HARDWARE-ROOTED DEVICE AUTHENTICATION FOR SMART DEVICES

by

Zhongjie Ba

April 2019

A dissertation submitted to the
Faculty of the Graduate School of
the University at Buffalo, The State University of New York
in partial fulfilment of the requirements for the
degree of

Doctor of Philosophy

Department of Computer Science and Engineering

The thesis of Zhongjie Ba was reviewed by the following:

Dr. Kui Ren
Professor of Computer Science and Engineering
Thesis Advisor, Chair of Committee, IEEE Fellow

Dr. Chang Wen Chen
Professor of Computer Science and Engineering
Committee Member, IEEE Fellow

Dr. Dimitrios Koutsonikolas
Associate Professor of Computer Science and Engineering
Committee Member

Dr. Wenyao Xu
Associate Professor of Computer Science and Engineering
Committee Member

Acknowledgments

This dissertation would not have been possible without the generous support of an incredible number of individuals. First and foremost, I would like to express my deepest appreciation to my advisor and committee chair, Dr. Kui Ren, who guided me into the intriguing research area of IoT security and privacy. His great vision, unconditional support, inspiring encouragement and extensive knowledge in this field played a decisive role in completing the research of this dissertation. In the past four years, we explored promising research ideas, formulated challenging problems, and finding solutions together. I benefited tremendously from the scheduled individual meetings and group meetings and was greatly thankful for the time he allocated to me from his busy schedule. I appreciate all the insightful suggestions Dr. Ren has given to me and the profound belief he has put in my work. His great passion for research and his dedication to mentoring and teaching set a very good example for me to follow with my own academic career.

Second, I'm extremely grateful to my thesis committee, Dr. Chang Wen Chen, Dr. Dimitrios Koutsonikolas, and Dr. Wenyao Xu, for serving on my dissertation committee, and for giving valuable comments and criticisms that

have undoubtedly made the dissertation better. I would also like to extend my deepest gratitude to Dr. Chunming Qiao, Dr. Xinwen Fu, Dr. Aziz Mohaisen, Dr. Zhi Sun, and Dr. Lu Su for their kind support and invaluable suggestions throughout my PhD training.

My sincere gratitude further goes to current and former fellow graduate members at UbiSeC Lab, including Si Chen, Zhan Qin, Muyuan Li, Chaowen Guan, Sixu Piao, Zihao Shan, Tianhang Zheng, and Xiao Zhang. It is a great joy to work at UbiSeC Lab. I wish everyone the best in their lives.

I also would like to 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.

Finally, I would like to thank my family. I owe a great debt of gratitude to my parents, Haijin Ba and Shihong Wang. Thanks for giving me the courage and strength to pursue what I wanted. I am also grateful to my parents-in-law, Xiangjie Cao and Zhiqiang Yuan, for their love, encouragement and support. My deepest gratitude goes to my dearest wife, Yang Cao, who has made countless sacrifices to help me get to this point. Thanks for accompanying me throughout this entire process, for taking good care of our son Orick, and for always believing in me. This thesis is dedicated to all my dearest family members.

Table of Contents

List of Tables

List of Figures

Abstract

Smart devices play an increasingly important role in our daily life. Unintentional faults and malicious attacks could bring great danger to human lives and the environment, especially in safety-critical applications like medical devices, automobiles, building controls and the smart grid. However, due to the fact that industry is driven by functional requirements and fast-moving markets, a large number of sensors and devices are distributed in public areas unprotected and do not have the resources to support complex cryptographic mechanisms, which makes the authentication of smart devices a big challenge. In this dissertation, we seek to address this issue through exploiting the physical characteristics of their embedded sensors. In particular, we propose and investigate hardware-rooted device authentication systems which utilize the hardware fingerprint of various on-board sensors as the unique identity of smart Devices.

We first study the security issues underlying the hardware-rooted device authentication. In the literature, an enormous amount of research has been carried out in an attempt to identify devices through modeling the manufacturing imperfections of their built-in transducers. However, the vast majority of the work

in this area has focused on device tracking and identification. For adversarial settings like forensics and authentication, it remains unclear whether these methods will provide reliable identification results when the outputs of transducers are tampered by adversaries intentionally. In our work, we describe the architecture of hardware-rooted device authentication modalities and propose two kinds of challenge-response schemes for the authentication of different transducers. We outline two specific attacks that need to be taken into account while designing such system and describe several desirable properties that a fingerprinting method should have in order to be applicable for the authentication scenario.

We then carry out in-depth study on a specific hardware fingerprint named Photo Response Non-Uniformity (PRNU). PRNU is a reliable hardware fingerprint of digital cameras for image-to-camera matching in digital forensics. Unlike most hardware fingerprints that are composed of a few features drawn from the time domain and frequency domain of sensor outputs, this camera fingerprint is a large matrix consisting of millions of variables, which makes the fingerprint of each individual camera remarkably unique. This salient feature makes the PRNU a good candidate for the physical layer proof of a device. In this thesis, we conduct extensive experiments to understand the characteristics of a smartphone camera's PRNU and formulate the problem of the fingerprint forgery attack and the replay attack in camera-based authentication. We present new primitives for the PRNU forgery detection and propose two novel and practical camera-based smartphone authentication systems.

Finally, in order to further improve the security of camera-based authenti-

cation systems, we propose a privacy-preserving architecture for on-line image sharing. We first study the problem of camera fingerprint leakage in current image sharing practices. Our experimental results show that the PRNU fingerprint is robust enough to survive most image processing operations, including filtering, watermarking, beautifying, and compressing. Most images posted on the Internet expose the camera fingerprints of their photographing device directly to the public. These fingerprints enable the adversary not only to launch fingerprint forgery attacks against camera-based smartphone authentication systems, but also to launch identity linking attacks, which re-identify anonymous social network accounts through exploiting the digital cameras' fingerprints that are carried by the posted images. In order to counter the above attacks, we propose an intermediary between smartphone users and image sharing platforms that conceals the camera fingerprint of the photographing device. The proposed system is enabled to prevent malicious utilizations of camera fingerprint while preserving the beneficial applications.

Chapter 1

Introduction

1.1 Motivation

To ensure that smart devices can be trusted to be what they purport to be, it requres a unique and reliable identity that can be authenticated when a device attempts to connect to a central server. However, because smart devices usually have lower memory and processing capabilities, traditional authentication systems are complex and not ideally suited in securing smart devices. To address this issue, we propose and investigate hardware-rooted device authentication systems which utilize the hardware fingerprint of various on-board sensors as the unique identity of smart devices. In addition to the general study of hardware-rooted device authentication modalities, we also carry out in-depth study on a specific hardware fingerprint named Photo Response Non-Uniformity (PRNU) and propose two practical camera-based authentication systems for smartphones. Moreover, we also identify and address the security and privacy issues raised by the leakage of camera fingerprint in the current

image sharing practices.

Hardware-rooted device authentication. Reliably identifying and authenticating heterogeneous smart devices is a challenging topic as most smart devices do not have the resources to support complex cryptographic mechanisms. This project seeks to address this challenge through verifying the hardware fingerprint of the embedded sensors of smart devices. During the authentication stage, the verifier needs only to look at the sensor measurements uploaded by the target device and to verify its device-specific "noise component" (i.e., the hardware fingerprint), which is particularly light-weight and of low cost. Moreover, such authentication modality can easily be extended to support continuous authentication and multi-factor authentication (for devices with multiple sensors).

To enable hardware-rooted device authentication, we proposed two kinds of challenge-response schemes to authenticate sensors of different types (emitter and receiver). We then investigated the security issues underlying each scheme and identified two specific attacks: the replay attack and the fingerprint forgery attack. Based on the proposed system and threat model, we studied the feasibility of multiple existing fingerprinting approaches and outlined three desirable properties for a usable hardware fingerprinting method.

Camera-based smartphone authentication. The Photo-Response Non-Uniformity (PRNU) (Lukas, Fridrich and Goljan, 2006) 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. Unlike most hardware fingerprints that are composed of a few features drawn from the time domain and

frequency domain of sensor outputs (Dey et al., 2014; Zhou et al., 2014; Brik et al., 2008), this camera fingerprint is a large matrix consisting of millions of variables, which makes the fingerprint of each individual camera remarkably unique. Results have shown that, the PRNU-based identification approach can accurately differentiate over one million images captured by thousands of devices (Goljan, Fridrich and Filler, 2009). These salient features make the PRNU a good candidate for the physical layer proof of a smartphone.

The PRNU however is vulnerable against fingerprint forgery attacks. With a handful of images (e.g., on social media) from a victim smartphone, an adversary can extract the fingerprint of the victim device and embed the obtained fingerprint into arbitrary images of the same resolution (Goljan, Fridrich and Chen, 2011; Steinebach et al., 2010). Despite decades of research on camera fingerprinting, only few detection mechanisms have been proposed to detect forged fingerprints, and these mechanisms are either impractical or have security flaws.

To enable camera-based smartphone authentication, I conducted extensive experiments to understand the characteristics of a smartphone camera's PRNU and designed two authentication systems with reliable forgery detection mechanisms: **ABC**: the idea is to detect forgery attacks through tracking the fingerprint of the adversary's smartphone. This fingerprint in question is introduced during the challenge response stage where the adversary is required to capture a freshly generated QR code. The fingerprint of the adversarial device will be preserved in forged images, which renders the similarity value between forged images significantly higher than a normal value. Based on this observation, we designed ABC, the first forgery-resilient camera-based smartphone authentication system. **CIM**: this system innovatively introduced burst mode photograph-

ing and camera movement into the challenge-response process. It is built on top of two fundamental observations: (i) Because burst images are captured in rapid succession, the random noise components of a captured image can be partially preserved across multiple images captured in a row. The preserved noise forms a forgery-sensitive noisechain embedded in burst images. (ii) There exist various correlations between the movement of the camera and the noise components of the captured images. Both the noisechain and these correlations enable new detectors for fingerprint forgery attacks.

Obfuscation-based camera fingerprint concealment. To further improve the security of camera-based authentication systems, we investigate the problem of fingerprint leakage in current image sharing practices. We evaluated images from four representative social networks: Facebook, Wechat, Weibo and Flickr and studied the impact of a comprehensive set of image post-processing techniques. Our experimental results show that the PRNU fingerprint is robust enough to survive most image processing operations, including filtering, watermarking, beautifying, and compressing. As a result, most images posted on Internet carries the unique hardware fingerprint of the photographing device. Camera fingerprints are directly exposed to the public, which gives rise to security and privacy issues. For instance, an adversary can use these leaked fingerprints to launch fingerprint forgery attacks against camera-based smartphone authentication systems. The adversary can also use them to launch identity linking attacks, which re-identify anonymous social network accounts through exploiting the camera fingerprints that are carried by the posted images.

In order to protect users' camera fingerprints from being leaked, we propose to introduce an intermediary between smartphone cameras and social medias. It conceals the camera fingerprint of the image of interest before uploading it to

the Internet. The objective of the fingerprint concealment mechanism is to prevent malicious uses of PRNU while preserving the beneficial ones. This requires our system to conceal the PRNU fingerprint of an image without removing it, which is a grand challenge. The idea is to obfuscate the PRNU fingerprint of an image through embedding an irremovable noise component. With a specially designed noise component, the similarity between two sanitized images is always high, whether or not they are captured by the same device. An adversary can no longer use thresholding to determine if two images are captured by the same device. Moreover, using the embedded noise component as a probe, a fake fingerprint extracted from a user's online images can be easily detected. Meaning while, beneficial applications of PRNU are preserved because sanitized images will still carry the fingerprint of the photographing device.

1.2 Contribution

In this dissertation, we seek to strengthen security and reliance for smart devices through exploiting the physical characteristics of their embedded sensors. The research issues we addressed are summarized below:

From Hardware Fingerprinting to Multi-factor Authentication: in chapter 2, we investigate the feasibility of using a hardware fingerprint as the unique identity of a smartphone. We discuss various security issues underlying the hardware-rooted smartphone authentication and outline two specific attacks that need to be taken into account while designing such system. We also describe several desirable properties that a fingerprinting method should have in order to be applicable for the authentication scenario. Several classical fingerprinting methods are studied, and their security and usability are analyzed.

PRNU-based Smartphone Camera Identification and Authentication: in chapter 3, 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%.

Towards Practical Camera-based Smartphone Authentication via Camera Movement and Continuous Photographing: in chapter 4, we propose CIM, a practical and reliable camera-based smartphone authentication system. In CIM, a user is asked to move his/her smartphone along a specific route, take pictures of QR codes displayed on the verifier's interface in burst mode, and submit particular burst pictures to the verifier for authentication. We find that, because burst images are captured in rapid succession, the random noise components of a captured image can be partially preserved across multiple images captured in a row. The preserved noise forms a forgery-sensitive noisechain embedded in burst images. We also find that there exist various correlations between the movement of the camera and the noise components of the captured images. The

noisechain and these correlations are then explored for forgery detection. We performed extensive experiments with 22 smartphones of 5 different models. Our experiment results show that *CIM* can achieve 100% true acceptance rate at 0% false acceptance rate in both fingerprint matching and forgery detection.

