat{\mathbf{K}}_{(c)}$)

Fig. 5. Basic Scheme I. *Registration*: the user uploads an arbitrary image captured by her smartphone. *Authentication*: the verifier challenges the user to capture a freshly constructed QR code shown on its interface. The QR code is encoded with an abstract of the ongoing transaction, which enables the user to verify the information before authorizing.

an abstract of the ongoing transaction. We propose to use a Quick Response Code (QR code) as the challenge since it can carry long messages and support fast image content matching.

The registration phase has no constraint on the user's reference image $\mathbf{I}_{(r)}$. Upon receiving the reference image uploaded by the user, the verifier extracts the fingerprint $\hat{\mathbf{K}}_{(c)}$ contained in this image and uses it to construct a profile $P_{(c)}$ for this smartphone.

During the authentication phase, upon receiving the user's authentication request, the verifier generates a QR code $\mathbf{I}_{(s)}$ that encodes an abstract of the ongoing transaction ω, a random string str and a time stamp T, displays this QR code on its interface, and challenges the user to capture it. The user photographs the QR code with her smartphone and examines the transaction embedded in the QR code. In this stage, any modification to the user's request will be noticed by the user (defeat man in the middle attack). She then uploads the captured image $\mathbf{I}_{(c)}$ to the verifier. Finally, the verifier performs *image content matching* and *fingerprint matching* to make the authentication decision. Image content matching ensures the liveness of the authentication process through detecting the newly presented QR code in the received image. Fingerprint matching verifies the producer of the received image by matching the noise residue extracted from the QR image to the target smartphone's reference fingerprint. A legitimate image token should consist of the challenging QR code and the target smartphone's fingerprint.

Although this scheme provides great convenience and strong resistance against replay attacks and man in the middle attacks, it is vulnerable to fingerprint forgery attacks. During the authentication process, the adversary could capture the presented QR code with a foreign smartphone and embed the victim smartphone's fingerprint in the captured image. Since the forged image contains both the challenging QR code and the victim smartphone's fingerprint, the verifier will accept this image as a legitimate token.

3) Basic Scheme II: To address the fingerprint forgery attack against Basic Scheme I, Basic Scheme II adopts the state-of-the-art forgery detection mechanism named *triangle test*. The main reason for not using the *fragile fingerprint* detection technique is that transmitting large number of uncompressed raw images will lead to a huge latency as discussed in section II-B. With a complete history image set, triangle test can determine with a high level of confidence whether or not the received image contains a forged fingerprint. The triangle test has two requirements for the verifier: 1) the reference fingerprint $\hat{\mathbf{K}}_{(c)}$ for the target smartphone should be extracted from a private image that is not accessible to the adversary; 2) the verifier should maintain a history image set for the target smartphone. This image set contains all of this smartphone's public images that might be accessible to the adversary.

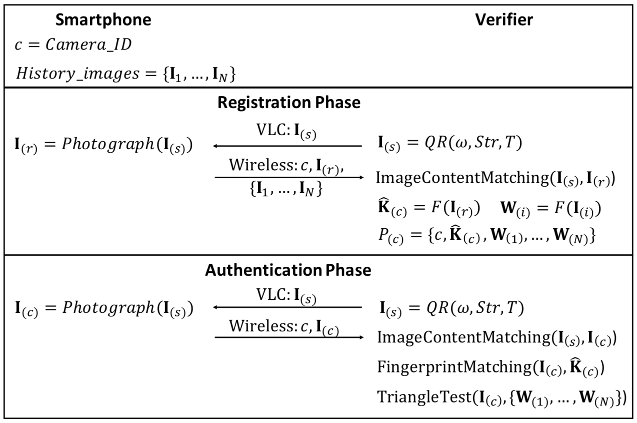

Smartphone
Verifier
$c = Camera\_ID$
$History\_images = \{\mathbf{I}_1, \dots, \mathbf{I}_N\}$
Registration Phase
$\mathbf{I}_{(r)} = Photograph(\mathbf{I}_{(s)})$
VLC: $\mathbf{I}_{(s)}$
$\mathbf{I}_{(s)} = QR(\omega, Str, T)$
Wireless: $c, \mathbf{I}_{(r)}$, $\{\mathbf{I}_1, \dots, \mathbf{I}_N\}$
ImageContentMatching($\mathbf{I}_{(s)}, \mathbf{I}_{(r)}$)
$\hat{\mathbf{K}}_{(c)} = F(\mathbf{I}_{(r)})$ $\mathbf{W}_{(i)} = F(\mathbf{I}_{(i)})$
$P_{(c)} = \{c, \hat{\mathbf{K}}_{(c)}, \mathbf{W}_{(1)}, \dots, \mathbf{W}_{(N)}\}$
Authentication Phase
$\mathbf{I}_{(c)} = Photograph(\mathbf{I}_{(s)})$
VLC: $\mathbf{I}_{(s)}$
$\mathbf{I}_{(s)} = QR(\omega, Str, T)$
Wireless: $c, \mathbf{I}_{(c)}$
ImageContentMatching($\mathbf{I}_{(s)}, \mathbf{I}_{(c)}$)
FingerprintMatching($\mathbf{I}_{(c)}, \hat{\mathbf{K}}_{(c)}$)
TriangleTest($\mathbf{I}_{(c)}, \{\mathbf{W}_{(1)}, \dots, \mathbf{W}_{(N)}\}$)

Fig. 6. Basic Scheme II. *Registration*: the user uploads one image freshly captured by her smartphone and all other images the smartphone has ever captured. *Authentication*: this process is similar to the process in basic scheme I, except that triangle test is applied to detect forged images.

Algorithm 1 Triangle Test

```
function TriangleTest($\mathbf{I}_{(q)}, \{\mathbf{W}_{(1)}, ..., \mathbf{W}_{(N)}\}$)
    $\mathbf{W}_{(q)} \leftarrow F(\mathbf{I}_{(q)})$
    for i:= 1 to N do
        $\eta \leftarrow PCE(\mathbf{W}_{(i)}, \mathbf{W}_{(q)})$
        If ($\eta > threshold$) then
            Reject
        end if
    end for
    Accept.
end function
```

Fig. 6 shows the second baseline authentication scheme. The registration phase of this scheme requires the user to upload their history image set $\{\mathbf{I}_1, ..., \mathbf{I}_N\}$ and a freshly captured image $\mathbf{I}_{(r)}$. The verifier extracts the noise residues of these images and uses them to construct a profile $P_{(c)}$ for this smartphone.

During the authentication phase, this scheme also asks the user to photograph a freshly generated QR code. After verifying the QR code and the fingerprint contained in the received image, this scheme further conducts the triangle test to detect the fingerprint forgery attack, as shown in Algorithm 1. The verifier first extracts the query image's noise residue $\mathbf{W}_{(q)}$. For each history image's noise residue $\mathbf{W}_{(i)}$, it then calculates the similarity η between $\mathbf{W}_{(q)}$ and $\mathbf{W}_{(i)}$. An η higher than a threshold suggests that $\mathbf{I}_{(q)}$ is a forged image fabricated with $\mathbf{W}_{(i)}$. The accuracy of this detection mechanism depends on the completeness of the history image set.

