A breach into the Authentication with Built-in Camera (ABC) Protocol

Cezara Benegui [1] and Radu Tudor Ionescu [2]

[1]Affiliation not available
[2]University of Bucharest

October 30, 2023

Abstract

[Paper accepted at ACNS 2020]

In this paper, we propose a simple and effective attack on the recently introduced Smartphone Authentication with Built-in Camera Protocol, called ABC. The ABC protocol uses the photo-response non-uniformity (PRNU) as the main authentication factor in combination with anti-forgery detection systems. The ABC protocol interprets the PRNU as a fingerprint of the camera sensor built-in a smartphone device. The protocol works as follows: during the authentication process, the user is challenged with two QR codes (sent by the server) that need to be photographed with a pre-registered device. In each QR code, the server embeds a unique pattern noise (not visible to the naked eye), called probe signal, that is used to identify potential forgeries. The inserted probe signal is very similar to a genuine fingerprint. The photos of QR codes taken by the user are then sent to the server for verification. The server checks (i) if the photos contain the user's camera fingerprint (used to authenticate the pre-registered device) and (ii) if the photos contain the embedded probe signal. If an adversary tries to remove (subtract) his own camera fingerprint and replace it with the victim's camera fingerprint (computed from photos shared on social media), then he will implicitly remove the embedded probe signal and the attack will fail. The ABC protocol is able to detect these attacks with a false acceptance rate (FAR) of 0.5%. However, the ABC protocol wrongly assumes that the attacker can only determine his own camera fingerprint from the photos of the presented QR codes. The attack proposed in our work is able to get past the anti-forgery detection system with a FAR of 54.1%, simply by estimating the attacker's camera fingerprint from a different set of photos (e.g. five photos) owned by the attacker. This set of photos can be trivially obtained before the attack, allowing the adversary to compute his camera fingerprint independently of the attack. The key to the success of our attack is that the independently computed adversary's camera fingerprint does not contain the probe signal embedded in the QR codes. Therefore, when we subtract the adversary's camera fingerprint and add the victim's camera fingerprint, the embedded probe signal will remain in place. For this reason, the proposed attack can successfully pass through the anti-forgery detection system of the ABC protocol. In this paper, we also propose a potential fix based on analyzing signals from built-in motion sensors, which are not typically shared on social media.

A breach into the Authentication with Built-in Camera (ABC) Protocol

Cezara Benegui, Radu Tudor Ionescu

Faculty of Mathematics and Computer Science
University of Bucharest, 14 Academiei, Bucharest, Romania
cezara.benegui@fmi.unibuc.ro, raducu.ionescu@gmail.com

Abstract. In this paper, we propose a simple and effective attack on the recently introduced Smartphone Authentication with Built-in Camera Protocol, called ABC. The ABC protocol uses the photo-response non-uniformity (PRNU) as the main authentication factor in combination with anti-forgery detection systems. The ABC protocol interprets the PRNU as a fingerprint of the camera sensor built-in a smartphone device. The protocol works as follows: during the authentication process, the user is challenged with two QR codes (sent by the server) that need to be photographed with a pre-registered device. In each QR code, the server embeds a unique pattern noise (not visible to the naked eye), called probe signal, that is used to identify potential forgeries. The inserted probe signal is very similar to a genuine fingerprint. The photos of QR codes taken by the user are then sent to the server for verification. The server checks (*i*) if the photos contain the user's camera fingerprint (used to authenticate the pre-registered device) and (*ii*) if the photos contain the embedded probe signal.
If an adversary tries to remove (subtract) his own camera fingerprint and replace it with the victim's camera fingerprint (computed from photos shared on social media), then he will implicitly remove the embedded probe signal and the attack will fail. The ABC protocol is able to detect these attacks with a false acceptance rate (FAR) of 0.5%. However, the ABC protocol wrongly assumes that the attacker can only determine his own camera fingerprint from the photos of the presented QR codes. The attack proposed in our work is able to get past the anti-forgery detection system with a FAR of 54.1%, simply by estimating the attacker's camera fingerprint from a different set of photos (e.g. five photos) owned by the attacker. This set of photos can be trivially obtained before the attack, allowing the adversary to compute his camera fingerprint independently of the attack. The key to the success of our attack is that the independently computed adversary's camera fingerprint does not contain the probe signal embedded in the QR codes. Therefore, when we subtract the adversary's camera fingerprint and add the victim's camera fingerprint, the embedded probe signal will remain in place. For this reason, the proposed attack can successfully pass through the anti-forgery detection system of the ABC protocol. In this paper, we also propose a potential fix based on analyzing signals from built-in motion sensors, which are not typically shared on social media.

Keywords: ABC protocol, PRNU fingerprint, camera fingerprint, impersonation attack, forgery attack, authentication with built-in camera.

1 Introduction

With the rapid growth of the online environments, e.g. social media platforms, in which users generate content on a daily basis using their smartphones, it becomes easier and easier for attackers to gather information about specific individuals. The information collected can be used in different identity forgery attacks, especially impersonation attacks. Since more than half of the smartphone users are using mobile banking services [8], preventing identity forgery attacks is critical. One possible approach to prevent such impersonation attacks (from different devices) is to determine that the user is actually using a known (pre-registered) device. Typical verification protocols are based on sending a confirmation code by SMS [17], tying the user to his mobile device. An alternative approach is the recently-proposed Smartphone Authentication with Built-in Camera Protocol, called ABC [28], which represents the main focus of our work. The ABC protocol uses the photo-response non-uniformity (PRNU) [21] signal as the main authentication factor. The PRNU is a fixed pattern noise specific to a camera sensor, and it can be estimated using different techniques [3, 4, 10, 18, 20, 21]. The ABC protocol is mainly based on interpreting the PRNU as a fingerprint of the camera sensor that is usually built-in any smartphone device.