Preventing Camera Fingerprint Leakage via Obfuscation-based Fingerprint Concealment: in chapter 5, we first evaluate the effectiveness of several critical attacks caused by camera fingerprint leakage. We then propose CFP, an intermediary between smartphone users and image sharing platforms that conceals the camera fingerprint of the photographing device. Instead of removing the camera fingerprint from the image of interest, our system protects user privacy through obfuscating the camera fingerprint with a specially designed random perturbation component. With such design, the proposed system is enabled to prevent malicious utilizations of camera fingerprint while preserving the beneficial applications. Extensive experiments are conducted to demonstrate the effectiveness and efficiency of the CFP system on various social platforms. Using the CFP obfuscated images, the True Positive Rate of identity linking attacks is reduced by around 85%. For identity forgery attacks, our system enables an effective detection mechanism that could achieve 100% detection rate.

1.3 Roadmap

The rest of this dissertation is organized as follows.:

Chapter 2 explore authentication modalities that verify a smartphone's identity through tracking the hardware fingerprints of its built-in transducers. We begin with an overview of hardware fingerprinting approaches in section 2.2.

Section 2.3 describes the architecture of hardware-rooted smartphone authentication systems, focusing on the players involved and the communication channels. Section 2.4 studies several existing fingerprinting methods and discuss their performance under replay attacks and fingerprint forgery attacks. Section 2.5 concludes this chapter.

Chapter 3 demonstrates the feasibility of using PRNU as a smartphone's unique identity and presents our first camera-based smartphone authentication system. We begin with reviewing the current PRNU-based digital camera fingerprinting method in Section 3.2. Section 3.3 formulates the problem to be addressed in this chapter. Section 3.4 presents our smartphone authentication protocol. Section 3.5 analyzes the security feature of the proposed protocol. Section 3.6 conducts the performance evaluation. Section 3.7 reviews the related existing work on hardware fingerprinting. Section 3.8 concludes this chapter.

Chapter 4 presents new primitives for the PRNU forgery detection and introduce a new camera-based smartphone authentication protocol which is more practical and more secure. Section 4.2 reviews related work. Section 4.3 introduces background knowledge for camera-based smartphone authentication. We present and validate the two primitives for forgery detection in Section 4.4. Section 4.5 gives the design of *CIM*. Section 4.6 and Section 4.7 evaluates *CIM* with extensive experiments. Section 4.9 concludes this chapter.

Chapter 5 highlights and evaluates the critical attacks caused by camera fingerprint leakage and proposes a real-time fingerprint concealment system to counter the attacks. We first introduce the background knowledge in Section 5.2. Section 5.3 highlights two specific attacks and evaluates their effectiveness on current image sharing practices. Section 5.4 discusses the failure of fingerprint removal and gives the design of CFP in detail. Section 5.5 evaluates the

proposed system with extensive experiments. Section 5.6 concludes this chapter.

Chapter 6 concludes this dissertation and discuss several directions for my future work.

Chapter 2

From Hardware Fingerprinting to Multi-factor Authentication

2.1 Introduction

Driven by the concern of password and fingerprint leakage, many organizations today tend to use the smartphone that a user already carry as an extra authentication factor in order to provide enhanced security as well as convenience. The verification of a smartphone is typically carried out through checking a software-level identity or a hardware-level component. A software-level identity can be a cryptographic private key or a public identification number. While the key is normally private to a specific device, it is vulnerable to relay attacks and device clones (Danev, Zanetti and Capkun, 2012). A hardware-level component is a hardware platform (e.g., the Secure Element) designed to securely host applications and store cryptographic data. Although this approach can provide enhanced security, its scalability is limited by the requirement of additional hardware.

One promising direction to authenticate a smartphone is to verify the hardware fingerprints of its built-in transducers. A transducer is a device that converts a signal from one form to another. Its hardware fingerprint refers to a pattern noise on its output signal incurred by manufacturing imperfection. For most types of transducers, the hardware fingerprint remains constant over time and is difficult to replicate physically, making it a good candidate for the physical layer proof of a smartphone.

In this article, we first describe the architecture of hardware-rooted smartphone authentication modalities, focusing on the players involved and their communication channels. Two kinds of challenge-response schemes are proposed for the authentication of different transducers: *Emitter-based scheme* and *Receiver-based scheme*. We then discuss the security issues underlying each scheme and identify specific attacks. The desirable properties of a hardware fingerprinting method is also discussed in the context of smartphone authentication. Finally, to explore the solution space, we study several existing fingerprinting methods and analyze their suitability for hardware-rooted smartphone authentication systems.

2.2 Hardware Fingerprinting Overview

In this paper, a hardware fingerprint refers to the physical feature of a transducer exploited for the identification of a smartphone. During the transformation of an input signal, a transducer usually introduce a systematic distortion into its output signal due to manufacturing imperfection. As this distortion remains constant overtime, it has widely been explored to track smartphones.

There is a rich set of transducers that have been demonstrated to have unique

fingerprints (Fig. 2.1). An acoustic sensor's fingerprint is its frequency response curve in a given frequency range. The fingerprint of an image sensor is caused by the non-uniform dimension of its pixels. It introduces a constant white Gaussian noise into every image captured by the smartphone. For wireless transmitters, their fingerprints mostly come from the front-end components like amplifiers and filters. For motion sensors, their fingerprints come from the imperfection of their electro-mechanical structure which results in calibration errors and some statistical features. The magnetometers are also known to have calibration errors (linear bias). This rich set of hardware fingerprints provide a variety of modalities for smartphone authentication.

The verification of a smartphone's hardware fingerprint involves two stages: a *training stage* and an *identification stage*. During the *training stage*, the verifier collects a number of signals generated by a transducer on the target smartphone and extracts the fingerprint shared by those signals. The form of the fingerprint and the extraction process may vary depending on the type of the transducer. During the *identification stage*, the verifier checks if a query device is the target smartphone it claimed to be. She first catches a signal known to be generated by the query device. Then, she extracts the fingerprint on that signal and matches it to the fingerprint record of the target smartphone. The matching method also varies depending on the type of the transducer.

Using above procedures, most fingerprinting methods can achieve high accuracy under lab settings. However, in real-world deployment, there are many environmental factors that could affect their identification results. In section 2.4, we will use examples to discuss the impact of different factors and how they can help/hinder the design of authentication systems.

Figure 2.1. Sensors in a smartphone

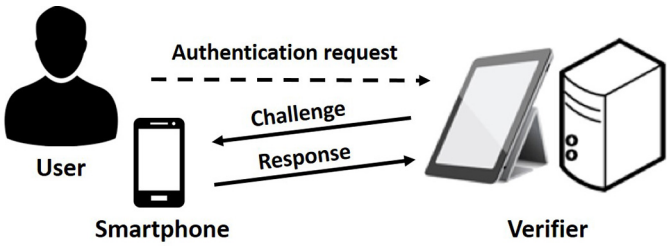


Figure 2.2. System model

2.3 Hardware-rooted Smartphone Authentication

2.3.1 Architecture

A hardware-rooted smartphone authentication system involves three entities: a user, her smartphone, and a verifier. As illustrated in Fig. 2.2, the user is an individual that needs to be authenticated into the system. She tries to prove her identity to the verifier using her smartphone. The smartphone serves as a possession factor and is identified by the hardware fingerprint of its built-in transducer. The verifier consists of a terminal and a server. She uses the terminal to interact with the user and to collect the signal generated by the user's smartphone. Her server maintains the fingerprint database and determines the identity of smartphones.

Like all other authentication systems, the essence of a hardware-rooted

smartphone authentication system is the challenge-response scheme. Depending on the form of the applied transducer, challenge-response schemes can be classified into two categories: *Emitter-based scheme*: the transducer applied in this scheme is an emitter that converts electrical signals to other signals. During the challenge-response stage, the verifier sends a stimulation signal to the user's smartphone through a digital channel (e.g., WiFi). The user then feeds the received signal into the transducer on her smartphone. Finally, the verifier uses a receiver on her terminal to collect the signal emitted by the transducer. *Receiver-based scheme*: the applied transducer is a receiver that converts other signals to electrical signals. In this case, the verifier directly emits a stimulation signal (e.g., an audio) via an emitter on her terminal. The user uses her smartphone to convert the stimulation signal to an electrical signal and transmits the obtained signal to the verifier through a digital channel. For both scheme, the verifier determines the identity of the smartphone through checking the fingerprint on the collected signal.

It is important to note that, in most cases, the signal collected by the verifier will also contain the fingerprint of the verifier's emitter/receiver. For instance, in the speaker-based scheme, the frequency response curve extracted from the received audio will be a product of the speaker's frequency response and the microphone's. Therefore, the verifier should always eliminate the fingerprint of her device before conducting fingerprint matching.

2.3.2 Security Threats and Attacks

In this article, we consider the existence of malicious users impersonating legitimate ones. The objective of a malicious user is to convince the verifier that

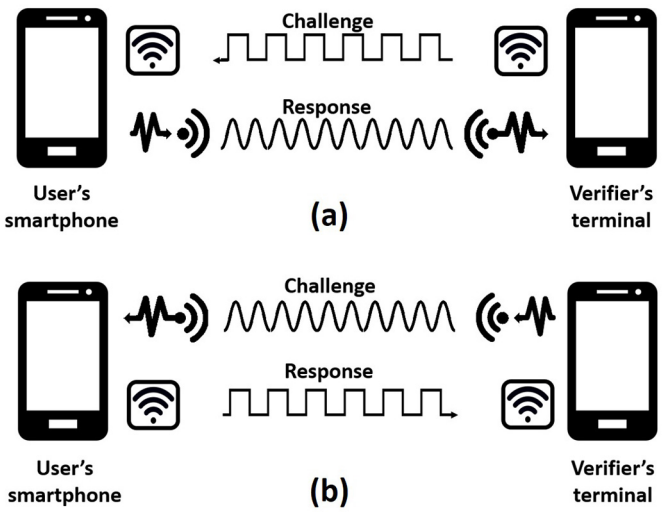


Figure 2.3. Challenge-response schemes: a) emitter-based scheme; b) receiver-based scheme.

a transducer on her smartphone is the one associated with a legitimate user. Although it is difficult for an adversary to steal or to physically replicate the transducer inside a user's smartphone, she might be able to collect a number of signals generated by that transducer (e.g., from the Internet). After collecting a certain number of signals, the adversary may attack the system via *replay attacks* or *fingerprint forgery attacks*.

In *replay attacks*, the adversary fraudulently repeats a victim smartphone's previous output signal to the verifier. For *receiver-based schemes*, because the outputs of a receiver will be transmitted through a digital channel, the adversary can easily repeat the victim's signal without introducing any additional distortion. For *emitter-based schemes*, the signal repeated by the adversary could be badly distorted during their transmission since the emitter produces analog outputs. It is much more difficult for the adversary to conduct successful replay attacks.

In *fingerprint forgery attacks*, the adversary feeds the stimulation signal into a

foreign device and synthesize a forged signal that contains the fingerprint of a legitimate smartphone. For *receiver-based schemes*, the stimulation signal is emitted on the server side and is captured on the user side. The adversary can embed the obtained signal with the victim's fingerprint before uploading it to the verifier. The success rate of this attack mainly depends on the effectiveness of the embedding method. For *emitter-based schemes*, the stimulation signal is emitted on the user side and is captured on the server side. Since the adversary can no longer access to the captured signal, the forgery strategy here is modifying the stimulation signal before feeding it into the transducer. An adversarial device playing a modified stimulation signal should generate a similar output as the victim device playing the original stimulation signal. The success rate of this attack is mainly determined by the quality of the modified stimulation signal.

2.3.3 Desirable Properties of Hardware Fingerprint

In the context of smartphone identification and tracking, a hardware fingerprinting method can be considered usable as long as it allows a certain degree of universality, uniqueness, permanence and collectability (Baldini and Steri, 2017). However, in the authentication scenario, the fingerprinting method should not only be able to identify smartphones, but also be able to defeat various impersonation attacks. Here we list several desirable properties that a fingerprinting method should have in order to be applicable for authentication systems.

Data independence: the applied fingerprint can be extracted from various signal patterns. This enables the verifier to defeat replay attacks through generating fresh stimulation signals for each authentication attempt. The dimension of

possible challenges/responses is determined by the amount of signal patterns supported by the fingerprinting method.

Repeating sensitivity: the applied fingerprint is sensitive to the extra distortions introduced during replay attacks, such as the fingerprint of adversarial devices and environmental noise.

Forgery resilience: the fingerprinting approach should be able to differentiate synthesized signals from genuine ones. Both the forgery technique and the detection mechanism may vary depending on the form of the fingerprint.

For receiver-based schemes, the applied fingerprinting approach must be *Data independent* and *Forgery resilient* in order to defeat replay attacks as well as forgery attacks. For emitter-based schemes, *Data independent* could be replaced or combined with *Repeating sensitive*. Considering real-world deployment and usability, the fingerprinting approach applied in both scheme should also be *Time efficient* and *User friendly*.

2.4 Case Studies

In this section, we study several existing hardware fingerprinting methods from the perspective of smartphone authentication. We start from their support of the above mentioned properties and discuss their performance under replay attacks and fingerprint forgery attacks. We also summarize various practical challenges that need to be resolved while fulfilling the security and usability requirements of a hardware-rooted smartphone authentication system.

2.4.1 Accelerometer Fingerprinting

An accelerometer is a device that measures the current acceleration of its body. When the sensor experiences an acceleration in one direction, a movable seismic mass inside the accelerometer will shift to the opposite direction. The sensor then converts the shift of the mass into an electrical signal proportional to the acceleration. Its fingerprint comes from the imperfection of its electro-mechanical structure (e.g., the flexibility of the seismic mass).