Although the triangle test addresses the vulnerability against the fingerprint forgery attack, it has the following

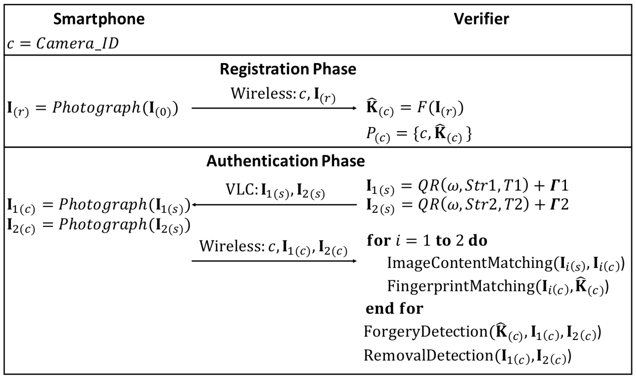

Smartphone
Verifier
$c = Camera\_ID$
Registration Phase
$\mathbf{I}_{(r)} = Photograph(\mathbf{I}_{(0)})$
Wireless: $c, \mathbf{I}_{(r)}$
$\hat{\mathbf{K}}_{(c)} = F(\mathbf{I}_{(r)})$
$P_{(c)} = \{c, \hat{\mathbf{K}}_{(c)}\}$
Authentication Phase
VLC: $\mathbf{I}_{1(s)}, \mathbf{I}_{2(s)}$
$\mathbf{I}_{1(s)} = QR(\omega, Str1, T1) + \boldsymbol{\Gamma}1$
$\mathbf{I}_{2(s)} = QR(\omega, Str2, T2) + \boldsymbol{\Gamma}2$
$\mathbf{I}_{1(c)} = Photograph(\mathbf{I}_{1(s)})$
$\mathbf{I}_{2(c)} = Photograph(\mathbf{I}_{2(s)})$
Wireless: $c, \mathbf{I}_{1(c)}, \mathbf{I}_{2(c)}$
for i = 1 to 2 do
ImageContentMatching($\mathbf{I}_{i(s)}, \mathbf{I}_{i(c)}$)
FingerprintMatching($\mathbf{I}_{i(c)}, \hat{\mathbf{K}}_{(c)}$)
end for
ForgeryDetection($\hat{\mathbf{K}}_{(c)}, \mathbf{I}_{1(c)}, \mathbf{I}_{2(c)}$)
RemovalDetection($\mathbf{I}_{1(c)}, \mathbf{I}_{2(c)}$)

Fig. 7. Full-fledged authentication protocol. *Registration*: the user uploads an arbitrary image captured by her smartphone. *Authentication*: the verifier enforces the user to capture two consecutive images shown on its interface.

drawbacks: 1) This scheme can not guarantee real-time authentication. Since the verifier needs to test the whole history image set, the response time may increase dramatically as the size of the image set increases. 2) It brings a huge burden to the user and the verifier. To maintain an up-to-date history image set for the smartphone, the user has to notify the verifier whenever they publish new pictures. 3) It is difficult to guarantee the completeness of the history image set. An incomplete history image set will make the detection result unreliable. 4) Collecting all the history images of a user might create privacy issues.

C. Full-fledged Authentication Protocol

Overcoming the drawbacks in the two baseline schemes requires a reliable and real-time detection mechanism against fingerprint forgery attacks. ABC detects the forgery attack through tracking the fingerprint of the adversary's smartphone. This fingerprint in question is introduced during the challenge response stage where the adversary captures the challenge QR code with their own smartphone. Since this fingerprint of the attacking smartphone is preserved in forged images, its existence implies a fingerprint forgery attack. ABC requires a smartphone to upload two freshly captured images. If these images are forged by an adversary, their noise residues will contain both the victim device's fingerprint and the adversary's camera fingerprint. This renders their similarity value significantly higher than a normal value.

Since a camera fingerprint can be removed with a denoising filter, the adversary can forge images containing only the victim device's fingerprint. ABC detects fingerprint removal by embedding each challenge with a probe signal that can survive photographing but not fingerprint removal and checking the existence of the probe signal in the received images.

Using the above detection mechanisms as building blocks, we now present the full-fledged ABC protocol (Fig. 7). Its registration phase is the same as the one in Basic Scheme I, which collects only one reference image from the user. The authentication phase is as follows:

Step 1. The verifier generates **two** different QR codes encoded with a transaction abstract, a time stamp and a random string. Each QR code is embedded with independent white Gaussian noise Γ_i, the variance of which is 5. The challenging scenes with QR codes can be represented as $\mathbf{I}_{i(s)} = QR(str_i, T_i) + \Gamma_i, \ i = 1, 2$. The verifier displays the two QR codes on its interface in a sequence.

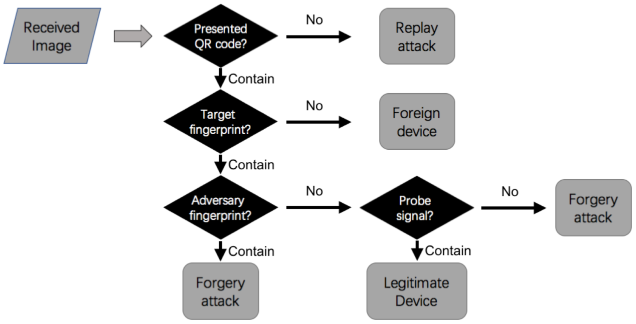


Fig. 8. Attack detection flow: since the user has confirmed the information of the ongoing transaction, the verifier needs only to detect replay attack and fingerprint forgery attack.

Step 2. The user captures $\mathbf{I}_{1(s)}$ and $\mathbf{I}_{2(s)}$, and uploads captured images to the verifier through the wireless channel.

Step 3. Upon receiving the images uploaded by the user, the verifier performs the actions shown in Fig. 8 to identify the user's smartphone:

Image content matching. Detects the challenging QR code in the received images. This can easily be achieved with off-the-shelf QR code scanning tools.

Fingerprint matching. Detects the target smartphone's camera fingerprint $\mathbf{K}_{(c)}$ in the received images by correlating the noise residue of each received image to the noise residue of the reference image.

Forgery detection. Detects the adversary's camera fingerprint $\mathbf{K}_{(a)}$ in the received images. As shown in Algorithm 2, the verifier extracts the noise residues $\mathbf{W}_{i(c)}$ of each received image $\mathbf{I}_{i(c)}$ and calculates their similarity values $PCE(\mathbf{W}_{1(c)}, \mathbf{W}_{2(c)})$. If these images are forged by the adversary, both $\mathbf{W}_{1(c)}$ and $\mathbf{W}_{2(c)}$ will contain $\mathbf{K}_{(a)}$ and $\mathbf{K}_{(c)}$, which will make $PCE(\mathbf{W}_{1(c)}, \mathbf{W}_{2(c)})$ significantly higher than the normal similarity value $PCE(\mathbf{W}_{1(c)}, \hat{\mathbf{K}}_{(c)})$.

Removal detection. Detects the added white Gaussian noise Γ_i in the received images. As shown in Algorithm 3, the verifier first subsamples each received image $\mathbf{I}_{i(c)}$ and obtains $\hat{\mathbf{I}}_{i(c)}$. With an appropriate subsampling method, $\hat{\mathbf{I}}_{i(c)}$ should contain the embedded probe signal Γ_i. The verifier then calculates the similarity value between Γ_i and the noise residue of $\hat{\mathbf{I}}_{i(c)}$. If $\mathbf{I}_{i(c)}$ has gone through a fingerprint removal process, due to Γ_i's sensitivity to fingerprint removal, the similarity value will be lower than a threshold.

V. Security Analysis

In this section, we analyze the security of the ABC protocol by examining its resistance against the *replay attack*, *man in the middle attack* and *fingerprint forgery attack*.