Associating the information available on the Internet to a potential victim can easily offer attackers access to a set of images (or at least an image) taken by the victim, which in the context of PRNU-based verification, can be used to compute the camera fingerprint of the victim. With this information, the attackers can impersonate the victim and pursue transactions or other fraudulent activities in that person's name. As it is generally well known that the PRNU fingerprint is vulnerable to such forgery attacks [24, 28], the ABC protocol is equipped an anti-forgery detection mechanism. Indeed, the ABC protocol claims to solve the fingerprint forgery problem along with other possible attacks, such as replay attacks, with a total error rate lower than 0.5% [28]. The ABC authentication process consists in a set of steps that require the user to take two photos of two QR codes displayed on a screen and send the photos to a server for verification. The server processes the images and identifies if the content from the QR codes is legitimate, then it verifies the user's camera fingerprint and checks for forgery attacks. The forgery attack detection process scans the received image for a fixed pattern noise (probe signal) included in the two QR codes, a noise that is very similar to a device fingerprint (not visible to the naked eye). In case of an attack, in which the fingerprint of the attacker is replaced with the victim's fingerprint, the forgery detection system detects that the fixed pattern noise added to the initial QR code images is missing [28]. However, the ABC protocol assumes that the attacker computes the camera fingerprint of his own device, using the photos of the presented QR codes, taken during the authentication. As explained below, this assumption is wrong.

We propose a different approach for the attack, in which the attacker (adversary) uses an external set of photos (even a single photo is enough) to compute his own camera fingerprint. Clearly, an external set of photos can be trivially collected by the attacker before performing the attack, independently. Hence, the

adversary's camera fingerprint can also be computed in a completely independent manner from the attack. More importantly, the independently computed adversary's camera fingerprint will no longer contain the fixed pattern noise embedded in the QR codes by the verification system. Therefore, when we subtract the adversary's camera fingerprint and add the victim's camera fingerprint during the attack, the embedded fixed pattern noise will remain in place. For this reason, the proposed attack can successfully pass through the anti-forgery detection system of the ABC protocol. Since our attack requires several changes to the photos sent for verification (subtracting attacker's fingerprint, adding victim's fingerprint), the fixed pattern noise can be deteriorated by these changes. Therefore, our attack succeeds in about 50% of the cases. To estimate the number of successful attempts, we conduct experiments using 630 photos collected from six different smartphone devices. During registration, we use either one or five photos per device to compute the fingerprint of each device. While Zhongjie et al. [28] use one photo during registration, we noticed that our attack has a better success rate when using more photos, e.g. five. Since the attacker can trivially take any number of photos with his own smartphone and the victim is likely to post multiple photos on social media, we believe that using five photos for PRNU estimation is realistic although different from Zhongjie et al. [28]. In the experiments, one by one, each device is considered as being the victim's device in order to be able to simulate attacks. We attack each victim's device with photos from the other devices, using 100 image samples per attacker's device. In total, we perform a set of 3000 attacks, achieving a successful attack rate (false acceptance rate) of 54.1% when using five images for PRNU estimation and a successful attack rate of 47.7% when using one image for PRNU estimation, respectively. Since our attack is successful in about half of the cases, we consider it as a viable threat to the ABC protocol. We thus conclude that the anti-forgery detection system of the ABC protocol needs to be revised. In this paper, we also propose a revised ABC protocol based on using signals captured from built-in motion sensors, which are not typically shared on social media. The false acceptance rate of the revised ABC protocol is 5.3%.

The rest of this paper is organized as follows. Recent related work on authentication protocols and vulnerabilities is presented in Section 2. The ABC protocol and our attack scheme are described in Section 3. Our comparative experiments and results are presented in Section 4. Our revised ABC protocol is described in Section 5. Finally, we draw our conclusions in Section 6.

2 Related Work

Aghili et al. [1] presented attacks for breaking into a lightweight machine-to-machine (M2M) authentication protocol [12] used for communication in Industrial Internet of Things (IIoT) environments. The authors showed that the M2M authentication protocol [12] is vulnerable to Denial-of-Service (DoS) and router impersonation attacks. In a different work, Aghili et al. [2] showed that the untraceable and anonymous three-factor authentication scheme [6] for Het-

erogeneous Wireless Sensor Networks is vulnerable to user impersonation, desynchronization and traceability attacks. Aghili et al. [2] also proposed an improved protocol that is resilient to these kinds of attacks.

To our knowledge, there are no previous works that study attacks for PRNU-based authentication protocols using the built-in camera of smartphone devices. However, there are previous works that study the implementation of PRNU-based fingerprinting methods as a single authentication protocol [28] or as a component in a multifactor authentication scheme [27]. Different from the approach studied by Zhonjie et al. [28], which implemented the camera's fingerprint as the main component of an authentication protocol, Valsesia et al. [27] employed the PRNU of the built-in camera as a weak physical unclonable function (PUF) [16] in a multifactor authentication scheme. Moreover, there are other works that use multiple device sensor fingerprints, including PRNU, and combine them with machine learning, to build strong authentication systems [5].

In this section, we provide a brief overview of commonly used smartphone authentication approaches and some of their vulnerabilities. The most common approach used in the recent user authentication systems is to employ a multifactor scheme. Systems based on multifactor authentication are composed of a known secret, which is usually a password, that is complemented by one or more hardware or software tokens [9]. One of the most commonly used tokens is the One-Time Password [22], which consists of a token that is sent to the user via e-mail or SMS, in order to better assess the possession of a hardware or software element which identifies the user. Using the PRNU as an authentication system or as a component in a multifactor scheme requires additional security measures. Considering that PRNU fingerprints are vulnerable to forgery attacks [13, 15], it is not a secure option to rely on PRNU fingerprint authentication alone. Hence, along with a fingerprint matching technique, other systems such as forgery detection must be implemented [28].

In our paper, we study the vulnerability of the ABC Protocol [28], presenting a simple attack scheme that showcases the weakness of the ABC protocol against forgery attacks and adversary fingerprint removal attacks. We also propose a revised ABC protocol that is based on multi-factor authentication, i.e. it considers the signals captured by the built-in motion sensors, e.g. accelerometer or gyroscope, along with the images captured by the built-in camera.

3 Method

In this section, we present in detail the ABC protocol [28] and the protection methods implemented in this protocol. We then explain in detail our impersonation attack scheme that is able to bypass the ABC protocol.