In order to measure the difference between accelerometers, Dey *et al.* (Dey et al., 2014) propose to stimulate each accelerometer with a vibration motor and use bagged decision tree to differentiate their outputs. Sixteen time-domain features and twenty frequency-domain features are extracted to identify each individual accelerometer. Under lab setting, their approach achieved over 96% precision and recall in differentiating 107 individual devices. However, according to their experimental results, there are many factors that can affect the fingerprint, such as sampling rate, the smartphone case, and the surface on which the device is placed. The precision and recall could drop below 70% when the test environment is different from the training environment. For authentication systems, the accelerometer can be used in conjunction with the gyroscope to achieve higher identification accuracy. This is because that the gyroscope can also respond to the stimulus generated by the vibration motor and is known to have calibration errors.

Accelerometer-based authentication systems should follow the receiver-based challenge-response scheme. Using above identification method, they will suffer from both replay attacks and forgery attacks. 1) *Replay attacks*: Due to the openness of accelerometer readings (Das, Borisov and Caesar, 2016), an ad-

versary can easily collect a victim smartphone's output signal and retransmit it to the verifier to impersonate the user. Worse yet, the dimension of the challenges/responses is extremely limited since the applied fingerprinting approach only accepts vibration and gravity as stimulation signals. 2) *Forgery attacks:* denoting the real acceleration as a^o, the measurement of an accelerometer is normally modeled as $a^M = Sa^o + O$, where S and O respectively represents the gain and offset errors in the output signal (Das, Borisov and Caesar, 2016). By manipulating these two errors, an adversary can control the mean and the deviation of an accelerometer's measurement and generate the fingerprint of another device. Although the fingerprinting approach uses many different features to identify each device, the mean and the deviation are the most discriminating features. Manipulating these two features can significantly degrades the classification accuracy of the fingerprinting approach (Das, Borisov and Caesar, 2016).

2.4.2 Digital Camera Fingerprinting

A digital camera is a device that produces digital images. It uses an image sensor to convert light signals into electrical signals. The fingerprint of a digital camera comes from its image sensor's non-uniform sensitivity to light (Goljan, Fridrich and Filler, 2009). This imperfection introduces an constant white Gaussian noise onto every image captured by the same camera.

To track this unique noise, Goljan *et al.* (Goljan, Fridrich and Filler, 2009) propose to use denoising filter to extract the fingerprint carried on an image and to use Peak to Correlation value to match fingerprints. Under public setting, their approach achieved a false rejection rate less than 2.38% at a false acceptance rate below 0.0024% in differentiating 6896 individual cameras. The only factor that

affects the PRNU fingerprint is the intensity of ambient light (Ba et al., 2018). By rising the light intensity of the photographing environment, the identification accuracy can be further improved.

Camera-based authentication systems also follow the receiver-based challenge-response scheme. They can be resilient to both replay attacks and forgery attacks. 1) *Replay attacks:* although it is very easy for an adversary to collect a victim's images (e.g., from her Facebook) and conduct replay attacks, the verifier can defeat this attack through challenging the user to provide images with specific features. The dimension of this challenges/responses is very large due to the data independence of the camera fingerprint. For instance, Valsesia et al. (Valsesia et al., 2017) propose to challenging the user to provide uncompressed images to be authenticated into the system. Since most users have shared only JPEG images with the public, it is difficult for an adversary to collect raw images and conduct successful replay attacks. In another system, Ba et al. (Ba et al., 2018) propose to challenging the user to capture and upload freshly generated QR codes. The verifier then conducts image content matching to detect the liveness of the query image. 2) *Forgery attacks:* as the fingerprint of a camera is an additive noise component on the image content, an adversary can easily plant a victim device's fingerprint ($\hat{\mathbf{K}}$) into a foreign image ($\mathbf{J}$) using equation 2.1 (Goljan, Fridrich and Chen, 2011):

$$\mathbf{J}' = \mathbf{J}(1 + \alpha\hat{\mathbf{K}}) \tag{2.1}$$

where $\mathbf{J}'$ and α respectively represents the forged image and the strength of the fingerprint. This approach enables the adversary to fabricate arbitrary images carrying the victim's camera fingerprint and to bypass most liveness detection

mechanisms. In the literature, three mechanisms (Quiring and Kirchner, 2015; Goljan, Fridrich and Chen, 2011; Ba et al., 2018) have been proposed for the detection of fingerprint forgery attacks. Under certain assumptions, each of them can achieve a high success rate.

2.4.3 Loudspeaker Fingerprinting

A loudspeaker is a transducer that converts electrical signals into sound signals. When an electrical current arrives at the loudspeaker, it flows through the loudspeaker's electromagnet and generates a varying magnetic field. The magnetic field then drives the loudspeaker's diaphragm to vibrate and to generate sound waves. Due to manufacturing imperfection, loudspeakers have non-uniform responses at difference frequency bands, which can be used to identify individual devices (Zhou et al., 2014).

To calculate a loudspeaker's frequency response Ξ at a certain frequency f, the verifier needs to feed the loudspeaker with a stimulation signal covering frequency f and record the played sound. Then, the target frequency response Ξ_f can be calculated using equation 2.2:

$$\Xi_f = \frac{S(f)}{R(f)} \tag{2.2}$$

where $S(f)$ and $R(f)$ respectively represents the stimulation signal's and the recorded signal's Fourier coefficients at the frequency f. The stimulation signal is normally designed as a fixed audio pattern in order to generate robust frequency response curves. Chen et al. (Chen, Zhang, Qin, Mao, Qin, Shen and Li, 2017) propose to stimulate the speaker with a combination of 41 single tone signals from 4KHZ to 20KHZ. In most scenarios (e.g., office and roadside),

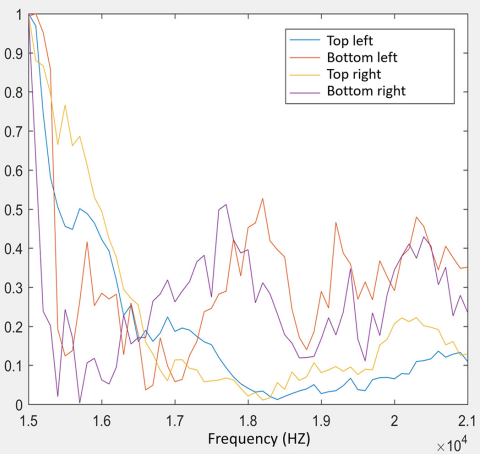


Figure 2.4. The frequency response of a speaker measured at the top left, top right, bottom left and bottom right of a microphone.

their method can achieve an equal error rate less than 2%. For acoustic signal, the measured frequency response is also affected by the recording device, the relative position between the speaker and the microphone (Fig. 2.4), and multi-paths. Fortunately, Chen et al. have shown that it is possible to eliminate those unwanted distortions and achieve high accuracy under public settings (Chen, Zhang, Qin, Mao, Qin, Shen and Li, 2017).

Speaker-based authentication schemes use the emitter-based challenge-response scheme to authenticate users. They are robust against replay attacks but are vulnerable to forgery attacks. 1) Replay attacks: the adversary collects a victim smartphone's sound signal and replays it to the verifier's terminal using a foreign loudspeaker. In this attack, the sound recorded by the verifier's terminal has been distorted not only by the fingerprint of the victim device but also the frequency response of the adversarial devices. Due to the existence of extra distortions, existing fingerprint matching approaches (Chen, Zhang, Qin, Mao, Qin, Shen and Li, 2017) can effectively defeat audio replay attacks. 2) Forgery attacks: following the forgery strategy described earlier, the adversary can at-

tack the system through playing a modified stimulation signal. Upon receiving the stimulation signal generated by the verifier, the adversary first divides the spectrum of that signal by the fingerprint of the adversarial loudspeaker, thereby eliminating the distortions caused by that foreign device. Then, in order to embed the victim device's fingerprint, the adversary further multiplies the spectrum of the signal obtained in the above step by the frequency response curve of the victim loudspeaker. It has been shown that, by playing this modified stimulation signal on the adversary's device, the fingerprint extracted by the verifier can be very similar to the victim's fingerprint (Ba, Piao and Ren, 2017).

2.4.4 Wireless Transmitter Fingerprinting

The wireless transmitter converts data streams into Radio Frequency (RF) signals. When a wireless device attempts to send out a data stream, the data stream will be processed by the modulator, the digital to analog converter, the local oscillator, and the power amplifier, transformed into RF signals that can propagate in various mediums. The fingerprints of wireless transmitters basically come from the imperfections of these hardware components.

There are mainly two kinds of schemes for modeling the hardware imperfections of a wireless transmitter: the transient-based scheme and the modulation error-based scheme. Transient-based systems identify devices via the time-domain and/or frequency-domain features of the transition signal. The highest accuracy that those systems can reach exceeds 95%. However, the variations of propagation channels might significantly degrade the performance of these systems. PARADIS proposed by Brik et al. (Brik et al., 2008) is a modulation error-

based scheme, which assigns 5 modulation errors as the device fingerprint. It is experimentally demonstrated that PARADIS can differentiate more than 130 NICs with over 99% accuracy and is resilient against ambient noise and channel variations.

Wireless transmitter-based authentication systems should also resort to the emitter-based scheme. The transient-based systems are vulnerable to both replay attacks and forgery attacks (Danev et al., 2010). Systems that use modulation errors will suffer from forgery attacks. 1) *Replay attacks*: due to the open wireless medium, an adversary can easily record and repeat a vicitm device's RF signals through setting up a transceiver. If the applied transceiver is a high-end device (e.g., Ettus USRP), the replayed signal will carry almost the same features as the original signal. This makes transient-based systems vulnerable to replay attacks. For modulation error-based systems, due to their data independence, the verifier is able to randomize the challenged signal packet and conduct liveness detection. 2) *Forgery attacks*: most existing RF fingerprinting systems are vulnerable to forgery attacks. With a high-end Software-Defined Radio (SDR), an adversary is able to modify the characteristics of RF signals, such as the modulation errors, to generate the authenticated RF fingerprints. In this case, the signal transmitted by the SDR will carry a similar fingerprint as the signal generated by a legitimate device. Attacks of this nature have not been addressed so far.

2.4.5 Practical Challenges

In summary, hardware fingerprinting is a promising direction for the future smartphone authentication, but the state-of-the-arts cannot satisfy all the desir-

able properties listed in section 2.3.3. There are still several practical challenges that need to be resolved.

Detecting fingerprint forgery attacks: forgery detection is the main issue that constrains the solution space of the hardware-rooted smartphone authentication. While most fingerprinting approaches are subject to fingerprint forgery attacks, few of them have reliable detection mechanisms.

User capacity analysis: in the literature, the user capacity of a hardware fingerprinting approach is usually determined through a series of identification experiments. While these experiments establish a preliminary understanding of the user capacity of an approach, the results are just limited to their individual experiment setups and cannot be universally applied to various authentication scenarios (Wang et al., 2016).

Conflict between security and efficiency: although most fingerprinting approaches have proven to be efficient in fingerprint extraction and matching, there is no guarantee that their forgery detection mechanisms will also be efficient. In order to detect forged signals, existing mechanisms normally need to compare the query signal with a significant amount of signals generated by the target device (Goljan, Fridrich and Chen, 2011), which lead to a considerable time and storage overhead.

Conflict between security and usability: a practical authentication system should only involve operations that are familiar and convenient for most users. However, detecting forged signals normally requires the user to perform complex operations, which may degrade the user experience of the authentication system.

Identifying and addressing influential factors: the fingerprinting approach should be applicable to different deployment environments. Extensive exper-

iments should be conducted to identify and address factors that affect the identification accuracy, such as the aging of the query device, the hardware fingerprint of the verifier's terminal and other environmental factors.

2.5 Concluding Remarks

Due to the explosive growth of data leakage, passwords and fingerprints can no longer provide adequate protection for users' on-line accounts. Many organizations began to seek alternative authentication modalities to provide enhanced security. In this article, we explore authentication modalities that verify a smartphone's identity through tracking the hardware fingerprints of its built-in transducers. We first describe the architecture of hardware-rooted smartphone authentication systems, focusing on the players involved and the communication channels. Two kinds of challenge-response schemes are presented to collect the output signals of different transducers. We then analyze the security threats underlying these schemes and list several desirable properties for a usable hardware fingerprinting method. After that, we study several existing fingerprinting methods and discuss their performance under replay attacks and fingerprint forgery attacks. Hardware-rooted smartphone authentication is an important research area with great challenges. We believe it will attract more and more research efforts in the next few years.

Chapter 3

PRNU-based Smartphone Camera Identification and Authentication

3.1 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 (Index, N.d.), 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 (Chen, Ren, Piao, Wang, Wang, Weng, Su and Mohaisen, 2017), 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 (Dey et al., 2014; Remley et al., 2005; Brik et al., 2008; Chen et al., 2015) 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* (Ba and Ren, 2017). 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 (Chen et al., 2015; Danev et al., 2010). 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) (Lukas, Fridrich and Goljan, 2006) 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 us-

able 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 (Goljan, Fridrich and Chen, 2011; Quiring and Kirchner, 2015; Gloe et al., 2007; Steinebach et al., 2010). 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 detection mechanisms suffer from either poor reliability (Goljan, Fridrich and Chen, 2011) or huge transmission and storage overhead (Quiring and Kirchner, 2015).

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 to-

tal 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.

3.2 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.