A. Replay Attack

An adversary may attempt to impersonate a legitimate smartphone by fraudulently replaying a captured image token

Algorithm 2 Forgery Detection

```
F2 function ForgeryDetection ($\hat{\mathbf{K}}_{(c)}$, $\mathbf{I}_{1(c)}, \mathbf{I}_{2(c)}$)
    $\mathbf{W}_{1(c)} \leftarrow F(\mathbf{I}_{1(c)})$
    $\mathbf{W}_{2(c)} \leftarrow F(\mathbf{I}_{2(c)})$
    $\delta \leftarrow PCE(\mathbf{W}_{1(c)}, \mathbf{W}_{2(c)}) - PCE(\mathbf{W}_{1(c)}, \hat{\mathbf{K}}_{(c)}))$
    If ($\delta > threshold$) then
        Reject.
    end if
end function
```

Algorithm 3 Removal Detection

```
F4 function RemovalDetection($\mathbf{I}_{1(c)}, \mathbf{I}_{2(c)}$)
    for i in [1,2] do
        $\hat{\mathbf{I}}_{i(c)} \leftarrow Subsample(\mathbf{I}_{i(c)})$
        $\hat{\mathbf{W}}_{i(c)} \leftarrow F(\hat{\mathbf{I}}_{i(c)})$
        $\Gamma_i \leftarrow i_{th}$ probe signal
        if $PCE(\hat{\mathbf{W}}_{i(c)}, \Gamma_i) < threshold$ then
            Reject.
        end if
    end for
end function
```

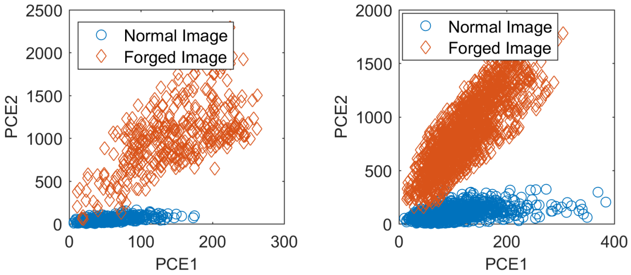
that is previously sent to the verifier. Since this image token is indeed photographed by the legitimate smartphone, without appropriate detection mechanisms, it will pass the authentication system.

To detect replayed images, ABC challenges the user to photograph a freshly generated QR code, in which a random string and a time stamp are encoded. The random string ensures that the presented QR code is hard to predict and the time stamp ensures that each QR code will be used only once for each user. In this way, the verifier can detect replay attack through checking the existence of the presented QR code in the received image. The reliability of this liveness detection mechanism is mainly determined by the entropy of the presented challenge. For QR codes, even the lowest QR code version can generate 5.7×10^{45} different images [2]. It is hardly possible for an adversary to predict the QR code to be requested in a future authentication process. Therefore, ABC has strong resistance against the replay attack.

B. Man in the Middle Attack

An adversary may attempt to lure a legitimate user to authorize a malicious request through modifying the communication between the user and the verifier. The attacking process is as follows: 1) The legitimate user initiates her request on the verifier's interface. 2) The adversary (e.g., a malicious terminal) intercepts the user's request and sends the verifier a malicious one. 3) The verifier's server sends a freshly generated QR code to the interface and challenges the user to capture it. 4) The user captures and uploads the image using her smartphone. Since the smartphone presented by the user is indeed the legitimate one, the captured image sure will pass the authentication process. However, the transaction authorized by this smartphone is not the one requested by the user.

To address this attack, ABC further embeds an abstract of the ongoing transaction into the challenging QR code. During the authentication process, the user will be required to capture the challenging QR code and to verify the information of the transaction. With this design, an adversary conducting man in the middle attack will have two options after receiving the challenging QR code (step 3): 1) Display it on the screen and ask the user to capture it. In this case, the user will terminate the authentication as the transaction encoded in the QR code is different from the one she requested. 2) Fabricate and display a forged QR code, in which an abstract of the user's original transaction in encoded. In this way, the user will confirm the transaction and photograph the QR code shown on the screen. However, since the QR code shown on the screen is different from the one generated by the verifier, the captured image token will not pass image content matching. In both cases, the adversary's transaction will not be authorized.



(a) iPhone 6 (b) Galaxy Note 5

Fig. 9. PCE for forgery detection. PCE1 measures the correlation between one tested image and the reference fingerprint. PCE2 measures the correlation between two tested images.

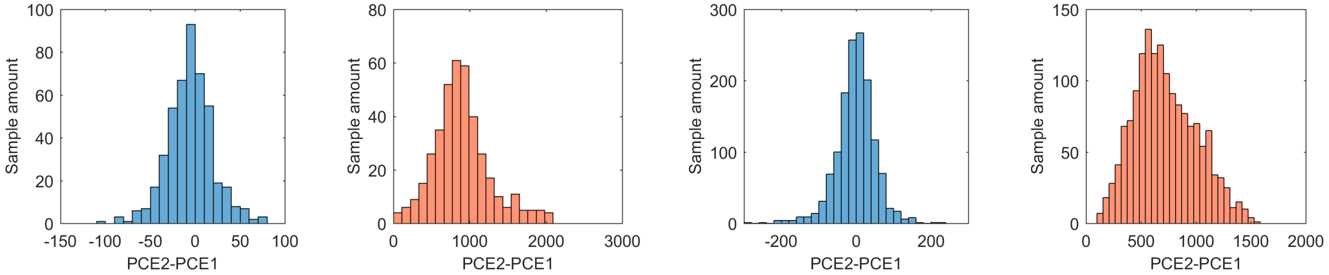
C. Fingerprint Forgery Attack

An adversary may impersonate a legitimate smartphone through fabricating images that contain the challenging QR code and the target smartphone's fingerprint. Two forgery strategies could be used: 1) directly inject the victim's camera fingerprint into an image captured by the adversarial device; 2) remove the adversary's camera fingerprint from the captured image before the injection process.

1) Forgery Strategy I: This forgery process works as follows: 1) derive two reference fingerprints from two different sets of images captured by the victim device; 2) photograph the challenging QR codes with another smartphone of the same model; 3) embed each captured image with a different reference fingerprint. Images fabricated in this way consist of the challenging QR code, the victim's camera fingerprint $\mathbf{K}_{(c)}$ and the adversary's camera fingerprint $\mathbf{K}_{(a)}$, along with other random noise components.

In order to detect this attack, our protocol adopts a forgery detection mechanism that can detect the existence of $\mathbf{K}_{(a)}$. Based on the observation that forged images sharing $\mathbf{K}_{(a)}$ will have a significant higher correlation value than legitimate images, our protocol enforces the user to capture two challenging QR codes with the same device, and uses the correlation between the captured images to detect this forgery attack.

The reliability of the detection mechanism above lies in the significance of the correlation caused by $\mathbf{K}_{(a)}$. To prove the effectiveness of this mechanism, we also look at the worst-case scenario where all tested images are captured in an indoor



(a) Normal iPhone 6 image pair

(b) Forged iPhone 6 image pair

(c) Normal Galaxy Note 5 image pair

(d) Forged Galaxy Note 5 image pair

Fig. 10. Distribution of PCE2-PCE1. For normal image pairs, PCE1 and PCE2 both measure the correlation between two legitimate images. The distribution of PCE2-PCE1 is roughly a zero mean Gaussian. For forged image pairs. PCE2 measures the correlation between two forged images sharing both the target smartphone's fingerprint and a foreign smartphone's. The foreign smartphone's fingerprint makes PCE2 significantly higher than PCE1.