3.1 ABC Protocol

The ABC protocol [28] is composed of two main phases: **registration** and **authentication**. In the **registration** phase, the user sends a sample image $I_{(r)}$

(taken with the smartphone camera) to the server (verifier) that implements the ABC protocol. The process does not impose any constraint on the reference image $I_{(r)}$. The image is used to register the smartphone device into the system. More exactly, after the image is received, the sever extracts the PRNU fingerprint $\hat{K}_{(c)}$ from the image $I_{(r)}$ and builds a user profile for the specific device. As Zhongjie et al. [28], we use the notation $\hat{K}_{(c)}$ to denote an accurate estimation of the actual PRNU fingerprint $K_{(c)}$. Once the user's device is registered into the system, we can perform one or more authentications.

The **authentication** process is composed of three main steps: Quick Response (QR) codes generation by the server, pictures upload by the user and pictures verification by the server.

In the **first authentication step** (i), in which the QR codes are generated, the system embeds information about the transaction in progress. The transaction details are accompanied by a timestamp T_i and a random string str_i. Along with this information, the QR code images also embed a non-related white Gaussian noise Γ_i, called probe signal, with a variance equal to 5. In this step, the verifier generates two images defined as:

$$I_{i(s)} = QR(str_i, T_i) + \Gamma_i, \forall i \in \{1, 2\}, \tag{1}$$

which are displayed on a screen to the user.

In the **second authentication step** (ii), the user captures the above images $I_{1(s)}$ and $I_{2(s)}$ with the registered smartphone's built-in camera, and sends the captured images securely back to the server. The captured images, denoted by $I_{i(c)}$, should contain a noise residue $W_{i(c)}$ composed of the PRNU fingerprint of the user and the probe signal Γ_i:

$$I_{i(c)} = QR(str_i, T_i) + W_{i(c)}, \forall i \in \{1, 2\}, \tag{2}$$

where the noise residue is formally defined as follows:

$$W_{i(c)} = \Gamma_i + K_{(c)}. \tag{3}$$

We note that $K_{(c)}$ is present in Equation (3) only if the authentication is performed by the registered user. Otherwise, the noise residue will contain an adversary's camera fingerprint $K_{(a)}$ instead of $K_{(c)}$.

The **third authentication step** (iii) of the protocol is composed of multiple sub-steps: verification of the presented QR codes, fingerprint verification, forgery detection and probe signal verification. The verification of the received QR codes step checks the content of the QR codes to match with the ones generated in the first authentication step (i). Then, the verifier detects if the images $I_{1(c)}$ and $I_{2(c)}$ captured during the second authentication step (ii) contain the same fingerprint $\hat{K}_{(c)}$ as the reference image $I_{(r)}$ provided in the registration stage. Proceeding forward, the forgery detection system tries to identify whether an adversary's camera fingerprint $\hat{K}_{(a)}$ is present in the analyzed image. If an adversary's fingerprint $\hat{K}_{(a)}$ is detected, the system rejects the transaction. In the last sub-step, the protocol verifies if the probe signal Γ_i is present. If the unique pattern noise was removed (subtracted) in the forgery process, or in a counterfeit attempt, then the system rejects the transaction.

3.2 ABC Protocol Defense Systems

Forgery detection: The anti-forgery detection system implemented in the ABC protocol [28] protects the system from forged images in which an adversary's fingerprint, $K_{(a)}$, might be present. For each of the two images received by the verifier, the noise residue $W_{i(c)}$ is extracted. Next, the noise residue from the first received image is compared with both the noise residue from the second image and the noise residue (PRNU fingerprint) of the image sample provided by the user during registration. The similarity value between the analyzed noise residues $W_{1(c)}$ and $W_{2(c)}$ is given by:

$$PCE(W_{1(c)}, W_{2(c)}), \tag{4}$$

while the similarity value between the analyzed noise residue $W_{1(c)}$ and the registered PRNU fingerprint $\hat{K}_{(c)}$ is given by:

$$PCE(W_{1(c)}, \hat{K}_{(c)}), \tag{5}$$

where PCE is the Peak to Correlation Energy [14].

If the images captured during authentication are forged, they should contain the attacker's fingerprint along with the victim's fingerprint. Therefore, the similarity between $W_{1(c)}$ and $W_{2(c)}$ is higher in comparison with the similarity of the noise residue $W_{1(c)}$ and the registered PRNU fingerprint $\hat{K}_{(c)}$, i.e.:

$$PCE(W_{1(c)}, W_{2(c)}) > PCE(W_{1(c)}, \hat{K}_{(c)}) + t_1, \tag{6}$$

where t_1 is a pre-established threshold. As noted in [28], the forgery detection system can be bypassed if the adversary removes his own PRNU fingerprint $K_{(a)}$ and replaces it with the victim's PRNU fingerprint $K_{(c)}$. The removal detection system proposed by Zhongjie et al. [28] and described below is used to prevent this situation.

Removal detection: During the authentication procedure, the verifier sends two images that contain a probe signal to the user. When the system receives the verification photos back from the user, subsamples of the received images $I_{i(c)}$ are extracted, obtaining a larger set of images $\hat{I}_{i(c)}$ in which the presence of the unique pattern noise Γ_i is verified. If the captured images $I_{i(c)}$ are forged, the similarity value between the known probe signal Γ_i and the noise residue $W_{i(c)}$ should be substantially lower, falling below a precisely chosen threshold t_2:

$$PCE(W_{i(c)}, \Gamma_i) < t_2. \tag{7}$$

If Equation (7) holds, then the transaction is rejected. We note that Equation (7) is based on the supposition that the adversary estimates his own PRNU fingerprint from the captured images $I_{i(c)}$. In this case, the estimated PRNU fingerprint $\hat{K}_{(a)}$ will contain the probe signal Γ_i. Consequently, removing the PRNU fingerprint $\hat{K}_{(a)}$ in order to pass forgery detection will implicitly remove the probe signal. In this case, the attack is successfully stopped by the removal detection system. However, as we are about to discuss in detail next, the ABC protocol does not consider the trivial case in which the adversary estimates his

own PRNU fingerprint from a different set of images than those photographed during the authentication. We exploit this vulnerability in our attack described below.