3.2.1 PRNU-based Camera Fingerprinting

PRNU (Lukas, Fridrich and Goljan, 2006; Böhme and Kirchner, 2013) 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 (3.1) (Brik et al., 2008),

$$\mathbf{I} = \mathbf{I}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \Theta, \tag{3.1}$$

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 (Lukas, Fridrich and Goljan, 2006; Chen, Fridrich and Goljan, 2007), it can be extracted using a denoising filter. The extracted noise residue $\mathbf{W}_{(i)}$ can be represented as Equation (3.2) (Chen et al., 2008),

$$\mathbf{W}_{(i)} = \mathbf{I}_{(i)}\mathbf{K} + \Xi_{(i)}, \tag{3.2}$$

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 ($\mathbf{K}$) (Cain, Hayat and Armstrong, 2001). 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 (3.3),

$$\hat{\mathbf{K}} = \frac{\sum_{i=1}^{N} \mathbf{W}_{(i)}\mathbf{I}_{(i)}}{\sum_{i=1}^{N} (\mathbf{I}_{(i)})^2} = \mathbf{K} + \Delta, \tag{3.3}$$

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 $q = corr(\mathbf{K}, \hat{\mathbf{K}})$ (Goljan, Fridrich and Chen, 2011), 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) (Goljan, 2008).

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. (Goljan, Fridrich and Filler, 2009) 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.

3.2.2 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 (3.4),

$$\mathbf{J}' = \mathbf{J}(1 + \alpha\hat{\mathbf{K}}), \tag{3.4}$$

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 (Quiring and Kirchner, 2015) and triangle test (Goljan, Fridrich and

Chen, 2011). *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.

3.3 Problem Statement

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

3.3.1 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.3.1 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

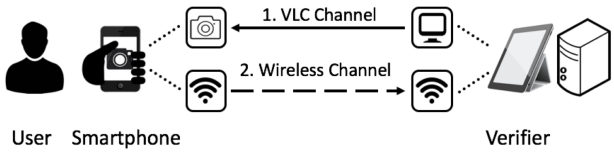


Figure 3.1. System model. The verifier authenticate a user's smartphone through tracking the fingerprint of its built-in camera. The verifier first challenges the smartphone to capture and upload the image shown on its interface. Then, the verifier extracts the fingerprint of the received image and correlates it to the reference fingerprint to authenticate the smartphone.

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.

3.3.2 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.

3.3.3 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.

Table 3.1. Examples of image sensors for digital cameras.

Digital camera	Sensor size (mm^2)	pixel amount (million)
Canon EOS 5D Mark II	36.00×24.00	21.1
Sony A850	35.90×24.00	24.6
Nikon D300s	23.60×15.80	12.3
Pentax Pentax K-30	23.70×15.70	16.3
Sigma SD1 Merrill	23.50×15.70	15.36

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.

3.4 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.

3.4.1 Smartphone Camera Fingerprinting

Table 3.1 and 3.2 (*Image Sensor Relative Size Comparison Tool*, N.d.) show 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

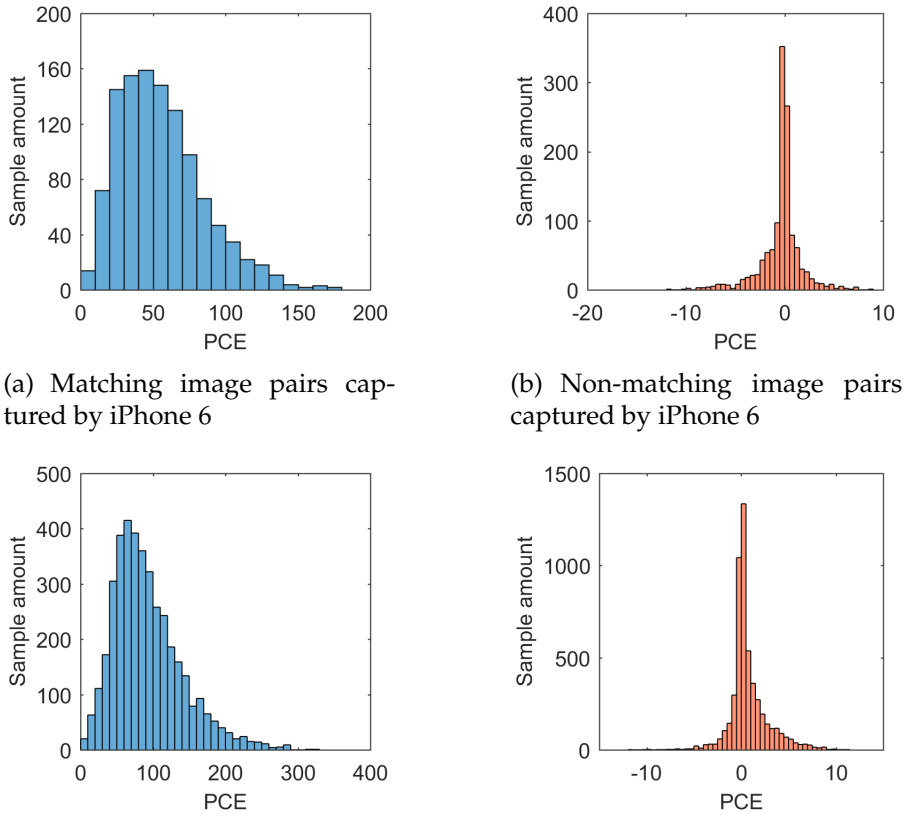
Table 3.2. Examples of image sensors for Smartphone cameras.

Smartphone camera	Sensor size (mm^2)	pixel amount (million)
Samsung Galaxy S4	4.69 ×3.53	13
Apple iPhone 6	4.89×3.67	8
HTC One X	4.54×3.42	8
LG G3	4.69×3.53	13.13
Nokia Lumia 920	4.80×3.60	8.7

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. 3.2) 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



(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

Figure 3.2. 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).

with the rise of the intensity of ambient light (will be shown in section 3.6).

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. 3.2 shows the distribution of the obtained PCE

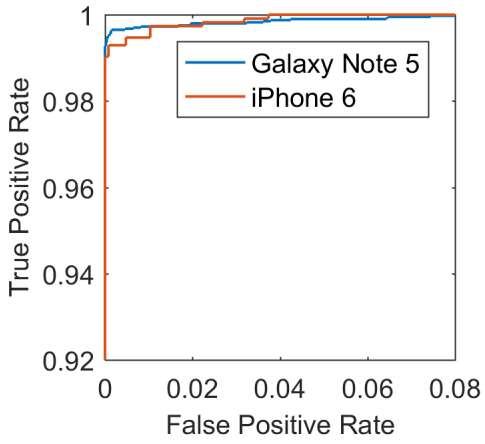


Figure 3.3. ROC curve for fingerprint matching. True positive rate measures the percentage of matching images that are correctly identified. False positive rate measures the percentage of non-matching images that are identified as matching ones.

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. 3.3. Minimizing the total error rate of fingerprint matching based on Fig. 3.3, 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 am-

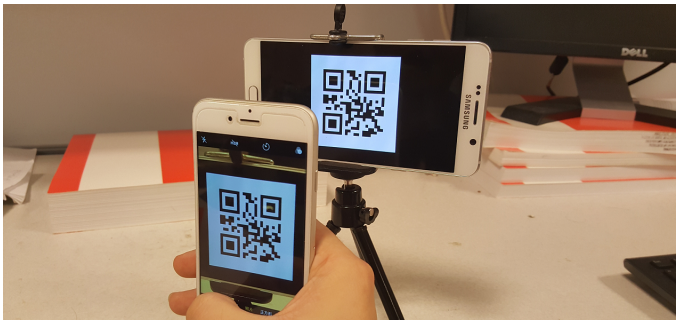


Figure 3.4. Use case: a user captures an image shown on the verifier's interface to be authenticated (or registered).

bient light or use a reference fingerprint extracted from a bright image. As will be shown in section 3.6, 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.

3.4.2 Basic Authentication Schemes

Before presenting the full-fledged ABC protocol that achieves all three design goals outlined in Section 3.3.3, 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.

3.4.2.1 System Framework

Fig. 3.4 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.

3.4.2.2 Basic Scheme I

This authentication scheme, shown in Fig. 3.5, 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 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 $\mathbf{I}_{(r)}$.

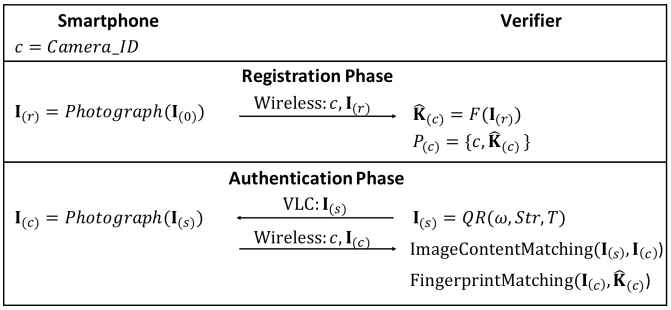


Figure 3.5. Basic Scheme I. *Registration*: the user uploads an arbitrary image captured by her smartphone. *Authentication*: the verifier challenges the user to capture a freshly constructed QR code shown on its interface. The QR code is encoded with an abstract of the ongoing transaction, which enables the user to verify the information before authorizing.

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

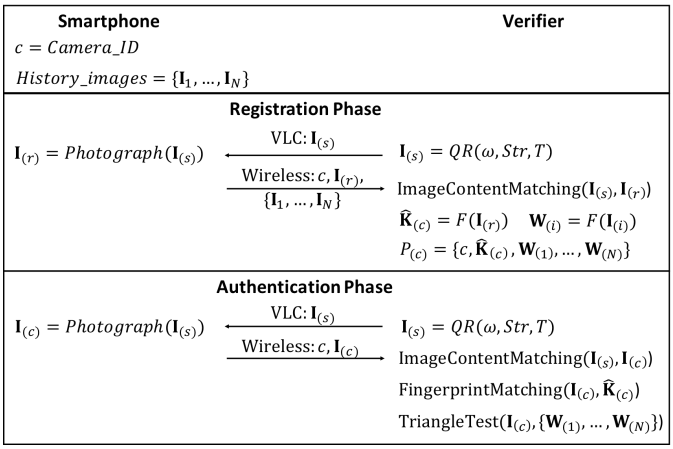


Figure 3.6. Basic Scheme II. *Registration*: the user uploads one image freshly captured by her smartphone and all other images the smartphone has ever captured. *Authentication*: this process is similar to the process in basic scheme I, except that triangle test is applied to detect forged images.

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.

3.4.2.3 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 3.2.2. 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.

Fig. 3.6 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 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

Algorithm 1 Triangle Test

F1 **function** TriangleTest($\mathbf{I}_{(q)}, \{\mathbf{W}_{(1)}, ..., \mathbf{W}_{(N)}\}$)
1. $\mathbf{W}_{(q)} \leftarrow F(\mathbf{I}_{(q)})$
2. **for** i:= 1 **to** N **do**
3. $\eta \leftarrow PCE(\mathbf{W}_{(i)}, \mathbf{W}_{(q)})$
4. **If** ($\eta > threshold$) **then**
5. Reject
6. **end if**
7. **end for**
8. Accept.

end function

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.

3.4.3 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.

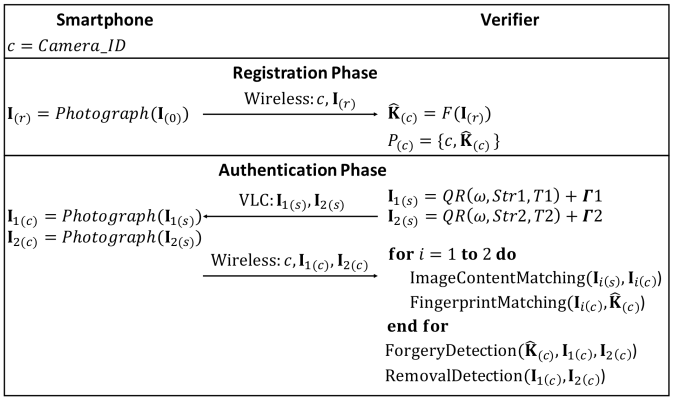


Figure 3.7. Full-fledged authentication protocol. *Registration*: the user uploads an arbitrary image captured by her smartphone. *Authentication*: the verifier enforces the user to capture two consecutive images shown on its interface.

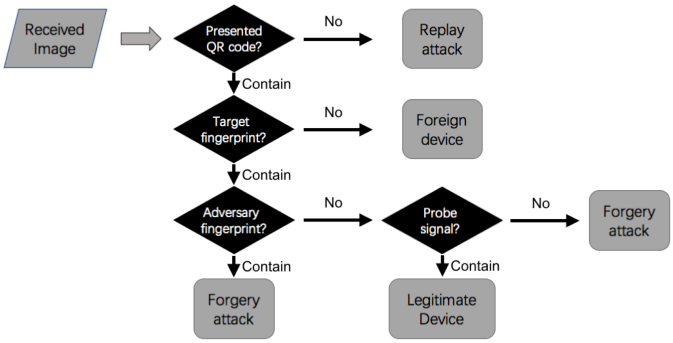


Figure 3.8. Attack detection flow: since the user has confirmed the information of the ongoing transaction, the verifier needs only to detect replay attack and fingerprint forgery attack.

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. 3.7). 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.

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. 3.8 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

Algorithm 2 Forgery Detection

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

end function

Algorithm 3 Removal Detection

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

end function

through a fingerprint removal process, due to Γ_i's sensitivity to fingerprint removal, the similarity value will be lower than a threshold.

3.5 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*.

3.5.1 Replay Attack

An adversary may attempt to impersonate a legitimate smartphone by fraudulently replaying a captured image token 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 (*Information capacity and versions of the QR Code*, N.d.). 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.