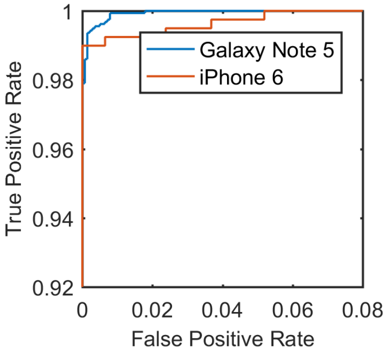


Fig. 11. Forgery detection. True positive rate measures the percentage of forged images which are correctly identified. False positive rate measures the percentage of legitimate images that are identified as forged ones.

environment. As will be shown in section VI, images captured in this environment has the weakest fingerprint. We tested two image sets collected from *Amazon Mechanical Turk* and our own device:

- *iPhone set*: 6,000 images taken by 30 different iPhone 6. The resolution is 2448×3264.
- *Samsung set*: 10,000 images taken by 10 different Galaxy Note 5. The resolution is 2048×1152.

For both image sets, we construct two kinds of image pairs for comparison: 1) Normal image pair: two images taken by the same camera, i.e., with the same $\mathbf{K}_{(c)}$. 2) Forged image pair: two forged images with the same $\mathbf{K}_{(c)}$ and $\mathbf{K}_{(a)}$. All forged image pairs are fabricated through Forgery Strategy I. For the *iPhone set*, we constructed 400 forged image pairs and 450 normal image pairs. For the *Samsung set*, we constructed 1600 forged image pairs and 1400 normal image pairs.

For each tested image pair, we calculate two similarity values. $PCE1 = PCE(\mathbf{W}_{1(c)}, \hat{\mathbf{K}}_{(c)})$ is the similarity value between one tested image's noise residue and the target smartphone's reference fingerprint. $PCE2 = PCE(\mathbf{W}_{1(c)}, \mathbf{W}_{2(c)})$ is the similarity value between tested images' noise residues. Since $PCE2$ is positively correlated to $PCE1$ for both kinds of image pairs, as shown in Fig. 9, we use the difference between $PCE1$ and $PCE2$ to differentiate normal images from forged ones. The distribution of the obtained difference is shown in Fig. 10.

ABC uses thresholding to detect fingerprint forgery attack. It counts an image pair as a forged one if the difference between $PCE2$ and $PCE1$ is above a threshold, and vice versa. Fig. 11 shows the performance of the detection result as a ROC curve. Both true positive rate and false positive rate increase with the reducing of the threshold. To minimize the total error rate of forgery detection, we choose 75.7 as iPhone set's forgery detection threshold and 162.9 as Samsung set's threshold. For iPhone 6, the chosen threshold yields a false positive rate of 0% and a false negative rate of 1.01%. For Galaxy Note 5, the false positive rate is 0.14% and the false negative rate is 0.64%.

The reason why some forged image pairs can successfully pass the forgery detection mechanism is because the $\mathbf{K}_{(a)}$ introduced during their forgery process is too weak. Because of the existence of random noise, the strength of $\mathbf{K}_{(a)}$ randomly varies between exposures even when the intensity of ambient light is fixed. If an adversary accidentally captures an image with a weak $\mathbf{K}_{(a)}$ during the authentication process, she may able to fabricate a forged image that can pass the forgery detection mechanism. However, as shown in Fig.2, the detection result will also be affected by the strength of $\mathbf{K}_{(c)}$. As PCE1 increases, the difference between PCE2 and PCE1 grows rapidly. If the verifier can increase the intensity of ambient light and raise the threshold for fingerprint matching, even images with weak $\mathbf{K}_{(a)}$ will not pass the forgery detection mechanism.

2) Forgery Strategy II: In this strategy, the adversary tries to defeat the forgery detection mechanism through removing his own fingerprint from forged images. The forgery process works as follows: 1) derive two reference fingerprints from two different sets of images from the victim; 2) photograph the challenging QR codes and remove the adversary's fingerprint from the captured image; 3) embed each obtained image with a different fingerprint of the victim. The constructed image consists of the challenging QR code, the victim's camera fingerprint, and other random noise component. This strategy may defeat our mechanism for defeating Forgery Strategy I.

ABC defeats this attack by detecting fingerprint removal. Fingerprint removal can be achieved in two ways: 1) filter the captured image with the adaptive PRNU denoising technique [31], [22]; 2) reconstruct an image containing the presented QR code. Since both removal strategies remove all noise com-

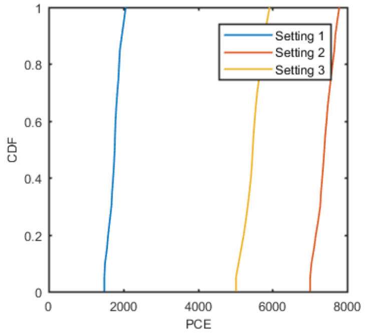


Fig. 12. Probe signal detection. Setting 1: The presented QR code does not contain the probe signal. Setting 2: The presented QR code contains a probe signal and fingerprint removal is not performed on the captured image. Setting 3: The presented QR code contains a probe signal and fingerprint removal is performed on the captured image.

ponents, we use a probe signal to detect fingerprint removal. The probe signal is semi-fragile: 1) *robust against camera-screen channel distortion* to ensure that it will be preserved in legitimate image tokens. 2) *sensitive against fingerprint removal* to ensure that the fingerprint removal process will change it. During the authentication process, the verifier embeds this probe signal Γ into the QR code to be captured by the user. In this way, fingerprint removal can be detected by checking the existence of this signal in the received image.

The reliability of this detection mechanism lies in the semi-fragility of the probe signal.

Sensitivity: The probe signal in ABC is of the same type as a camera fingerprint, i.e., white Gaussian noise with a variance of 3 to 5. With this design, the probe signal has an inherent sensitivity against adaptive PRNU denoising. Any filtering method that can remove the adversary's fingerprint will also remove the probe signal. For the second removal strategy, since the probe signal is unknown, the adversary cannot construct an image containing the probe signal without introducing their own camera fingerprint into a captured image.

Robustness: Camera-screen channel distortion may lead to an information loss in the high frequency band [29], [25]. Although this loss also affects the probe signal, the information loss caused by fingerprint removal is much more severe. To compare channel distortion and fingerprint removal, we test the probe signal with three different settings: 1) The presented QR code does not contain the probe signal. 2) The presented QR code contains an 800×800 probe signal, and the adversary does not conduct fingerprint removal on the captured image. 3) The presented QR code contains a 800×800 probe signal and fingerprint removal is performed on the captured image. In the experiment, we first put the smartphone (iPhone 6) in *parallel* to the verifier's interface (iPad mini 2) and photograph the presented QR code $I_{(s)}$. We then perform region detection and subsampling on the captured image $I_{(c)}$ to extract the challenging QR code and get $I'_{(c)}$. Finally, we calculate the PCE value between $I_{(s)}$ and $I'_{(c)}$. For each setting, we repeat the experiment 20 times and show the CDF of the PCE value in Fig. 12. It can be observed that: 1) the probe signal is preserved in the captured images. The PCE value of the second setting is significantly higher than that of the first setting; 2) using the probe signal, we can reliably detect fingerprint removal. The PCE distributions of the second and third setting have no overlapping. We note that the PCE value of the first setting is mainly caused by the image content shared between $I_{(s)}$ and $I'_{(c)}$.

Being sensitive to all fingerprint removal methods and robust against camera screen channel distortion, the probe signal applied in ABC can effectively detect fingerprint removal.

VI. Performance Evaluation

In this section, we first investigate the characteristics of a smartphone camera's PRNU. We then evaluate the efficiency of the proposed ABC protocol. Finally, a user study is conducted to demonstrate the usability of the system.

A. Experiment Setup

Configuration: The evaluation is conducted using Matlab on a Windows system with 8 Core Intel i7-4720HQ processor running at 2.6 GHz. The algorithm for fingerprint matching and extraction is based on the code by digital data embedding laboratory [28].