3.3 Proposed Attack Scheme

While the ABC Protocol assumes that the adversary computes his camera fingerprint using the photos $I_{i(c)}$ captured during the authentication phase, we propose a different approach for the attack, in which the adversary uses a pre-computed camera fingerprint $\hat{K}_{(a)}$, obtained from an external set of photos $I_{j(x)}$, captured with the same device used for the attack. In our experiments described in Section 4, we used either one or five images, i.e. $j \in \{1\}$ or $j \in \{1, 2, 3, 4, 5\}$. While the decision to use one image during registration is motivated by the fact that Zhongjie et al. [28] do the same, the decision to use five images is motivated by two facts: the attacker can easily take several images with his smartphone and the victim is likely to post multiple images on social media. We thus believe that it is realistic to consider that the attacker might use five images to compute his PRNU fingerprint $\hat{K}_{(a)}$ and another five images from social media to compute the victim's PRNU fingerprint $\hat{K}_{(c)}$. We empirically observed that using five images instead of one during registration increases the success rate of our attack.

When the verifier generates the two verification images defined as in Equation (1), the attacker takes pictures of those images in order to send them back to the verifier. In this step, the images taken by the attacker are defined as follows:

$$I_{i(c)} = QR(str_i, T_i) + \Gamma_i + K_{(a)}, \forall i \in \{1, 2\}. \tag{8}$$

We note that Equation (8) is similar to Equation (2), the only difference being that the captured image contains the PRNU fingerprint $K_{(a)}$ of the attacker instead of the PRNU fingerprint $K_{(c)}$ of the victim. In order to perform the attack, we aim to remove $\hat{K}_{(a)}$ and replace it with $\hat{K}_{(c)}$, assuming (as Zhongjie et al. [28]) that the attacker has access to a very small set of photos (or at least a photo), e.g. shared on social media, that belong to the victim, which allows the attacker to estimate the victim's PRNU fingerprint denoted by $\hat{K}_{(c)}$. At this stage, Zhongjie et al. [28] assume that the attacker estimates the PRNU fingerprint $\hat{K}_{(a)}$ using the images $I_{i(c)}$ defined in Equation (8), thus including the probe signal Γ_i into the estimation. Hence, the attempt to remove $\hat{K}_{(a)}$ will also remove Γ_i. Since we compute the adversary's camera fingerprint $\hat{K}_{(a)}$ on an independent set of images $I_{j(x)}$, removing $\hat{K}_{(a)}$ from the captured images $I_{i(c)}$ does not imply the removal of the probe signal Γ_i. Hence, the attacker can proceed with the forgery by subtracting the estimated PRNU fingerprint $\hat{K}_{(a)}$ and by adding the victim's fingerprint $\hat{K}_{(c)}$ to the captured images $I_{i(c)}$, resulting in a set of forged images defined by:

$$I_{i(f)} = I_{i(c)} - \hat{K}_{(a)} + \hat{K}_{(c)}, \forall i \in \{1, 2\}. \tag{9}$$

By replacing $I_{i(c)}$ in Equation (9), we obtain:

$$I_{i(f)} = QR(str_i, T_i) + \Gamma_i + K_{(a)} - \hat{K}_{(a)} + \hat{K}_{(c)}, \forall i \in \{1, 2\}. \tag{10}$$

We note that $\hat{K}_{(a)}$ and $\hat{K}_{(c)}$ are estimated values of actual PRNU fingerprints of the attacker's and the victim's smartphone built-in cameras, respectively. Through the operations performed in Equation (9), the probe signal Γ_i can be affected to some small extent. Therefore, the forged images $I_{i(f)}$ are only approximately equal to the results desired by the attacker:

$$I_{i(f)} \approx QR(str_i, T_i) + \Gamma_i + \hat{K}_{(c)}, \forall i \in \{1, 2\}. \tag{11}$$

The forged images $I_{i(f)}$, which contain the victim's fingerprint, are sent back to the verifier, easily passing the fingerprint verification process. Then, the forgery detection and removal detection algorithms process the images received by the verifier. The forgery detection algorithm processes the images and computes the similarity values defined in Equations (4) and (5). Then, the verifier applies Equation (6) to determine if the images are forged. Since the forged images do not contain the attacker's fingerprint $K_{(a)}$, the similarity values defined in Equations (4) and (5) are roughly equal. Thus, our attack can bypass the forgery detection system.

Since we compute the adversary's camera fingerprint using an external set of images, in the process of removing the attacker's fingerprint $\hat{K}_{(a)}$ and adding the victim's fingerprint $\hat{K}_{(c)}$, the value of the probe signal Γ_i is only slightly altered, but still present in the forged images. When the removal detection algorithm checks for the presence of the probe signal Γ_i using Equation (7) against a predefined threshold, the algorithm will find that Γ_i is included in the received images. Therefore, our attack can bypass the removal detection system.

With the proposed attack scheme, we can bypass both protection systems of the ABC protocol. Due to the approximation errors involved in the forgery process, the attack only succeeds in about one in every two cases, as detailed in the following experiments.

4 Experiments

4.1 Data Set

In order to test our attack scheme and estimate the number of successful attempts in which the ABC protocol fails to detect our attack, we collect our own data set of images. The data set consists of 630 images gathered from six different smartphone devices: two iPhone X, two Samsung S8, one Huawei P20 Lite and one Huawei P10 Lite. We select the first 1000×750 pixels to compute the PRNU fingerprints, as recommended in previous works [24, 28]. For each device we collect a number of 105 photos.

In the first set of experiments, we use the first five images to compute the reference PRNU fingerprint of each device, which leaves 100 images to perform authentications on the same device (simulating the actions of a registered user) or attacks on the other devices (simulating the actions of an impersonator). In total, we perform 600 authentications (100 per device) and, considering all possible combinations of device pairs, 3000 attacks (500 per device). The justification for using five images during registration is given by two facts: (1) the attacker can

take any number of photos on his device and (2) the victim is likely to post at least five photos on social media platforms.

In the second set of experiments, we use only the first image to compute the reference PRNU fingerprint, as the method [21] used for PRNU estimation can be applied on a single image and this is how Zhongjie et al. [28] conduct their experiments. Our second set of experiments are aimed at demonstrating that our attack can defeat the ABC protocol in the same setting as Zhongjie et al. [28]. As in the first set of experiments, we use the last 100 images to perform authentications on the same device or attacks on the other devices, resulting in the same number of total authentications (600) and attacks (3000).