3.5.2 Man in the Middle Attack

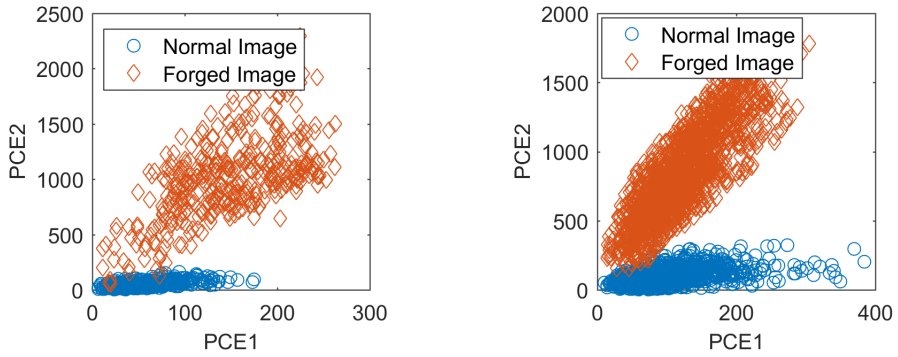
An adversary may attempt to lure a legitimate user to authorize a malicious request through modifying the communication between the user and the verifier. The attacking process is as follows: 1) The legitimate user initiates her request on the verifier's interface. 2) The adversary (e.g., a malicious 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.

3.5.3 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



(a) iPhone 6 (b) Galaxy Note 5

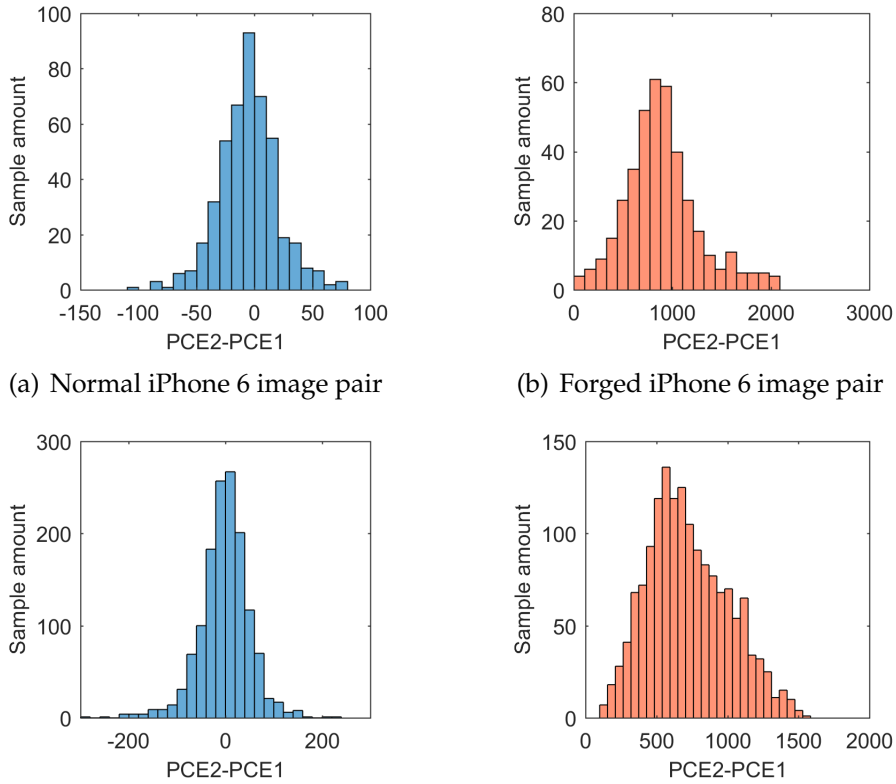
Figure 3.9. PCE for forgery detection. PCE1 measures the correlation between one tested image and the reference fingerprint. PCE2 measures the correlation between two tested images.

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.

3.5.3.1 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



(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

Figure 3.10. 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.

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 environment. As will be shown in section 3.6, images captured in this environment has the weakest fingerprint. We tested two image sets collected

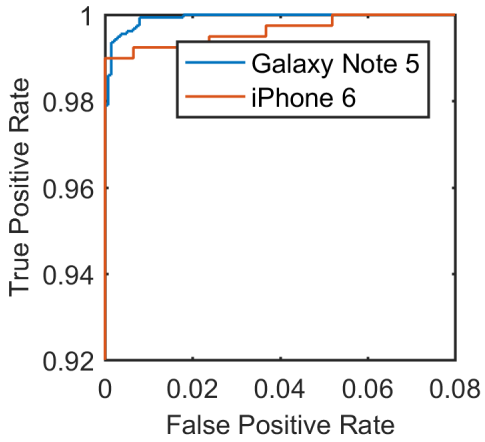


Figure 3.11. Forgery detection. True positive rate measures the percentage of forged images which are correctly identified. False positive rate measures the percentage of legitimate images that are identified as forged ones.

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. 3.9, 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. 3.10.

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. 3.11 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.

3.5.3.2 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 (Karaküçük and Dirik, 2015; Dirik and Karaküçük, 2014); 2) reconstruct an image containing the presented QR code. Since both removal strategies remove all noise components, 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 finger-

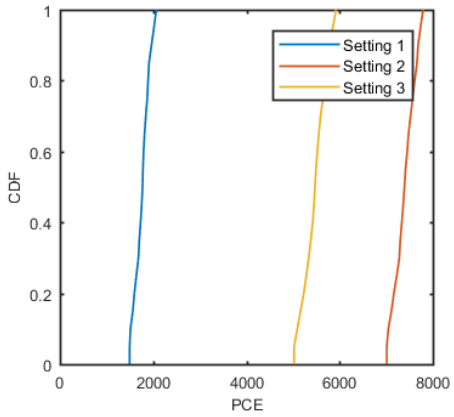


Figure 3.12. Probe signal detection. Setting 1: The presented QR code does not contain the probe signal. Setting 2: The presented QR code contains a probe signal and fingerprint removal is not performed on the captured image. Setting 3: The presented QR code contains a probe signal and fingerprint removal is performed on the captured image.

print, 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 (Hao, Zhou and Xing, 2012; Gohshi et al., 2005). 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. 3.12. 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.

3.6 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.

3.6.1 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 (Goljan, Fridrich and Filler, 2009).

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.

3.6.2 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 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.

3.6.2.1 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 (Armstrong, 2017), 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

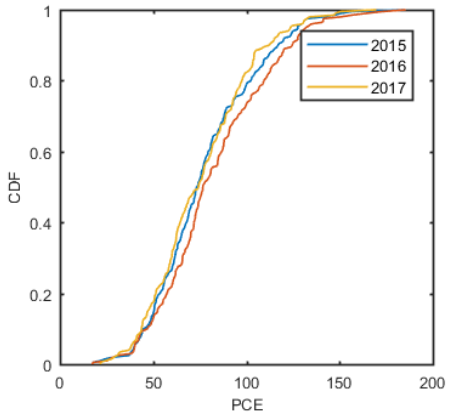
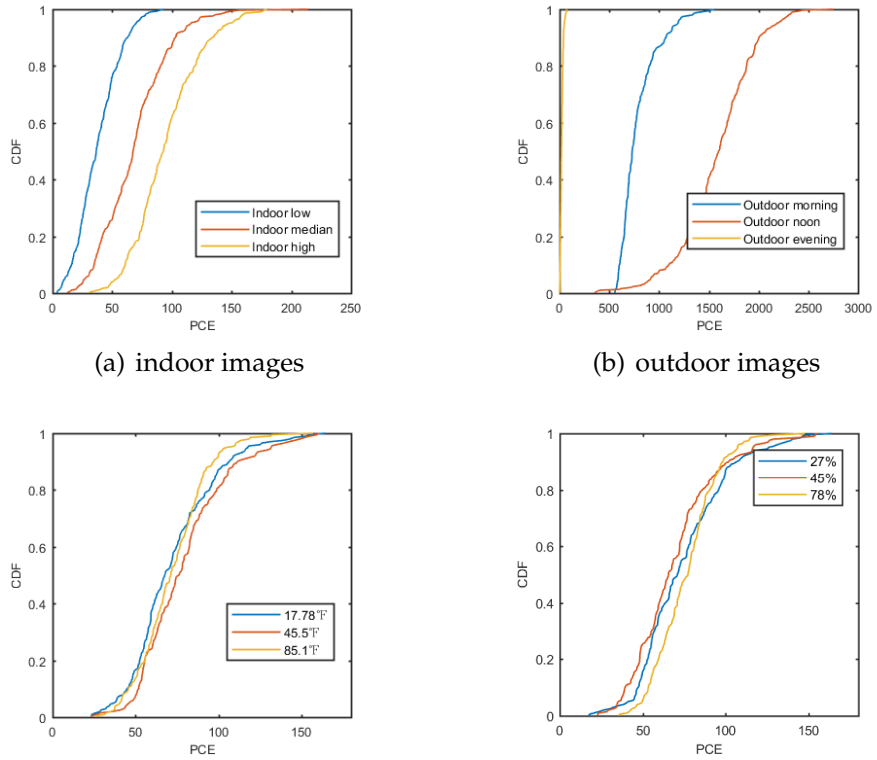


Figure 3.13. The impact of age. We use a reference image captured in 2017 and conduct fingerprint matching with images captured in different years. The CDF of each year shows the distribution of the PCEs obtained for that year.

200 images captured by the tested device. Fig. 3.13 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.

3.6.2.2 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



(a) indoor images (b) outdoor images

(c) Ambient temperature (d) Relative humidity

Figure 3.14. 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.

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. 3.14 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.

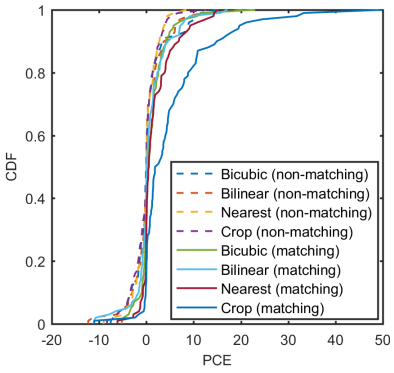
3.6.2.3 Impact of Ambient Temperature and Relative Humidity

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. 3.14(c) and Fig. 3.14(d), there is no significant difference between the CDF of different configurations. Therefore, PRNU is not affected by ambient temperature or relative humidity.

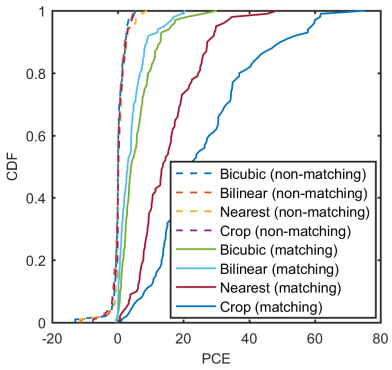
3.6.2.4 Impact of Image Resolution

Since the resolution of the image token significantly affects the overhead of the authentication process (Section 3.6.3) in terms of the time used for authentication, we now evaluate the fingerprint detection strategy on resizing images.

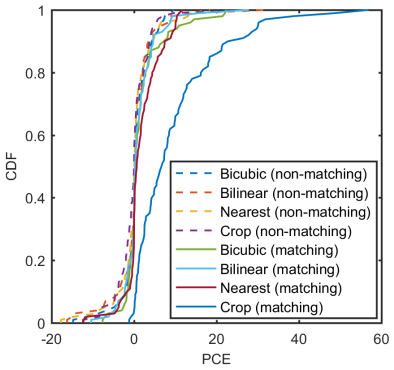
The images captured by a digital camera can be resized with down-sampling or image cropping. For down-sampling, we tested three most commonly used interpolation methods: *nearest-neighbor*, *bilinear*, and *bicubic*. For image cropping, we crop a rectangular area from the target image. After resizing an image, we also need to decide the image format to be used to store it. We test the two