Image sets: The applied image sets include 6,000 images captured by 30 individual iPhone 6 devices and 10,000 images captured by 10 individual Samsung Galaxy Note 5 devices. The resolution of iPhone 6 images and Samsung Galaxy Note 5 images are 2448×3264 and 2048×1152, respectively. These images are collected from Amazon Mechanical Turk and our own devices. To ensure the randomness of the collected images, the image collection tasks we published on Mechanical Turk had no limitation on image content or the way people take photographs.

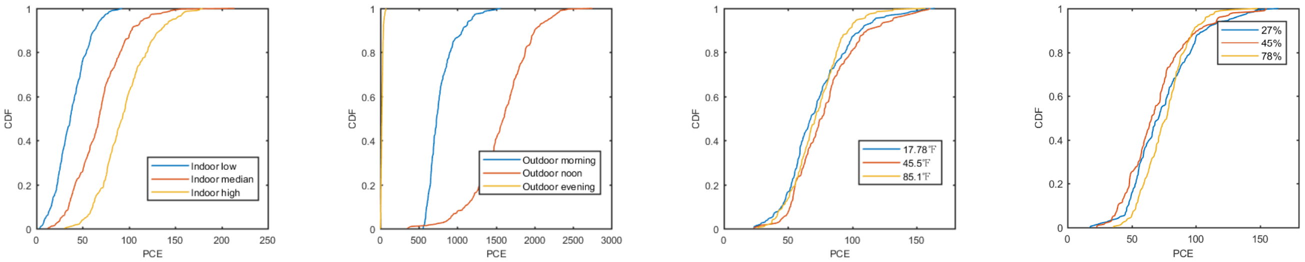
Metrics: We use the following metrics to evaluate the fingerprint of a smartphone camera. *Peak to Correlation Energy (PCE)* measures the correlation between a query image's noise residue and the reference fingerprint. It can be used to indicate the quality of the reference fingerprint and the strength of the fingerprint carried by the query image. *Cumulative distribution function (CDF)* is a graphical plot that illustrates the distribution of a value X. Given a specific value α, the CDF shows the probability that the X will take a value less than or equal to α. In this paper, CDF is used to compare the PCE distributions of different experimental settings. A setting with higher PCE value will achieve better accuracy in both fingerprint detection and forgery detection.

B. Smartphone Camera's PRNU

Before presenting the detailed setting of our experiments, we first summarize the investigated questions and our key observations as follows:

1 *Does PRNU change over time?* No. We have tested images captured in three different years. There is no significant difference in the fingerprints on those images.

2 *Will the ambient environment affect the fingerprint on an image?* Yes, we have tested the impact of **light**, **temperature** and **relative humidity**. The only factor that can affect the fingerprint is the intensity



(a) indoor images (b) outdoor images (c) Ambient temperature (d) Relative humidity

Fig. 13. Impact of ambient environment. The CDF of each setting plots a distribution of the correlation between two images captured in that environment. The only environmental factor that affects camera fingerprint is the intensity of ambient light. The strength of the fingerprint on a image significantly increases with the rise of the ambient light intensity.

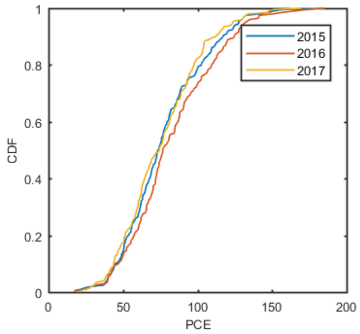


Fig. 14. The impact of age. We use a reference image captured in 2017 and conduct fingerprint matching with images captured in different years. The CDF of each year shows the distribution of the PCEs obtained for that year.

of ambient light. The strength of the fingerprint on a captured image significantly increases with the rise of the light intensity.

3 *What is the relationship between an image's resolution and the strength of its fingerprint?* Positively correlated. When cropping an image to different resolutions, the strength of its fingerprint is nearly proportional to the number of remaining pixels.

4 *How does the number of reference images affect the strength of the extracted reference fingerprint?* For each smartphone, the strength of the extracted reference fingerprint is nearly proportional to the number of reference images.

1) Impact of Age: In an authentication system, a usable hardware fingerprint should not change over time. Since the average life cycle for a smartphone is around 22 months [3], we evaluate a smartphone's PRNU with images captured in three different years: 2015, 2016 and 2017. All tested images were captured in the same room with fixed light intensity. The smartphone applied in this test is an iPhone 6.

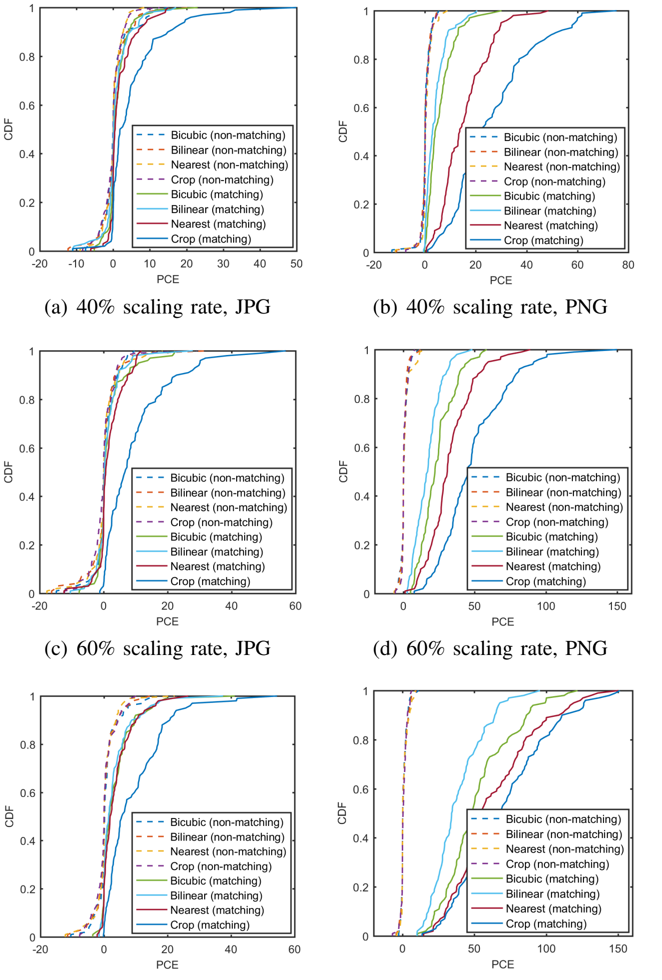
To find out if PRNU changes over time, we first extract a reference fingerprint from an image captured in 2017. Then, we conduct fingerprint matching with three image sets collected in different years. Each image set contains 200 images captured by the tested device. Fig. 14 shows the CDF of the obtained PCE value. As the reference fingerprint is captured in 2017, the CDF of 2017 shows the correlation between noise residues (fingerprints) from the same year, and the CDF of 2015 and 2016 show the correlation between noise residues from different years. Since there is no significant difference between these three CDFs, the PRNU of the tested smartphone did not change over the last three years.

2) Impact of Ambient Light: The quality of an extracted fingerprint is mainly determined by the noise components of the image of interest. Since the ambient light will affect the random noise component on a captured image, it is important to investigate the impact of ambient light on camera fingerprint. We evaluate images captured in six different environments: 1) *Indoor_low*: a windowless room with a dim filament lamp. 2) *Indoor_median*: a windowless room with several fluorescent lamps. 3) *Indoor_high*: an indoor environment with several windows. The ambient light source is the sun. 4) *Outdoor_morning*. 5) *Outdoor_noon*. 6) *Outdoor_evening*. The outdoor images are captured on a sunny day.