4.2 Evaluation Details

Evaluation Measures: We report the number of successful attacks (false acceptances) as well as the False Acceptance Rate (FAR), which is typically defined as the ratio of the number of false acceptances divided by the number of authentication attempts. A *false acceptance* is an instance of a security system, in our case the ABC protocol, incorrectly verifying an unauthorized person, e.g. an impersonator. We note that our attack does impact the False Rejection Rate (FRR) of the ABC protocol, i.e. the FRR is similar to that reported in [28]. Therefore, we focus only on reporting the FAR.

Evaluation Protocol: The main goal of the experiments is to validate the attack scheme proposed in this paper. While reporting the FAR values for our attack is necessary, we also have to validate that the forgery detection (FD) system and the removal detection (RD) system of the ABC protocol work properly. For this reason, we need to perform attacks as described in [28]. Our aim is to show that the protection systems of the ABC protocol are indeed able to reject the attacks specified in [28], while not being able to detect our own attack.

One by one, each of the n smartphone devices is considered as being the victim's device. In order to perform attacks, the remaining $n - 1$ devices are considered to belong to adversaries. Each adversary performs 100 attacks. Given that our data set consists of $n = 6$ devices, we obtain a number of 3000 ($6 \times 5 \times 100$) attacks. For each attack, we determine if it passes undetected by the Forgery Detection system and by the Removal Detection system. We consider a successful attack only if it succeeds to cross both Forgery Detection and Removal Detection systems. We count the number of successful attacks and compute the corresponding FAR at different PCE thresholds between 10000 and 50000, using a step of 100. We note that the threshold values are generally higher than those used in [28], because we compute the PRNU fingerprints on larger images. We determine the *optimal threshold* as the threshold that provides a FAR of roughly 0.5% for the attack scheme detailed in [28], because Zhongjie et al. [28] report a FAR of 0.5% in their paper. We note that they selected the threshold that corresponds to equal FAR and FRR.

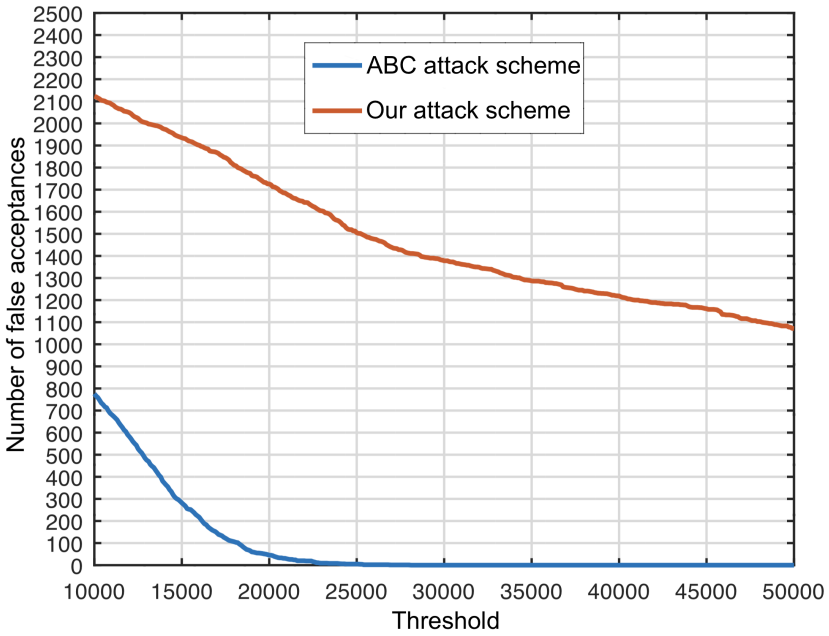

Fig. 1. Number of false acceptances (on the vertical axis) bypassing both Forgery Detection and Removal Detection systems, for the attack scheme proposed in our paper versus the attack scheme detailed in [28], when five images are used for PRNU estimation. False acceptances are counted for multiple PCE thresholds (on the horizontal axis) between 10000 and 50000, with a step of 100. Best viewed in color.

4.3 Results Using Five Images for PRNU Estimation

Figure 1 illustrates the number of false acceptances for the proposed attack scheme versus the attack scheme considered by Zhongjie et al. [28], using five images for PRNU estimation and multiple PCE thresholds between 10000 and 50000. Threshold values are taken at a step of 100. The false acceptance counts represent attacks that bypass both Forgery Detection and Removal Detection systems of the ABC protocol. Zhongjie et al. [28] reported a FAR of 0.5% for their attack scheme. In our case, we obtain a similar FAR for their attack when the PCE threshold is set to 22500. We thus select this value as the optimal threshold. We note that for each and every threshold between 10000 and 50000, our attack scheme provides significantly more successful attempts.

We present the number of false acceptances for the proposed attack scheme versus the attack scheme considered by Zhongjie et al. [28] in Table 1, using five different PCE thresholds between 10000 and 50000, additionally including results for the optimal threshold (22500). For the optimal threshold, there are 1624 successful attacks from the total of 3000 attacks. Hence, we conclude that more than half of the attacks are successful, rendering the ABC protocol unsafe in scenarios where an impersonator could gain access to the victim's photos.

Table 1. Number of successful attempts (false acceptances) for the attack scheme proposed in our paper versus the attack scheme detailed in [28], when five images are used for PRNU estimation. Successful attempts are counted for five PCE thresholds between 10000 and 50000. Results (highlighted in bold) for the optimal PCE threshold (22500) are also included. For each attack scheme, we report the number of false acceptances for the Forgery Detection (FD) system, the Removal Detection (RD) system, and both (FD+RD).

Threshold	Proposed attack			ABC attack		
	FD bypass count	RD bypass count	FD+RD bypass count	FD bypass count	RD bypass count	FD+RD bypass count
10000	2235	2701	2122	2628	920	776
20000	1879	2517	1726	2487	73	45
22500	**1800**	**2451**	**1624**	**2446**	**31**	**16**
30000	1600	2292	1378	2350	0	0
40000	1444	2184	1219	2234	0	0
50000	1315	2090	1071	2150	0	0

Table 2. False acceptance rates (FAR) for the attack scheme proposed in our paper versus the attack scheme detailed in [28], when five images are used for PRNU estimation. False acceptance rates are computed for five PCE thresholds between 10000 and 50000. Results (highlighted in bold) for the optimal PCE threshold (22500) are also included. For each attack scheme, we report the false acceptance rates for the Forgery Detection (FD) system, the Removal Detection (RD) system, and both (FD+RD).