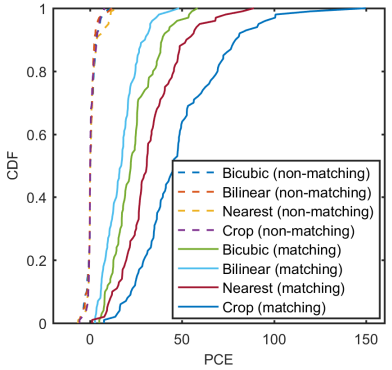
(a) 40% scaling rate, JPG



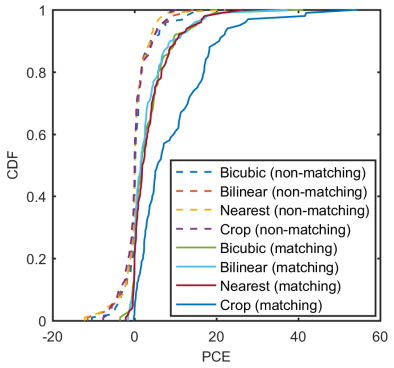
(b) 40% scaling rate, PNG



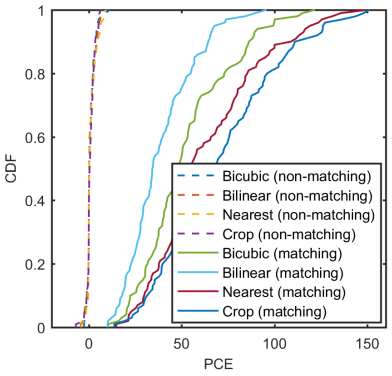
(c) 60% scaling rate, JPG



(d) 60% scaling rate, PNG



(e) 80% scaling rate, JPG



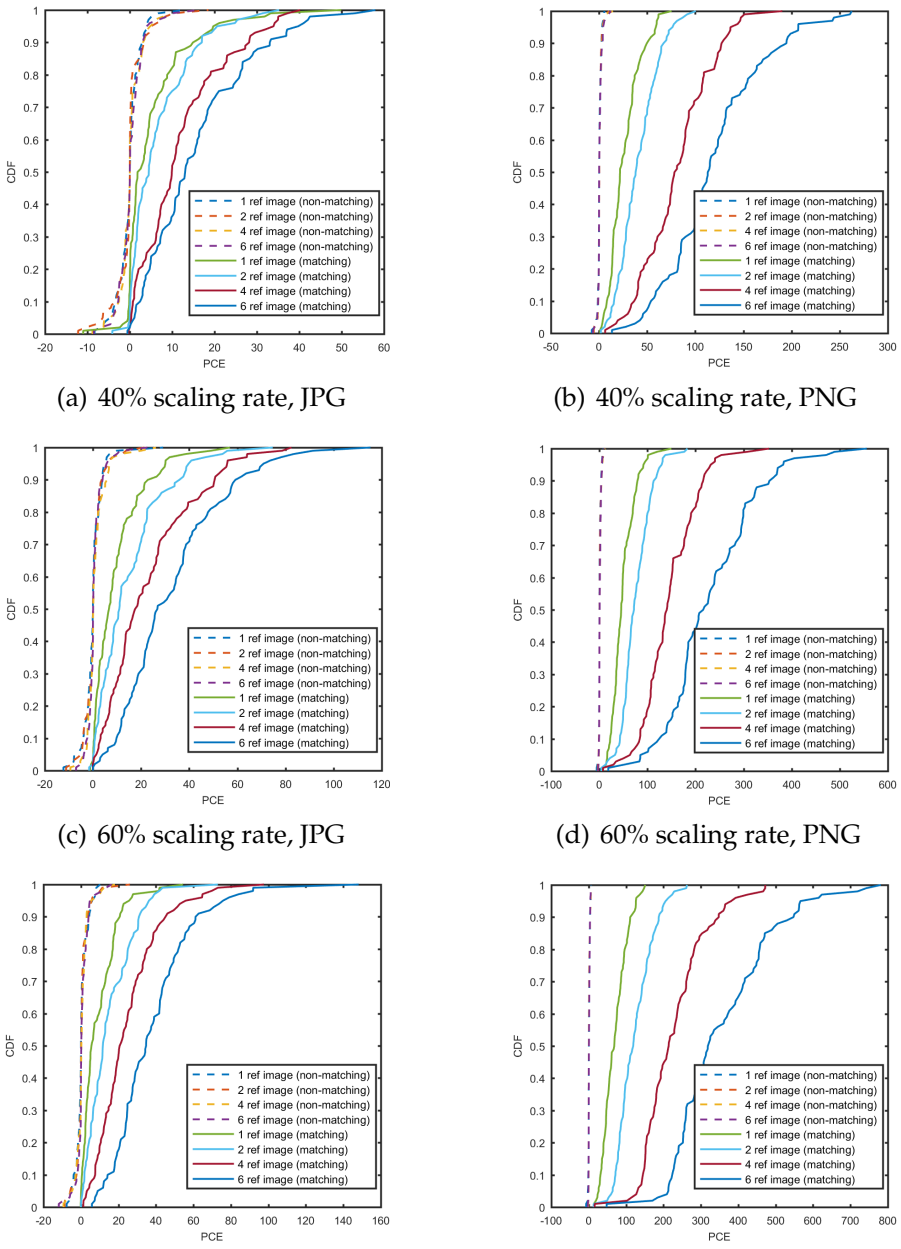
(f) 80% scaling rate, PNG

Figure 3.15. 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.

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. 3.15 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.



(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

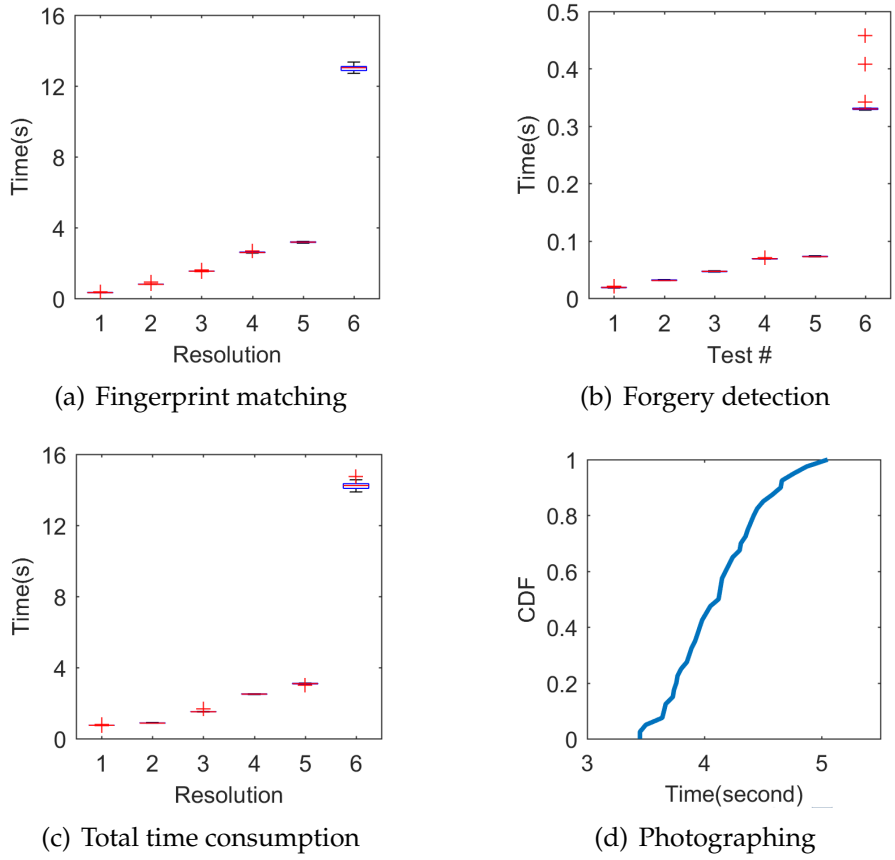
Figure 3.16. Impact of number of reference images. For every scaling ratio and image format, the PCE value obtained from a matching image pair is nearly proportional to the number of reference images.

3.6.2.5 Impact of the Number of Reference Images

For images with a low scaling ratio, one approach to improve the accuracy of fingerprint detection is to increase the number of reference images uploaded by

Table 3.3. Experimental settings for overall performance evaluation

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



(a) Fingerprint matching (b) Forgery detection

(c) Total time consumption (d) Photographing

Figure 3.17. Time overhead of the ABC protocol. The resolutions of the tested images are shown in Table 3.3.

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. 3.16 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.

3.6.3 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 3.3.

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 (Zhang et al., 2016), 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. 3.17(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. 3.17(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 3.3. As shown in Fig.3.17(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 (Goljan, Fridrich and Filler, 2009) does not take advantage of GPU computing and parallel computing. With further optimization, the efficiency would be significantly improved. Moreover, as shown in Sections 3.6.2.4 and 3.6.2.5, 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.

3.6.4 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. 3.17(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.

3.7 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*. (Dey et al., 2014) 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*. (Bojinov et al., 2014) 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*. (Das, Borisov and Caesar, 2016) 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 (Das, Borisov and Caesar, 2016, 2015).

Acoustic fingerprints can also be used to uniquely identify smartphones. Das *et al*. (Das, Borisov and Caesar, 2014*b*,*a*) extract auditory fingerprints from a process of playing and recording audio clips. Zhou *et al*. (Zhou et al., 2014) explore the speaker's frequency response to a specially designed audio input. Chen *et al*. (Chen et al., 2015) 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 (Das, Borisov and Caesar, 2015).

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 (Danev, Zanetti and Capkun, 2012), can be the source of the RF fingerprints. Different fingerprint sources include the clock jitter (Zanetti, Danev et al., 2010), device antenna (Danev, Heydt-Benjamin and Capkun, 2009), DAC sampling error (Polak, Dolatshahi and Goeckel, 2011), power amplifier non-linearity (Polak, Dolatshahi and Goeckel, 2011; Polak and Goeckel, 2011; Liu and Doherty, 2008), modulator sub-circuit (Brik et al., 2008), and the mixer or local frequency synthesizer (Toonstra and Kinsner, 1996).

Although hardware fingerprinting has been proved to be effective in track-

ing 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 (Chen et al., 2015; Danev et al., 2010).

3.8 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.

Chapter 4

Towards Practical Camera-based Smartphone Authentication via Camera Movement and Continuous Photographing

4.1 Introduction

Protecting users' on-line accounts has always been a challenging task. According to Breach Level Index (*Breach Level Index H1 2018 Infographic*, N.d.), over 3.3 billion data records were compromised in the first half of 2018, an increase of 72% compared to the first half of 2017. The exposed data includes names, social security numbers, passwords, fingerprints, and so on. Due to the fact that most users are prone to reuse their passwords across different web services, many organizations began to implement multi-factor authentication systems that not only prompt the users for what they know but also what they have.

Along this direction, we explore the camera-based smartphone authentication system which verifies the identity of a smartphone through checking the hardware fingerprint of its built-in camera (Ba et al., 2018; Valsesia et al., 2017). In such an authentication system, the user sends one or more images captured by his/her smartphone to a verifier, which authenticates the user by matching the fingerprint of the received images to the reference fingerprint of the legitimate device. Compared with other physical-layer authentication systems that either require additional hardware or suffer from poor usability, the camera-based authentication system is convenient and of low cost. For instance, a user can photograph and upload a transaction on a merchant's point of sale (POS) terminal to grant a payment. In another instance, a user can photograph a challenging scene (e.g., a QR code) displayed on a laptop to authorize a login. The usability of such authentication modality is preserved since taking photos is familiar and convenient to most smartphone users.

The essence of the camera-based authentication system is the detection of whether an image is captured by a specific smartphone. Photo Response Non-Uniformity (PRNU) has been recognized as the most reliable hardware fingerprint of digital cameras for image-to-camera matching in digital forensics (Lukas, Fridrich and Goljan, 2006). Unlike most hardware fingerprints that are composed of a few features drawn from the time domain and frequency domain of sensor outputs (Dey et al., 2014; Zhou et al., 2014; Brik et al., 2008), this camera fingerprint is a large matrix consisting of millions of variables, which makes the fingerprint of each individual camera remarkably unique. Results have shown that, the PRNU-based identification approach can accurately differentiate over one million images captured by thousands of devices (Goljan, Fridrich and Filler, 2009). Moreover, according to the experimental results in

(Ba et al., 2018), this fingerprint does not change overtime and is robust against most environmental changes, such as light, temperature and relative humidity. These salient features make the PRNU a good candidate for the physical layer proof of a smartphone.

The PRNU however is vulnerable against fingerprint forgery attacks. With a handful of images (e.g., on social media) from a victim smartphone, an adversary can extract the fingerprint of the victim device and embed the obtained fingerprint into arbitrary images of the same resolution (Goljan, Fridrich and Chen, 2011; Steinebach et al., 2010; Ba et al., 2018). Despite decades of research on camera fingerprinting, only few detection mechanisms have been proposed to detect forged fingerprints, and these mechanisms are either impractical or have security flaws. Goljan *et al*. (Goljan, Fridrich and Chen, 2011) detect forged images by tracking their abnormal correlation with the victim images used by the adversary. Their approach could achieve a high detection rate if the verifier knows the images used by the adversary. This is a strong assumption that is often hard to meet. Quiring *et al*. (Quiring and Kirchner, 2015) propose a fragile camera fingerprint that can only be extracted from raw images. Since most of the images shared online are in a compressed format such as JPG, the use of fragile fingerprints raises the bar for fabricating forged images. Unfortunately, this approach requires a large number of raw images to obtain a reliable fingerprint, and its performance relies heavily on the secrecy of raw images. In (Ba et al., 2018), it is found that forged images generated by an adversarial device share the fingerprints of both the victim device and the adversarial device. The similarity value between two forged images fabricated by the same adversarial device exceeds the normal range. Therefore, the authentication system proposed in (Ba et al., 2018) detects forged images by requiring that a user captures

two time-variant QR codes and sends the two images to a server. However, this mechanism fails if the adversary uses multiple adversarial devices to fabricate different images since images fabricated by different devices will only share the victim's camera fingerprint. The effectiveness of this attack is demonstrated in Section 4.3.4.

In this paper, we present new primitives for the PRNU forgery detection and introduce a novel and practical camera-based smartphone authentication system. Our major contribution can be summarized as follows.

We find that a noisechain is embedded in the random noise components of continuously captured images when a smartphone camera operates in burst mode. In burst mode, images are captured in a short interval, and the random noise components of an image can partly be preserved across multiple images. We refer to the preserved noise components as the short-term noise. It has a positive impact on the similarity value between nearby images and gradually attenuates with the increasing distance between images. The similarity values between a burst image and its nearby images always demonstrate an attenuation pattern, which is considerably sensitive to the injection of foreign fingerprints. Therefore, a noisechain-based forgery detector is built upon such observations.