During the experiment, we construct 300 image pairs for each configuration and conduct fingerprint matching on those image pairs. The PCE value calculated for each image pair indicates the strength of the fingerprints carried on them. Fig. 13 shows the CDF of the obtained PCE values. The observations are as follows: 1) The strength of the fingerprint on a captured image significantly increases with the rise of the intensity of ambient light. 2) Compared with an indoor image, an outdoor image normally carries a stronger fingerprint. Therefore, one possible way to improve the identification accuracy is to extract the reference fingerprint from an outdoor image.

3) Impact of Ambient Temperature and Relative Humidity: To understand how ambient environments affect the fingerprint on a captured image, we further investigate the impact of ambient temperature and relative humidity. In order to eliminate the impact of ambient light, all tested images are captured in an indoor environment with fixed light intensity. For ambient temperature, we have tested 17.78°F, 45.5°F and 85.1°F. For relative humidity, the tested images cover 27%, 45% and 78%(a rainy day). Similar to the last experiment, we construct 200 image pairs for each configuration and conduct fingerprint matching. As shown in Fig. 13(c) and Fig. 13(d), there is no significant difference between the CDF of different configurations. Therefore, PRNU is not affected by ambient temperature or relative humidity.

4) Impact of Image Resolution: Since the resolution of the image token significantly affects the overhead of the



(a) 40% scaling rate, JPG

(b) 40% scaling rate, PNG

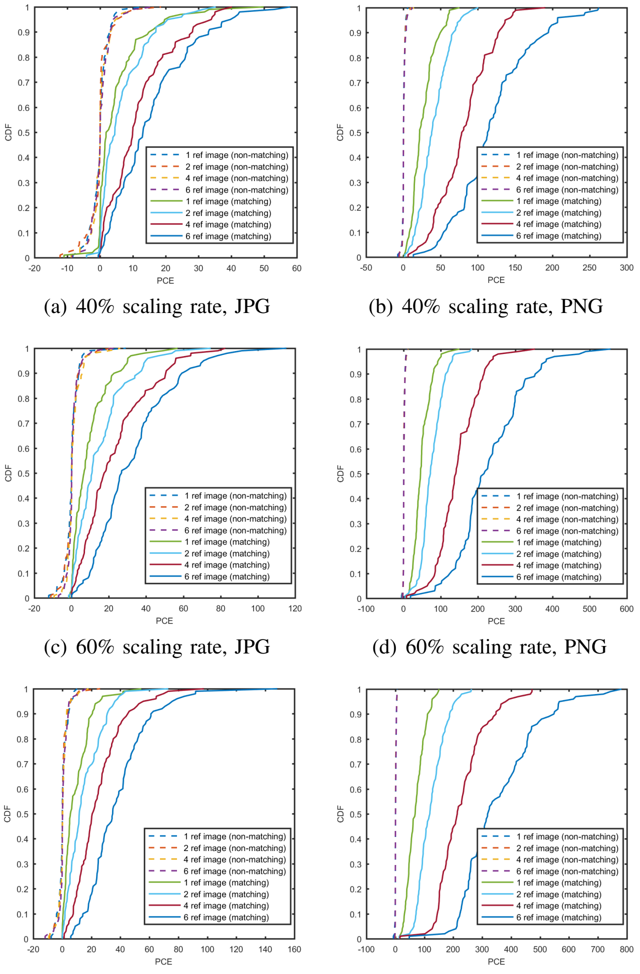
(c) 60% scaling rate, JPG

(d) 60% scaling rate, PNG

(e) 80% scaling rate, JPG

(f) 80% scaling rate, PNG

Fig. 15. Impact of image resolution. For each setting, we conduct fingerprint matching with matching and non-matching image pairs. When the resized image is stored in JPG format, the scaling ratio has no significant impact on the obtained PCE values. When PNG is used, the PCE value obtained from a matching image pair is nearly proportional to the number of remaining pixels.



(a) 40% scaling rate, JPG

(b) 40% scaling rate, PNG

(c) 60% scaling rate, JPG

(d) 60% scaling rate, PNG

(e) 80% scaling rate, JPG

(f) 80% scaling rate, PNG

Fig. 16. Impact of number of reference images. For every scaling ratio and image format, the PCE value obtained from a matching image pair is nearly proportional to the number of reference images.

authentication process (Section VI-C) in terms of the time used for authentication, we now evaluate the fingerprint detection strategy on resizing images.

The images captured by a digital camera can be resized with down-sampling or image cropping. For down-sampling, we tested three most commonly used interpolation methods: *nearest-neighbor*, *bilinear*, and *bicubic*. For image cropping, we crop a rectangular area from the target image. After resizing an image, we also need to decide the image format to be used to store it. We test the two most commonly used image formats: 1) PNG, which supports lossless image compression. The obtained image has accurate pixel values but requires more storage space. 2) JPG, which supports lossy compression. The obtained image is noisy but smaller. The scaling ratio is defined as the proportional ratio of the size of the resized image to the size of the original image. We tested different image scaling ratios from 40%-80%. Overall, we have 24 different configurations, each of which is tested with 100 matching image pairs and 100 non-matching image pairs generated from the Samsung image set.

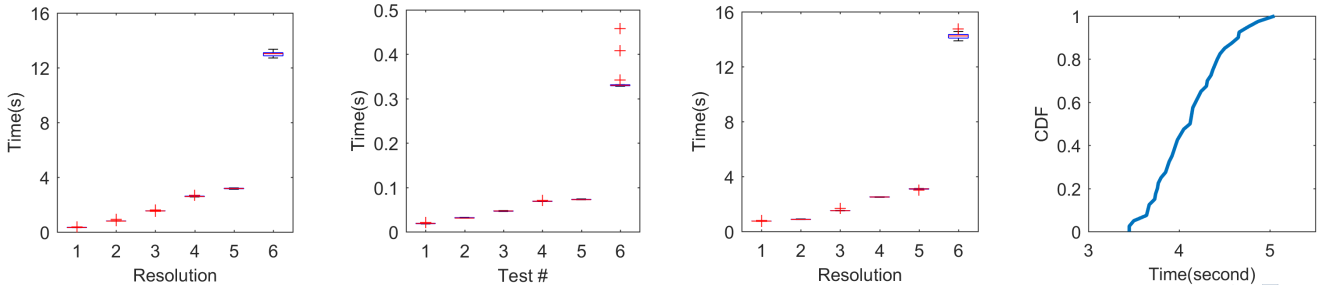
Fig. 15 shows the CDF of the obtained similarity value. We make the following observations. *Image resizing method*: image cropping is much better than all tested down-sampling methods and it has the most distinguishing similarity value in all configurations. We note that image cropping is also the most efficient one. *Image format*: PNG is better than JPG in fingerprint detection. For the matching image pairs, PNG images generate higher PCE values than JPG images. For non-matching image pairs, JPG images generate higher PCE values than PNG images due to the noise components introduced during the lossy compression process. *Scaling ratio*: a higher scaling ratio results in a higher PCE value for PNG images. The scaling ratio has no remarkable impact on JPG images.

To summarize, the best *resizing strategy* is to crop the image to the target resolution and save the obtained image in the PNG format. Comparing the distributions of matching and non-matching image pairs, it can be observed that even images with 40% scaling ratio (16% pixel amount) can achieve a decent accuracy.