Threshold	Proposed attack			ABC attack		
	FD FAR	RD FAR	FD+RD FAR	FD FAR	RD FAR	FD+RD FAR
10000	74.5%	90.0%	70.7%	87.6%	30.7%	25.9%
20000	62.6%	83.9%	57.5%	82.9%	2.4%	1.5%
22500	**60.0**%	**81.7**%	**54.1**%	**81.5**%	**1.0**%	**0.5**%
30000	53.3%	76.4%	45.9%	78.3%	0.0%	0.0%
40000	48.1%	72.8%	40.6%	74.5%	0.0%	0.0%
50000	43.8%	69.7%	35.7%	71.7%	0.0%	0.0%

In Table 2, we provide the false acceptance rates for the proposed attack scheme versus the attack scheme considered by Zhongjie et al. [28]. The values essentially correspond to those presented in Table 1, each number being divided by the total number of attacks (3000). Based on the results presented in Table 2, we conclude that our attack can bypass the Forgery Detection and the Removal Detection systems with a very high FAR (54.1%) at a PCE threshold of 22500. We note that, at the same threshold, the ABC protocol achieves a FAR of 0.5% for the attack scheme described in [28]. In the same time, we computed the False Rejection Rate (FRR) for the ABC protocol, using 600 authentications. At the respective threshold, the FRR is under 0.1%, further proving that the results are consistent with the numbers reported in [28].

We observe that the Forgery Detection and Removal Detection systems perform very well, but only for the attack scheme assumed by the ABC protocol [28]. Considering that in the respective attack scheme the adversary's camera fingerprint is computed from the QR code images which include the probe signal Γ_i, the value Γ_i is removed along with the attacker's fingerprint $\hat{K}_{(a)}$. This leads to a better performance of the Removal Detection system. For instance, the attack scheme considered in [28] is identified by the Removal Detection system with a FAR of 1% at a PCE threshold of 22500.

In our attack scheme, the adversary computes his own PRNU fingerprint by using an external set of five images. Due to the fact that the fingerprint is computed without including the probe signal Γ_i, when the adversary's fingerprint $\hat{K}_{(a)}$ removal (subtraction) and victim's fingerprint $\hat{K}_{(c)}$ addition occurs, the probe signal is mostly unaffected. In this case, the Removal Detection system is not able to identify our attack. For instance, our attack scheme is identified by the Removal Detection system with a FAR of 81.7% at a PCE threshold of 22500.

While our attack can bypass the Removal Detection system with a much higher FAR than the attack scheme considered in [28], it gives slightly lower FAR values in trying to bypass the Forgery Detection system, because the attacker's PRNU fingerprint is computed on a different set of images than the two QR code images used during the authentication. In other words, the lower FAR rates are generated by the approximation errors between the PRNU estimation $\hat{K}_{(a)}$ and the actual PRNU fingerprint $K_{(a)}$ in the QR code images. Nevertheless, our attack scheme achieves a much higher false acceptance rate even when the two protection systems, Forgery Detection and Removal Detection, are considered together.

Overall, the results presented in Table 1 and 2 prove that the ABC Protocol is vulnerable to our attack scheme, since about one in every two attacks succeeds.

4.4 Results Using One Image for PRNU Estimation

Figure 2 illustrates the number of false acceptances for the proposed attack scheme versus the attack scheme considered by Zhongjie et al. [28], using one image for PRNU estimation and multiple PCE thresholds between 10000 and 50000. Threshold values are taken at a step of 100. It is important to mention that in this setting, both the victim's and the adversary's PRNU are estimated from single images. Therefore, this setting is slightly more difficult and the attack is less likely to succeed. Although using five images during registration (as in the previous setting) is realistic, we consider the setting with one image for an apples to apples comparison with Zhongjie et al. [28].

Comparing the results presented in Figure 1 with those presented in Figure 2, we observe that the number of attacks that bypass the ABC protocol is typically lower when one image is used for PRNU estimation instead of five images. However, there are still enough successful attacks to pose a real problem for the ABC protocol. At the optimal threshold (22500), the number of successful attacks is 1430, which translates to a FAR of 47.7% with respect to the total number of

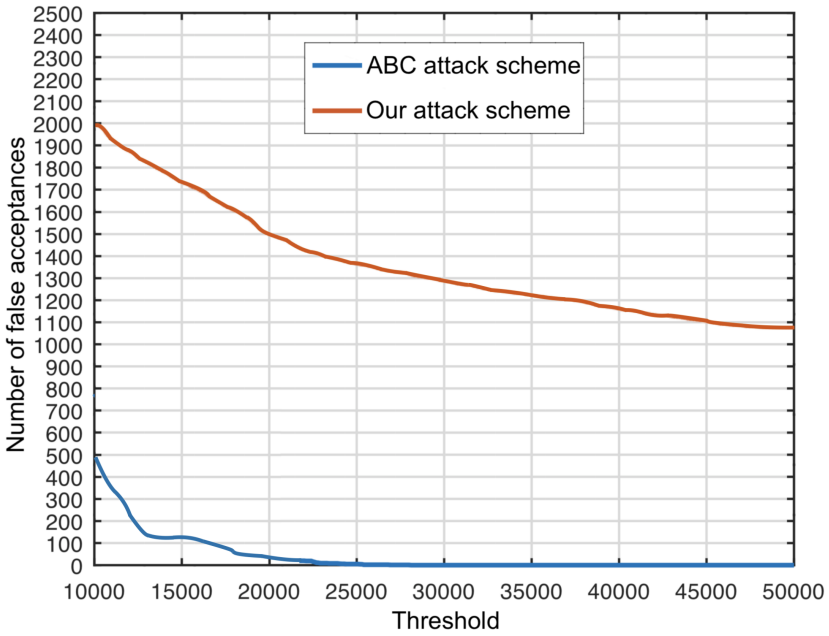

Fig. 2. Number of false acceptances (on the vertical axis) bypassing both Forgery Detection and Removal Detection systems, for the attack scheme proposed in our paper versus the attack scheme detailed in [28], when one image is used for PRNU estimation. False acceptances are counted for multiple PCE thresholds (on the horizontal axis) between 10000 and 50000, with a step of 100. Best viewed in color.

attacks (3000). These empirical results demonstrate that the ABC protocol can still be easily bypassed, even when we use a single image for estimating the PRNU.