Our second finding is there exist correlations between a *moving* smartphone camera in burst mode and the noise components of the captured image. The instantaneous velocity of the camera is positively correlated with the quality of the image's camera fingerprint and is negatively correlated with the Peak to Autocorrelation Energy (PAE) of the image's random noise components. Therefore, the verifier can challenge the user to upload images captured at a high movement speed. The images uploaded by the user will always have a strong cam-

era fingerprint and weak random noise components. A forgery attack will inevitably increase the noise PAE of the target image. A movement-based forgery detector is built upon such observations by checking the noise PAE of the images and their correlation with accelerometer readings. The movement-based detector works even if the user submits a single image for authentication. We do not find attacks against this one-image authentication system while we recommend the use of at least four images to take advantage of all forgery detectors. We are the first to observe the correlation between the movement of the photographing device and the fingerprint of the captured images. The universality of the observations are validated with 22 smartphones of 5 models.

Using the noisechain-based forgery detector and the movement-based forgery detector, we propose *CIM*, a camera-based smartphone authentication system that can defeat forgery attacks and is also user-friendly. *CIM* works as follows. When a user requests authentication, the verifier generates two fresh QR codes with random strings and displays them on its interface (e.g., a point-of-sell machine) simultaneously. The user then takes pictures of the QR codes in burst mode while moving the camera from the first QR code to the second one. The smartphone records the measurements of its accelerometer. After reaching the second QR code, the user stops the camera and uploads a certain number of captured images with the accelerometer readings to the verifier. The verifier authenticates the user as follows: 1) Detect replay attacks by checking the existence of the presented QR codes. 2) Detect foreign devices by matching the fingerprints of the received images to the reference fingerprint of the target smartphone. 3) Detect fingerprint forgery attacks through the noisechain-based forgery detector and movement-based forgery detector. Extensive experiments are conducted to evaluate the security of the proposed system under various

settings, including fingerprint forgery attacks and advanced adversaries. *CIM* achieves 100% True Acceptance Rate(TAR) at 0% False Acceptance Rate(FAR) in both fingerprint matching and forgery detection.

4.2 Related Work

Hardware Fingerprinting: In the literature, a rich set of sensors have been explored for identification of smartphones (Baldini and Steri, 2017; Amerini et al., 2017). Motion sensors like accelerometers and gyroscopes have been demonstrated to have unique statistical features (Dey et al., 2014; Das, Borisov and Caesar, 2016) and calibration errors (Das, Borisov and Caesar, 2016; Son et al., 2018). Acoustic sensors like speakers and microphones can be identified by the frequency response (Zhou et al., 2014) and auditory features (Das, Borisov and Caesar, 2014*a*). Image sensors on smartphone cameras exhibits non-uniform sensitivity to light (Lukas, Fridrich and Goljan, 2006; Fridrich, 2009*a*). For wireless transmitters (Danev, Zanetti and Capkun, 2012), their fingerprints come from the clock jitter (Jana and Kasera, 2010; Zanetti, Danev et al., 2010), device antenna (Danev, Heydt-Benjamin and Capkun, 2009), DAC (Polak and Goeckel, 2015), power amplifier (Polak and Goeckel, 2015), and modulator (Brik et al., 2008; Zhuang et al., 2018). Most of these methods have been demonstrated to be effective in the context of device tracking.

Fingerprint Forgery: Under adversarial settings like authentication and forensics, most of the above mentioned mechanisms are vulnerable to replay attacks and fingerprint forgery attacks. Das *et al*. (Das, Borisov and Caesar, 2016; Das, Borisov and Chou, 2018) and Hupperich *et al*. (Hupperich et al., 2015) show that the fingerprints of motion sensors are manipulatable. Goljan

et al. (Goljan, Fridrich and Chen, 2011) demonstrated the feasibility of injecting one camera's fingerprint into the image captured by another device. The works of (Danev et al., 2010; Edman and Yener, 2009; Rehman, Sowerby and Coghill, 2014; Fang, Liu and Ning, 2016) show that the fingerprints of wireless transmitters are vulnerable to impersonation attacks. There is very limited research on countermeasures to these fingerprint forgery attacks.

Hardware-rooted Smartphone Authentication: Recently, the explosive growth of data breaches has led to a renewed interest in hardware-rooted authentication modalities (Enterprise, 2017). Van *et al*. (Van Goethem et al., 2016) proposed to authenticate wireless devices by checking the statistical features of accelerometers. Chen *et al*. (Chen, Zhang, Qin, Mao, Qin, Shen and Li, 2017) implemented an authentication system using the frequency response of acoustic sensors. Ba *et al*. (Ba et al., 2018) and Valsesia *et al*. (Valsesia et al., 2017) explore the PRNU as the unique identities of smartphones. The works of (Zeng et al., 2011; Zeng, Govindan and Mohapatra, 2010; Jiang et al., 2013; Sharaf-Dabbagh and Saad, 2016; Patel, 2015; Xiong and Jamieson, 2013) authenticate wireless devices by checking the fingerprints of their transmitter and/or the wireless fading channel. Most of these works however suffer from either poor usability or are vulnerable to **fingerprint forgery attacks**.

Toward this end, we present new primitives for the forgery detection of camera fingerprint and propose *CIM*, a camera-based smartphone authentication system that can defeat fingerprint forgery attacks while preserving usability. Our work differs from existing methods in the sense that it exploits the camera fingerprint of burst-mode images and the correlation between the camera fingerprint and the camera movement.

4.3 Preliminary

In this section, we introduce a typical camera-based smartphone authentication system with the Photo Response Non-Uniformity (PRNU) and our threat model. We then introduce smartphone camera fingerprinting, particularly how PRNU is extracted from images as the fingerprint of a smartphone camera. At last, we describe two fingerprint forgery attacks which will be addressed in this paper.

4.3.1 Camera-based Smartphone Authentication

Fig. 4.1 shows the typical architecture for camera-based smartphone authentication. The *user* sends out a request (e.g., a payment request) to the verifier and needs to be authenticated. The *verifier* consists of an interface and a server that interact with the smartphone. The server maintains a database of <user name, reference fingerprint, smartphone model>. In the authentication process, the user takes pictures of the verifier's interface and uploads the captured images to the *server*. That is, the communication from the verifier's interface to the smartphone uses a visible light communication (VLC) channel. The verifier then authenticates the user by checking the camera fingerprint extracted from the received images, such as PRNU introduced below. The communication from the smartphone to the server is through a wireless channel.

4.3.2 Threat Model

We consider a powerful adversary (Ba et al., 2018). The adversary has prior knowledge of the victim's credential, including the user name, smartphone model, and camera fingerprint. The camera fingerprint can be extracted from

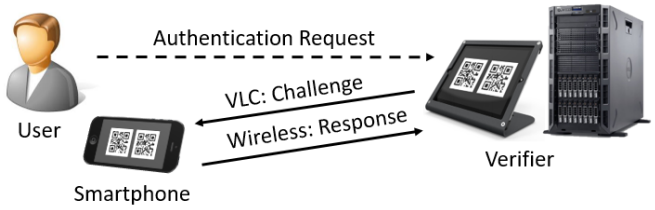


Figure 4.1. System model. The user initiates the authentication process on the verifier's interface. The user then captures what is shown on the screen and uploads the captured image to the verifier. The verifier determines the identity of the user through checking the fingerprint on the received image.

the online images posted by the victim. The adversary may also know the detailed setting of our authentication protocol and is able to collect any kind of images captured by the victim smartphone. However, the adversary does not physically possess the victim's smartphone during the authentication process.

The objective of the adversary is to impersonate a legitimate user using a *foreign* smartphone. The adversary initiates the authentication with his/her smartphone and submits images carrying the victim's camera fingerprint, in the hope of fooling the verifier into believing that the smartphone is the one associated with the legitimate user. In particular, the adversary may conduct two kinds of attacks. In a *replay attack*, the adversary collects images carrying the victim's camera fingerprint and submits them to the verifier. Such images can easily be obtained from the victim's social network. In a *fingerprint forgery attack*, the adversary takes pictures of the verifier's interface and embeds the captured images with the victim's camera fingerprint. In this attack, the adversary can fabricate arbitrary images carrying the victim's camera fingerprint.

4.3.3 Smartphone Camera Fingerprinting

Photo Response Non-Uniformity (PRNU) is a very reliable hardware fingerprint of digital cameras (Lukas, Fridrich and Goljan, 2006; Fridrich, 2009*a*). It is a multiplicative factor to the actual optical view and originates from the non-uniform light-sensitivity of millions of pixels. Denote $\mathbf{K}$ as the PRNU fingerprint of a digital camera. An image captured by the camera can be represented as $\mathbf{I} = (1 + \mathbf{K})\,\mathbf{I}^0 + \Theta$, where $\mathbf{I}^0$ and Θ represents the actual optical view and the random noise components respectively.

To determine if a query image $\mathbf{I}_q$ carries the PRNU fingerprint of a smartphone camera, the verifier needs to correlate the fingerprint of the query image to the reference fingerprint of the target camera. Specifically, the verifier first extracts the camera fingerprint of $\mathbf{I}_q$ using a denoising filter. Because the PNRU fingerprint behaves like a white Gaussian noise (Lukas, Fridrich and Goljan, 2006; Chen, Fridrich and Goljan, 2007), the obtained noise residue can be modeled as $\mathbf{W}_q = \mathbf{I}_q^0\mathbf{K}_q + \Xi_q$ (Chen et al., 2008), where Ξ_q is a combination of other white Gaussian noises of the query image. The verifier then prepares a reference fingerprint $\hat{\mathbf{K}}$ of the target device. Due to the fact that smartphone cameras possess strong fingerprints, the reference fingerprint can be the noise residue of an image $\mathbf{I}_r$ taken by the target smartphone (Ba et al., 2018). Finally, the verifier evaluates the similarity between $\mathbf{W}$ and $\hat{\mathbf{K}}$ using their Peak to Correlation Energy (PCE) (Goljan, 2008):

$$\begin{aligned} \rho &= PCE\left(\mathbf{W}_q, \hat{\mathbf{K}}\right), \\ &= PCE\left(\mathbf{I}_q^0\mathbf{K}_q + \Xi_q, \mathbf{I}_r^0\mathbf{K}_r + \Xi_r\right). \end{aligned} \tag{4.1}$$

The obtained PCE value is determined by the strength of the noise components

shared between $\mathbf{W}_q$ and $\hat{\mathbf{K}}$. It will be significantly higher if $\mathbf{K}_q$ equals $\mathbf{K}_r$. The verifier uses a threshold to determine if the query image is indeed captured by the target smartphone.

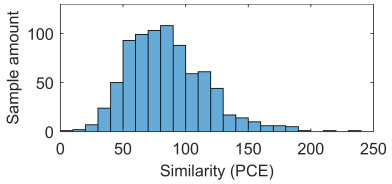
To illustrate PCE, we construct 800 matching image pairs and 800 non-matching image pairs using iPhone 6 and conduct fingerprint matching. The obtained PCE distributions are shown in Fig. 4.2. It can be observed that the PCE values of matching image pairs are almost always higher than that of non-matching image pairs. The reason why there is a small overlap between those two distributions is because of the existence of the random noise components (Ξ_r). In practice, the strength of Ξ_r can be suppressed through averaging the noise residues of multiple images (Cain, Hayat and Armstrong, 2001). With a high quality fingerprint estimated from multiple images, the PCE values obtained from matching images can be significantly improved.

4.3.4 Fingerprint Forgery

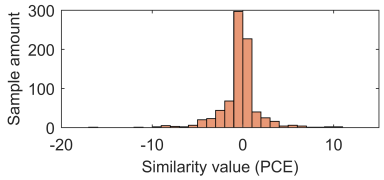
There are two strategies to modify the fingerprint of an image: *Quick Injection* (Goljan, Fridrich and Chen, 2011) and *Fingerprint Replacement* (Ba et al., 2018).

In *quick injection*, the adversary directly injects a victim's camera fingerprint into an image captured by a foreign device. Specifically, the adversary first estimates a fingerprint $\hat{\mathbf{K}}_V$ from multiple images captured by the victim smartphone. The adversary then injects the obtained estimation into an image $\mathbf{J}$ captured by a foreign smartphone by

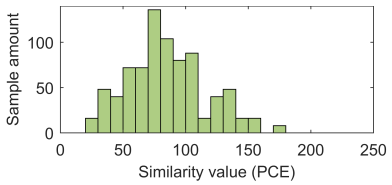
$$\begin{aligned} \mathbf{J}' &= \left(1 + \alpha\hat{\mathbf{K}}_V\right)\mathbf{J}, \\ &\approx \left(1 + \alpha\hat{\mathbf{K}}_V\right)\left(1 + \mathbf{K}_A\right)\mathbf{J}^0, \end{aligned} \tag{4.2}$$



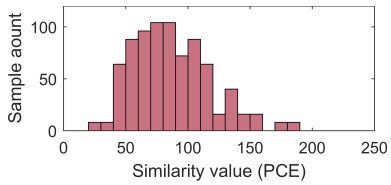
(a) Matching image pairs: The PCE between two images captured by the same smartphone



(b) Non-matching image pairs: the PCE between two images captured by different smartphones



(c) Quick injection attacks: the PCE between a victim image and a forged image



(d) Forged image pairs: the PCE between two forged images fabricated by different adversarial devices

Figure 4.2. PCE distributions of different image pairs. (a) and (b): Images captured by the same smartphone show significantly higher PCE than images captured by different smartphones. (c) In quick injection attacks, the forged image can easily bypass PRNU-based camera identification because the PCE between a forged image and a victim image lies in a similar range as the PCE from matching image pairs (two victim images). (d) For forged images fabricated by different adversarial devices, their PCE also lies in a similar range as the PCE from matching image pairs.

where $\mathbf{K}_A$ is the inherent fingerprint of $\mathbf{J}$ and α is the strength factor that controls the injected fingerprint $\hat{\mathbf{K}}_V$. With a proper α, the adversary can easily adjust the strength of $\mathbf{K}_V$ to the legitimate range (Fig. 4.2(c)).