5) Impact of the Number of Reference Images: For images with a low scaling ratio, one approach to improve the accuracy

TABLE II. Experimental settings for overall performance evaluation

Test#	1	2	3	4	5	6
Image Resolution	640x480	960x720	1280x960	1600x1200	2048x1152	3264x2448
Probe Resolution	200x200	200x200	400x400	400x400	400x400	800x800



(a) Fingerprint matching

(b) Forgery detection

(c) Total time consumption

(d) Photographing

Fig. 17. Time overhead of the ABC protocol. The resolutions of the tested images are shown in Table II.

of fingerprint detection is to increase the number of reference images uploaded by the user. Since this approach also increases the registration overhead of the authentication system, we further investigate how the number of reference images affects the similarity value of resized images.

Since the high registration overhead can severely degrade user experience, we only tested 1, 2, 4 and 6 reference images. The images are resized with image cropping and saved in both PNG and JPG formats. The image scaling ratios are 40%, 60% and 80%. Each of the 24 configurations is tested with 100 matching image pairs and 100 non-matching image pairs generated from the Samsung image set. Fig. 16 shows the CDF of the obtained similarity values. We observe that: 1) for the JPG format, although increasing the number of reference images can improve the accuracy of fingerprint detection, it is hardly possible for JPG images to achieve fair accuracy with reasonable registration overhead; 2) for the PNG format, even images with a scaling ratio of 40% can achieve high accuracy with a very low registration overhead.

C. Time Overhead

We first analyze the cost of each individual procedure involved in the authentication process and then discuss the overall protocol efficiency. The system is tested with six of the most common resolutions shown in Table II.

Image Content Matching: the cost of this procedure is mainly determined by the version of the applied QR code. Based on the experimental results in [41], smartphones can decode QR codes of a very high version (20) within 0.1 second.

Fingerprint Matching: this process involves two rounds of noise extraction and PCE calculation. The time consumption of this procedure is shown in Fig. 17(a).

Forgery Detection: since the required noise residues have been obtained in the previous procedure, this procedure only involves one round of PCE calculation. Fig. 17(b) shows the time consumption of this process.

Removal Detection: this process involves two rounds of noise extraction and PCE calculation. For the probe signal used in our prototype (800×800), the protocol uses up to 0.9 seconds to detect fingerprint removal.

Overall Protocol Efficiency: For each test, we utilize the *parallel pool* of Matlab with four *workers* on a local machine. Two of the *workers* conduct fingerprint matching and forgery detection sequentially, and the other two *workers* conduct removal detection with the probe signals shown in Table II. As shown in Fig.17(c), for most of the tested common resolutions, ABC achieves high efficiency. Compared with the fingerprint matching process, the security mechanisms integrated in the protocol only introduce 7.5% additional run time to the authentication process.

The latency for high resolution images is mainly caused by the fingerprint extraction process. We note here that the code published by the digital data embedding laboratory [28] does not take advantage of GPU computing and parallel computing. With further optimization, the efficiency would be significantly improved. Moreover, as shown in Sections VI-B4 and VI-B5, images with a low scaling rate can also achieve high accuracy with reasonable registration overhead. Therefore, for smartphone models with high resolution cameras, the verifier can reduce the overhead of the authentication process through cropping the received image to low resolution.

D. Usability Study

To understand the users' behaviors, needs, and attitudes towards the ABC protocol, we conducted a user study with a prototype using two Samsung Galaxy Note 5 devices as the smartphone to be authenticated and the verifier. In the prototype, we use a NFC channel to implement the wireless channel from the smartphone to the verifier. We tested our system on 40 participants (20 males and 20 females) aged from 21 to 54. They were randomly picked from the general public. During the test, we first gave a one-minute introduction to the system. Each participant was then required to conduct the smartphone authentication using our prototype without further guidance. Since people are familiar with photographing with smartphones, all participants were able to easily accomplish the task on their first attempt. Fig. 17(d) shows the CDF of the time taken by each participant in photographing the challenging QR code. 95% of the participants thought that the photographing phase is efficient and comfortable. In particular, 5 female participants pointed out that photographing is better than typing password since remembering passwords places a considerable burden on them. For the NFC transmission phase, 80% of the male participants criticized that the transmission speed of the NFC channel is a little slow while 90% of the female participants thought that the transmission speed is acceptable and the way it transfers data is interesting.

VII. Related Work

Hardware fingerprinting has been actively studied in recent years. Due to manufacturing imperfection, physical sensors introduce systematic distortions on their output. It has been shown that the distortions generated by motion sensors, acoustic sensors, and wireless transmitters are strong enough to fingerprint off-the-shelf smartphones.

Dey *et al.* [21] exploit the imperfection of the accelerometer. They stimulate the sensor with a vibration motor and use machine learning to create the fingerprint. Bojinov *et al.* [5] analyze the calibration error of the accelerometer and verify its effectiveness with a large number of devices. This method requires the user to perform a calibration of the accelerometer. Das *et al.* [19] further investigate combining the features of both accelerometers and gyroscopes to generate more accurate fingerprints. However, their method requires the user to precisely rotate the smartphone with several angles. Moreover, the fingerprints of motion sensors are manipulatable and can be easily eliminated [19], [18].

Acoustic fingerprints can also be used to uniquely identify smartphones. Das *et al.* [17], [16] extract auditory fingerprints from a process of playing and recording audio clips. Zhou *et al.* [42] explore the speaker's frequency response to a specially designed audio input. Chen *et al.* [9] combine the frequency response of one device's speaker and another device's microphone as the hardware fingerprint for device authentication. However, these methods require access to the microphone and lead to privacy concerns [18].

Radio frequency fingerprinting is also an active research area. Several individual steps in the process of generating wireless signals, all due to hardware imperfections of a transmitter [15], can be the source of the RF fingerprints. Different fingerprint sources include the clock jitter [40], device antenna [13], DAC sampling error [34], power amplifier non-linearity [34], [35], [32], modulator sub-circuit [6], and the mixer or local frequency synthesizer [39].

Although hardware fingerprinting has been proved to be effective in tracking smartphones, it is unclear whether these methods can resist an impersonation attack. Since the signal generated by a sensor is manipulatable, most fingerprinting methods are vulnerable against forgery attacks where an adversary tampers with the sensor data intentionally [8], [14].

VIII. Conclusion and future work

In this paper, we explore the idea of utilizing the image sensor's PRNU as a smartphone's unique fingerprint to implement the physical layer device authentication. We find that smartphone cameras demonstrate very strong PRNU. Based on this fact, we design ABC, an attack-resilient, real-time, and user-friendly smartphone authentication protocol that differentiates smartphones through the PRNU of their built-in cameras. The registration of a smartphone's PRNU requires only one image. We implement a prototype of ABC and test it with 16,000 images collected from Amazon Mechanical Turk and our own devices. The experimental results show that ABC can efficiently authenticate users' devices with an error rate less than 0.5% and detect fingerprint forgery attacks with an error rate less than 0.47%. Our user study suggests that the PRNU-based authentication is a promising approach for enhancing smartphone security.

With more and more smartphone manufacturers adopting a dual-camera (rare) system, we plan to investigate how to take advantage of the extra camera and improve the security of ABC as future work. With a dual-camera system, the verifier will be able to identify each smartphone with fingerprints of the two cameras and further increase the difficulty of fingerprint forgery. We will also consider the characteristics of different dual-camera system types: IPhone 7 plus is equipped with a wide-angle camera and a telephoto camera to achieve higher-quality zoom from farther away; Huawei P9 combines two image sensors, one RGB and one monochrome, to enhance the detail of the captured image.

Acknowledgment

We thank the anonymous reviewers for their helpful comments. This research is supported in part by US National Science Foundation under grants CNS-1421903 and CNS-1809000, and by Korea National Research Foundation under grant 2016K1A1A2912757.

References

[1] Image sensor relative size comparison tool. http://cameraimagesensor.com/size/.