Considering both experimental settings (with one or five images for PRNU estimation) and the corresponding results, we conclude that the Removal Detection and Forgery Detection systems of the ABC protocol need to be revised to prevent the attack scheme exposed in our work.

5 Discussion

In this section, we propose a revised ABC protocol that relies on additional built-in sensors, e.g. the accelerometer and/or the gyroscope. We note that motion sensors contain similar fabrication defects as the camera [19], deeming them recognizable based on the captured signals. The main advantage compared to the camera sensor is that motion signals, unlike photographs, are not typically shared on social media by people. Therefore, attackers cannot easily get their hands on these signals. The only disadvantage of our augmented ABC protocol is that it only works on devices that are equipped with motion sensors. However, most smartphones available nowadays do have built-in motion sensors.

In addition to the PRNU fingerprint check, we employ a machine learning system that classifies an authentication session as legitimate or not based on the signals recorded by the built-in motion sensors, following the approach described in [25]. The user is not required to perform any additional steps during authentication, we just have to record the motion signals while the user is pressing the button to take photos.

5.1 Data Set

In order to validate our revised protocol, we select a subset of motion signals recorded on six devices from the data set provided by Sitova et al. [26]. We record motion sensor values during screen taps (e.g. when the user taps the button to take a photo) for 1.5 seconds, starting the recording with 0.5 seconds before the tap event. The accelerometer and the gyroscope each provide 3-axis values at about 100 Hz. For each tap event, we thus have six signals (two sensors $\times$ three axes) composed of 150 discrete values (1.5 seconds at 100 Hz). For each of the six devices, we collect motion signals for 105 tap events. Each tap event is matched with one and only one of the photos used in the experiments presented in Section 4. The image and motion signal data sets are mixed and formatted in a way that simulates a realistic scenario, as if the motion signals are recorded during authentication with the ABC protocol.

5.2 Model

Since the signals recorded by the motion sensors contain noise and large variations, a machine learning model will not be able to learn invariant features from raw signal values. In order to obtain invariant features for each signal, we follow the approach proposed by Shen et al. [25], which is based on extracting a set of statistical features such as: the minimum value, the maximum value, the mean, the variance, the skewness (the orientation of the peak), the kurtosis (the width of the peak) and the quantiles (from 30% to 80%, with a step of 10%). The feature vector corresponding to a tap event is thus composed of 72 statistical features (six signals $\times$ 12 features). We take the feature vectors corresponding to the first five tap events and use them to train a Support Vector Machines (SVM) classifier [11] based on the Radial Basis Function (RBF) kernel. During optimization, the SVM finds a hyperplane that separates the training samples by a maximum margin. We use the SVM implementation from Scikit-learn [23], setting the regularization parameter $C = 100$ and leaving the RBF parameter γ to the default value (scale). Our motion-based verification system authorizes or rejects sessions based on the positive or negative labels provided by the SVM.

5.3 Results

We conduct experiments to show how our motion-based verification system performs by itself and in conjunction with the ABC protocol. When combining

Table 3. False acceptance rates (FAR) and false rejection rates (FRR) for the attack scheme proposed in our paper, when five images and/or motion signals are used during registration. For the ABC protocol, the FAR and the FRR measures are computed for the optimal PCE threshold (22500). For the motion-based verification system, the FAR and the FRR measures are computed for the SVM regularization parameter $C = 100$. We report results for the individual as well as the combined systems (i.e. for the revised ABC protocol).

Authentication system	FAR	FRR
ABC protocol	54.1%	0.1%
Motion-based verification	12.4%	11.0%
Revised ABC protocol (ABC + motion-based verification)	5.3%	11.0%

the ABC system with the motion-based verification system, a session must be validated by both systems, i.e. we use the AND operator. This reduces the FAR, but increases the FRR. The corresponding results are presented in Table 3.

First, we notice that our motion-based verification system alone attains a FAR of 12.4% and a FRR of 11.0%. Although the motion-based verification system is able to withstand the attacks better than the ABC protocol, it has a much higher FRR. The higher FRR can be caused by several factors: the number of training samples (five) might not be sufficient to learn a good SVM model, the chosen model (SVM based on statistical features) might not be the right choice to capture the defects of motion sensors, the task of recognizing motion sensors based on defects observed in output signals might simply be harder than the task of recognizing camera fingerprints. We leave the search for an explanation in this regard for future work.

By combining the ABC protocol with the motion-based verification system, we obtain the revised ABC protocol, which relies on multi-factor (images and motion signals) authentication. The revised protocol attains a lower FAR (5.3%), since attacks have to bypass both the ABC protocol and the motion-based verification system. However, the FRR stays at the same level as for the motion-based verification system (11.0%).

We note that the revised ABC protocol is able to reduce the FAR from 54.1% to 5.3%. However, we consider that the FAR and the FRR values of the revised ABC protocol are still higher than acceptable. In future work, we aim to improve or completely replace the motion-based verification system in order to further reduce the FAR and the FRR values to acceptable thresholds, e.g. below 1%. A better solution might require more than five training samples and end-to-end training, e.g. by employing deep neural networks [7].

6 Conclusion

In this paper, we have presented a simple and effective attack for the ABC protocol [28]. Our strategy is based on computing the adversary's PRNU fingerprint on an external set of samples, which do not include the fixed probe signal

used by the verifier to detect PRNU fingerprint removal. This allowed us to remove the adversary's fingerprint while preserving the probe signal, which led to successful attempts in bypassing the Removal Detection and Forgery Detection systems of the ABC protocol. We have conducted experiments on six mobile devices, performing 3000 attacks, in order to provide an empirical proof and validation of our attack. Our attack scheme provides a FAR of 54.1%, demonstrating that the ABC protocol is not entirely secure. We thus conclude that the ABC protocol is not suited as an authentication measure for high-risk applications, such as applications where financial transactions are involved.