[2] Information capacity and versions of the qr code. http://www.qrcode.com/en/about/version.html.

[3] Smartphone life cycles are changing. https://www.statista.com/chart/8348/smartphone-life-cycles-are-changing/.

[4] Ba, Z., and Ren, K. Addressing smartphone-based multi-factor authentication via hardware-rooted technologies. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on* (2017), IEEE, pp. 1910–1914.

[5] Bojinov, H., Michalevsky, Y., Nakibly, G., and Boneh, D. Mobile device identification via sensor fingerprinting. *arXiv preprint arXiv:1408.1416* (2014).

[6] Brik, V., Banerjee, S., Gruteser, M., and Oh, S. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM international conference on Mobile computing and networking* (2008), ACM, pp. 116–127.

[7] Cain, S. C., Hayat, M. M., and Armstrong, E. E. Projection-based image registration in the presence of fixed-pattern noise. *IEEE transactions on image processing 10*, 12 (2001), 1860–1872.

[8] Chen, D., Mao, X., Qin, Z., Wang, W., Li, X.-Y., and Qin, Z. Wireless device authentication using acoustic hardware fingerprints. In *International Conference on Big Data Computing and Communications* (2015), Springer, pp. 193–204.

[9] Chen, D., Mao, X., Qin, Z., Wang, W., Li, X.-Y., and Qin, Z. Wireless device authentication using acoustic hardware fingerprints. In *International Conference on Big Data Computing and Communications* (2015), Springer, pp. 193–204.

[10] Chen, M., Fridrich, J., and Goljan, M. Digital imaging sensor identification (further study). In *Electronic Imaging 2007* (2007), International Society for Optics and Photonics, pp. 65050P–65050P.

[11] Chen, M., Fridrich, J., Goljan, M., and Lukás, J. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security 3*, 1 (2008), 74–90.

[12] Chen, S., Ren, K., Piao, S., Wang, C., Wang, Q., Weng, J., Su, L., and Mohaisen, A. You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on* (2017), IEEE, pp. 183–195.

[13] Danev, B., Heydt-Benjamin, T. S., and Capkun, S. Physical-layer identification of rfid devices. In *Usenix Security Symposium* (2009), pp. 199–214.

[14] Danev, B., Luecken, H., Capkun, S., and El Defrawy, K. Attacks on physical-layer identification. In *Proceedings of the third ACM conference on Wireless network security* (2010), ACM, pp. 89–98.

[15] Danev, B., Zanetti, D., and Capkun, S. On physical-layer identification of wireless devices. *ACM Computing Surveys (CSUR) 45*, 1 (2012), 6.

[16] Das, A., Borisov, N., and Caesar, M. Do you hear what i hear?: Fingerprinting smart devices through embedded acoustic components. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (2014), ACM, pp. 441–452.

[17] Das, A., Borisov, N., and Caesar, M. Fingerprinting smart devices through embedded acoustic components. *arXiv preprint arXiv:1403.3366* (2014).

[18] Das, A., Borisov, N., and Caesar, M. Exploring ways to mitigate sensor-based smartphone fingerprinting. *arXiv preprint arXiv:1503.01874* (2015).

[19] Das, A., Borisov, N., and Caesar, M. Tracking mobile web users through motion sensors: Attacks and defenses. In *Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS)* (2016).

[20] Dey, S., Roy, N., Xu, W., Choudhury, R. R., and Nelakuditi, S. Accelprint: Imperfections of accelerometers make smartphones trackable. In *NDSS* (2014).

[21] Dey, S., Roy, N., Xu, W., Choudhury, R. R., and Nelakuditi, S. Accelprint: Imperfections of accelerometers make smartphones trackable. In *NDSS* (2014).

[22] Dirik, A. E., and Karaküçük, A. Forensic use of photo response non-uniformity of imaging sensors and a counter method. *Optics express 22*, 1 (2014), 470–482.

[23] Fridrich, J. Digital image forensics. *IEEE Signal Processing Magazine 26*, 2 (2009).

[24] Gloe, T., Kirchner, M., Winkler, A., and Böhme, R. Can we trust digital image forensics? In *Proceedings of the 15th ACM international conference on Multimedia* (2007), ACM, pp. 78–86.

[25] Gohshi, S., Nakamura, H., Ito, H., Fujii, R., Suzuki, M., Takai, S., and Tani, Y. A new watermark surviving after re-shooting the images displayed on a screen. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (2005), Springer, pp. 1099–1107.

[26] Goljan, M. Digital camera identification from images–estimating false acceptance probability. In *International Workshop on Digital Watermarking* (2008), Springer, pp. 454–468.

[27] Goljan, M., Fridrich, J., and Chen, M. Defending against fingerprint-copy attack in sensor-based camera identification. *IEEE Transactions on Information Forensics and Security 6*, 1 (2011), 227–236.

[28] Goljan, M., Fridrich, J., and Filler, T. Large scale test of sensor fingerprint camera identification. In *IS&T/SPIE Electronic Imaging* (2009), International Society for Optics and Photonics, pp. 72540I–72540I.

[29] Hao, T., Zhou, R., and Xing, G. Cobra: color barcode streaming for smartphone systems. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (2012), ACM, pp. 85–98.

[30] Index, B. L. Data breach statistics. http://breachlevelindex.com/.

[31] Karaküçük, A., and Dirik, A. E. Adaptive photo-response non-uniformity noise removal against image source attribution. *Digital Investigation 12* (2015), 66–76.

[32] Liu, M.-W., and Doherty, J. F. Specific emitter identification using nonlinear device estimation. In *Sarnoff Symposium, 2008 IEEE* (2008), IEEE, pp. 1–5.

[33] Lukas, J., Fridrich, J., and Goljan, M. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security 1*, 2 (2006), 205–214.

[34] Polak, A. C., Dolatshahi, S., and Goeckel, D. L. Identifying wireless users via transmitter imperfections. *IEEE Journal on Selected Areas in Communications 29*, 7 (2011), 1469–1479.

[35] Polak, A. C., and Goeckel, D. L. Rf fingerprinting of users who actively mask their identities with artificial distortion. In *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on* (2011), IEEE, pp. 270–274.

[36] Quiring, E., and Kirchner, M. Fragile sensor fingerprint camera identification. In *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on* (2015), IEEE, pp. 1–6.

[37] Remley, K., Grosvenor, C., Johnk, R., Novotny, D., Hale, P., McKinley, M., Karygiannis, A., and Antonakakis, E. Electromagnetic signatures of wlan cards and network security. In *Signal Processing and Information Technology, 2005. Proceedings of the Fifth IEEE International Symposium on* (2005), IEEE, pp. 484–488.

[38] Steinebach, M., Liu, H., Fan, P., and Katzenbeisser, S. Cell phone camera ballistics: attacks and countermeasures. *Proc. SPIE, Multimedia on Mobile Devices 2010 7542* (2010), 0B–0C.

[39] Toonstra, J., and Kinsner, W. A radio transmitter fingerprinting system odo-1. In *Electrical and Computer Engineering, 1996. Canadian Conference on* (1996), vol. 1, IEEE, pp. 60–63.

[40] Zanetti, D., Danev, B., et al. Physical-layer identification of uhf rfid tags. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking* (2010), ACM, pp. 353–364.

[41] Zhang, B., Ren, K., Xing, G., Fu, X., and Wang, C. Sbvlc: Secure barcode-based visible light communication for smartphones. *IEEE Transactions on Mobile Computing 15*, 2 (2016), 432–446.

[42] Zhou, Z., Diao, W., Liu, X., and Zhang, K. Acoustic fingerprinting revisited: Generate stable device id stealthily with inaudible sound. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (2014), ACM, pp. 429–440.