We also took important steps towards revising the ABC protocol. By analyzing and verifying the authenticity of signals recorded by built-in motion sensors, we were able to reduce the FAR from 54.1% to 5.3% when the protocol is exposed to our attack. In future work, we aim to identify other solutions to further reduce the FAR and the FRR values, since we believe that the ABC protocol still has enough potential to become a reliable authentication protocol.

References

1. Aghili, S.F., Mala, H.: Breaking a Lightweight M2M Authentication Protocol for Communications in IIoT Environment. IACR Cryptology ePrint Archive **2018**, 891 (2018)
2. Aghili, S.F., Mala, H., Peris-Lopez, P.: Securing Heterogeneous Wireless Sensor Networks: Breaking and Fixing a Three-Factor Authentication Protocol. Sensors **18**(11), 3663 (2018)
3. Akshatha, K., Karunakar, A., Anitha, H., Raghavendra, U., Shetty, D.: Digital camera identification using PRNU: A feature based approach. Digital Investigation **19**, 69–77 (2016)
4. Altinisik, E., Tasdemir, K., Sencar, H.T.: Extracting PRNU Noise from H.264 Coded Videos. In: Proceedings of European Signal Processing Conference (EUSIPCO). pp. 1367–1371 (2018)
5. Amerini, I., Bestagini, P., Bondi, L., Caldelli, R., Casini, M., Tubaro, S.: Robust smartphone fingerprint by mixing device sensors features for mobile strong authentication. In: Media Watermarking, Security, and Forensics. pp. 1–8. Ingenta (2016)
6. Amin, R., Islam, S.H., Kumar, N., Choo, K.K.R.: An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks. Journal of Network and Computer Applications **104**, 133–144 (2018)
7. Benegui, C., Ionescu, R.T.: Convolutional Neural Networks for User Identification based on Motion Sensors Represented as Images. arXiv preprint arXiv:1912.03760 (2019)
8. Board of Governors of the Federal Reserve System: Consumers and mobile financial services 2016. `https://www.federalreserve.gov/econresdata/consumers-and-mobile-financial-services-report-201603.pdf` (2016), accessed: 2019-04-01
9. Burr, W., Dodson, D., Polk, W.: Electronic authentication guideline. Tech. rep., National Institute of Standards and Technology (2004)
10. Cooper, A.J.: Improved photo response non-uniformity (PRNU) based source camera identification. Forensic Science International **226**(1-3), 132–141 (2013)

11. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning **20**(3), 273–297 (1995)
12. Esfahani, A., Mantas, G., Matischek, R., Saghezchi, F.B., Rodriguez, J., Bicaku, A., Maksuti, S., Tauber, M., Schmittner, C., Bastos, J.: A lightweight authentication mechanism for M2M communications in industrial IoT environment. IEEE Internet of Things Journal **6**(1), 288–296 (2019)
13. Gloe, T., Kirchner, M., Winkler, A., Böhme, R.: Can we trust digital image forensics? In: Proceedings of the ACM International Conference on Multimedia (ACMMM). pp. 78–86. ACM (2007)
14. Goljan, M.: Digital camera identification from images – estimating false acceptance probability. In: Proceedings of the International Workshop on Digital Watermarking (IWDW). pp. 454–468 (2008)
15. Goljan, M., Fridrich, J., Chen, M.: Defending against fingerprint-copy attack in sensor-based camera identification. IEEE Transactions on Information Forensics and Security **6**(1), 227–236 (2011)
16. Herder, C., Yu, M.M., Koushanfar, F., Devadas, S.: Physical Unclonable Functions and Applications: A Tutorial. Proceedings of the IEEE **102**(8), 1126–1141 (2014)
17. Jurcut, A.D., Liyanage, M., Chen, J., Gyorodi, C., He, J.: On the security verification of a Short Message Service protocol. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC). pp. 1–6 (2018)
18. Kang, X., Li, Y., Qu, Z., Huang, J.: Enhancing source camera identification performance with a camera reference phase sensor pattern noise. IEEE Transactions on Information Forensics and Security **7**(2), 393–402 (2012)
19. Khanna, N., Mikkilineni, A.K., Martone, A.F., Ali, G.N., Chiu, G.T.C., Allebach, J.P., Delp, E.J.: A survey of forensic characterization methods for physical devices. Digital Investigation **3**, 17–28 (2006)
20. Li, C.T.: Source camera identification using enhanced sensor pattern noise. IEEE Transactions on Information Forensics and Security **5**(2), 280–287 (2010)
21. Lukáš, J., Fridrich, J., Goljan, M.: Digital camera identification from sensor pattern noise. IEEE Transactions on Information Forensics and Security **1**(2), 205–214 (2006)
22. M'Raïhi, D., Machani, S., Pei, M., Rydell, J.: TOTP: Time-Based One-Time Password Algorithm. Internet Engineering Task Force pp. 1–16 (2011)
23. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
24. Quiring, E., Kirchner, M.: Fragile sensor fingerprint camera identification. In: Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–6 (2015)
25. Shen, C., Yu, T., Yuan, S., Li, Y., Guan, X.: Performance Analysis of Motion-Sensor Behavior for User Authentication on Smartphones. Sensors **16**(3), 345 (2016)
26. Sitová, Z., Šedenka, J., Yang, Q., Peng, G., Zhou, G., Gasti, P., Balagani, K.S.: HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users. IEEE Transactions on Information Forensics and Security **11**(5), 877–892 (2016)
27. Valsesia, D., Coluccia, G., Bianchi, T., Magli, E.: User Authentication via PRNU-Based Physical Unclonable Functions. IEEE Transactions on Information Forensics and Security **12**(8), 1941–1956 (2017)

28. Zhongjie, B., Sixu, P., Xinwen, F., Dimitrios, K., Aziz, M., Kui, R.: ABC: Enabling Smartphone Authentication with Built-in Camera. In: Proceedings of the Network and Distributed Systems Security Symposium (NDSS) (2018)