In a recent work, a forgery detection mechanism is proposed to detect quick injection attacks by challenging the user/adversary to provide two images and checking the PCE between the noise residues of the received images (Ba et al., 2018). If these two images are fabricated by the *same* adversarial device, the obtained PCE will exceeds the normal range because of $\mathbf{K}_A$. However, as shown in Fig. 4.2(d), forged images fabricated by *different* adversarial devices will only share the victim's camera fingerprint and have a similarity value within the

normal range. ABC fails in this case.

In *fingerprint replacement*, the adversary removes the inherent fingerprint of the target image before injecting the victim's camera fingerprint. Given an image $\mathbf{J}$ captured by a foreign device, the adversary first extracts its noise residue $\mathbf{W}_A$ using a denoising Filter. He/She then removes $\mathbf{W}_A$ from $\mathbf{J}$ by

$$\begin{aligned} \mathbf{J}'^0 &= (1 - \beta \mathbf{W}_A)\, \mathbf{J}, \\ &\approx \left(1 - \beta \mathbf{J}^0 \mathbf{K}_A\right) (1 + \mathbf{K}_A)\, \mathbf{J}^0, \end{aligned} \tag{4.3}$$

where β controls the strength of the removed fingerprint. By varying β, the adversary adjusts the strength of $\mathbf{K}_A$ to a negligible level. Finally, the adversary injects the victim's camera fingerprint into the sanitized image $\mathbf{J}'^0$ using equation 4.2. Images fabricated in this way will only contain the fingerprint of the victim smartphone and are harder to detect.

4.4 Intuition and Validation

The key challenge for a practical camera-based smartphone authentication system is to detect the quick injection attack and the fingerprint replacement attack described in section 4.3.4. Our system addresses these attacks through employing a specially designed photographing strategy which greatly increases the difficulty of fabricating forged images without leaving traces. Specifically, upon receiving a user's authentication request, the verifier requires the user to take pictures in burst mode and to involve a simple movement into the photographing process. Both the captured images and the measurements of the smartphone's accelerometer need to be uploaded for authentication. The verifier then detects

quick injection attacks and fingerprint replacement attacks through checking forgery-sensitive features unique to burst images and correlations between the captured images and the camera movement. In this section, we describe the features and correlations to be applied in our system, illustrate their sensitivity to each kind of forgery attack, and provide experimental validations.

4.4.1 Intuition

We have observed two novel phenomena that can be used to address fingerprint forgery attacks.

Noisechain in images taken in burst mode. Images in burst mode are captured in quick succession. Short-term noise is shared between sequentially captured images. Such short-term noise may form a noisechain linking the random noise components of burst images. This is counter-intuitive because it is often believed that random noise components of unmodified images (i.e. images that are not manipulated by an adversary) are always random and independent. Therefore, during the authentication process, the user shall take images in burst mode and the noisechain can be used to detect forgery attacks.

Correlation between camera movement and noise. A shaking camera may cause blurry images that a photographer wants to avoid. However, the distortion caused by movement is systematic, and it is likely to create correlations between the noise components of the captured image and the instantaneous velocity of the photographing device. Therefore, during the authentication process, a user may shake the smartphone while taking images and the correlation between camera movement and noise in images may be utilized to defeat forgery attacks.

In the rest of this section, we present our validation of the two phenomena

Table 4.1. Devices under investigation

Smartphone Model	Burst Rate	Image Resolution
Samsung S8	30 FPS	1440×2560
iPhone 6	10 FPS	2448×3264
LG G5	10 FPS	2976×2976
Samsung J3	3 FPS	1536×1536
Moto G4	3 FPS	1836×3264

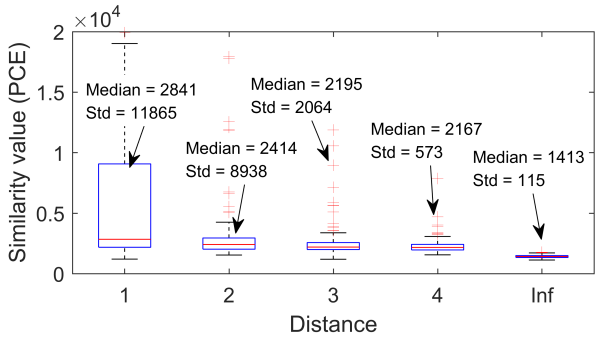


Figure 4.3. PCE distributions of burst image pairs. The distance between two images refers to the difference between their position in the burst. A distance of 1 indicates that the image pair contains two continuously captured images. A distance of Inf indicates that the image pair is unconnected. Each of the distribution is obtained from 800 image pairs captured by iPhone 6.

and also validate that any modification to the camera fingerprints from burst images will break the integrity of the noisechain as well as the correlations with movement. For each phenomenon, We employ 22 smartphones of 5 different models for evaluation: i) 10 iPhone 6; ii) 3 Samsung Galaxy S8; iii) 3 LG G5; iv) 3 Samsung J3; v) 3 Moto G4. The technical specifications of the smartphones are shown in Table 4.1. These smartphones cover all burst rates available in the market.

4.4.2 Existence of Noisechain

To demonstrate the existence of the noisechain, we compare two types of image pairs: 1) Connected image pair: two images in the same burst. The images are

expected to share both the camera fingerprint and the short-term noise if the noisechain exists. 2) Unconnected image pair: two images from different bursts but captured by the same smartphone. The only noise component shared between the selected images shall be the camera fingerprint. Please note that, the images applied this experiment are captured by hand-held cameras. No two images in the same burst have identical image content. Because the image content will only affect the PCE between two images when there exists a region in the two images that are identical and perfectly aligned, the PCE value obtained in this experiment is not caused by similar image content.

For each image pair, we extract the noise residues of both images and calculate their PCE value. The box plots of the PCE values are shown in Fig. 4.3. A box plot is often used to display the distribution of data. The distance between two images refers to the difference between their position in a burst. Each distribution is obtained from 800 image pairs captured by iPhone 6. It can be observed that the PCE distributions of connected image pairs have significantly higher median and standard deviation than the distribution of unconnected image pairs. With the increasing distance, the distribution of connected image pairs gets closer to the distribution of unconnected image pairs. Since the strength of the camera fingerprint is similar for all types of image pairs, these provide convincing evidence that there are short-term noises affecting the PCE values obtained from connected image pairs. Such short-term noise can be partially preserved across multiple images captured in a row, and the strength of the preserved part gradually decreases with the increasing distance. In other words, there is a noisechain embedded in the random noise components of continuously captured burst images.

Universality of Noisechain: We now repeat the above experiments with

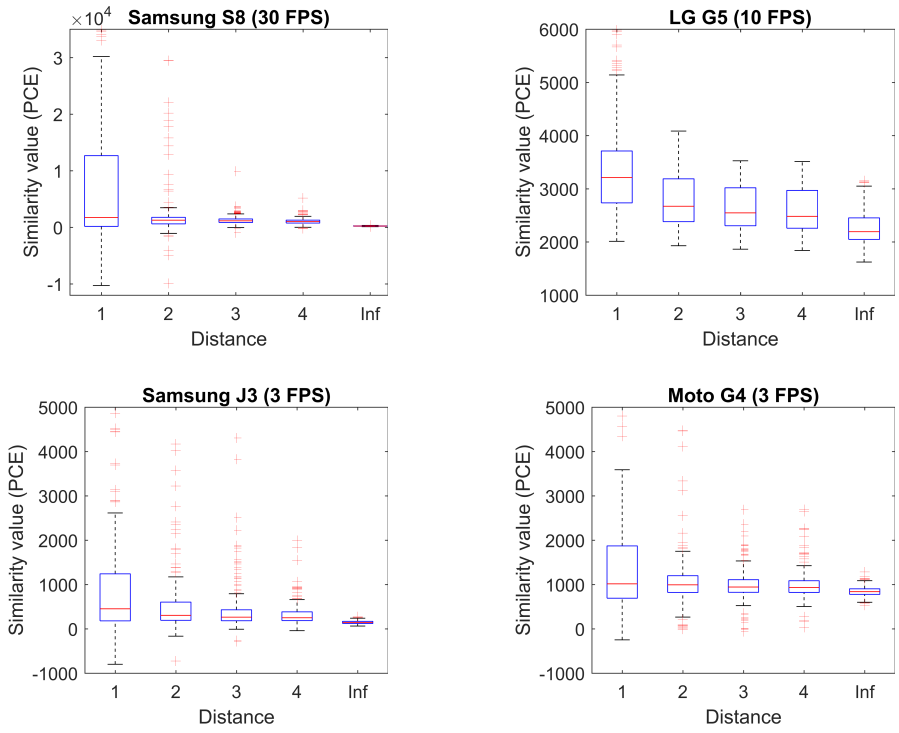


Figure 4.4. Universality of the noisechain. Each of the distribution is obtained from 800 image pairs.

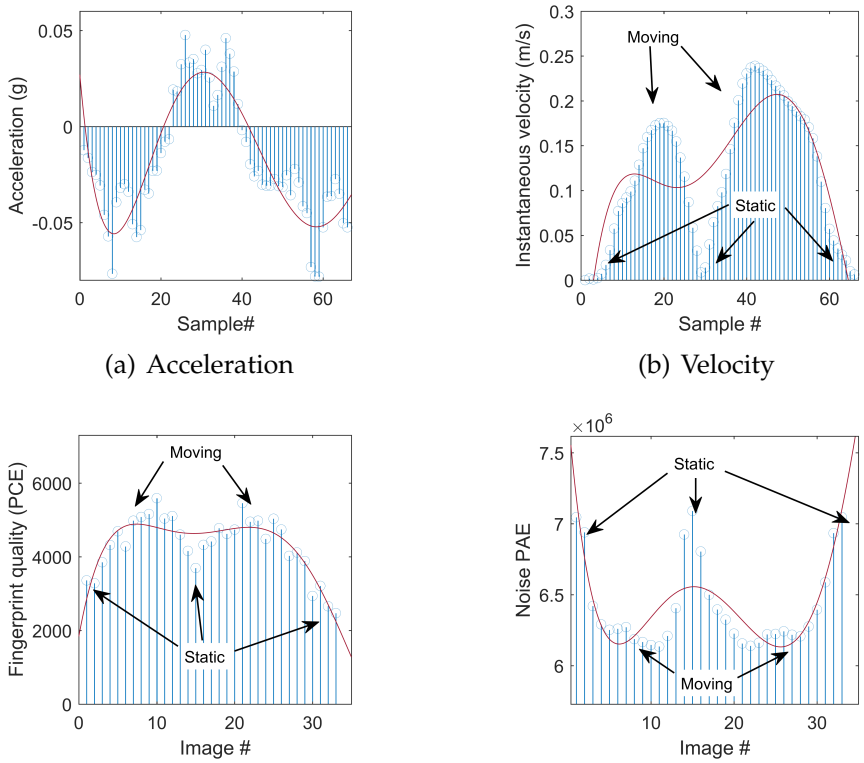
the four other smartphone models. The distribution of the obtained PCE values are shown in Fig. 4.4. We make following observations: 1) The noisechain is universal for all smartphone models under investigation. Regardless of the burst rate of the smartphone, the PCE distribution of connected image pairs always has a higher median and standard deviation than the distribution of unconnected image pairs. 2) For smartphone models with a higher burst rate, the standard deviation of their PCE distribution drops faster with the distance between images. This result indicates that the impact of the short-term noise increases with the burst rate of the smartphone. 3) The standard deviation of the PCE distributions obtained from LG G5 is relatively robust across different types of image pairs. This is possibly because LG G5 has employed many post processing techniques on the captured images. According to our experiment

results, most post-processing operations suppress the short-term noise shared between continuously captured images. Such robust standard deviation has a positive impact on the performance of the noisechain-based forgery detector.

4.4.3 Correlation between Movement and Noise

Our experiment results clearly show two correlations between an image's noise components and the movement of the camera. We now use one series of burst images to help illustrating the two correlations. The universality of these correlations will be demonstrated later. The burst series contains 33 burst images captured by an iPhone 6. During the photographing process, we first hold the smartphone still, and then move the smartphone to the right side. After a short stay, we move the smartphone back to the left side and then terminate the photographing process. Fig. 4.5(a) plots the smartphone's accelerometer readings along the moving direction (the X-axis). To visualize the movement pattern, we calibrate the obtained accelerometer readings (Das, Borisov and Caesar, 2016) and calculate the instantaneous velocity at each sample point. As shown in Fig. 4.5(b), the movement of the smartphone mainly involves five stages: static, move to the right side, static, move to the left side, and static.

Correlation with the camera fingerprint: The first feature affected by the camera movement is the fingerprint quality of the captured image. Although the PRNU of an image sensor remains constant overtime, the fingerprint signal on an image can be distorted by the image's random noise components (e.g., environmental noise and image content). Fingerprint quality is used to measure the similarity between the fingerprint of an image and the real fingerprint of the smartphone camera. A high fingerprint quality indicates that the finger-



(a) Acceleration (b) Velocity

(c) Fingerprint quality (d) Noise PAE

Figure 4.5. A burst series captured by an iPhone 6. The burst rate of the camera is 10 FPS, and the sampling rate of the accelerometer is 20 Hz.

print of the image is less distorted and that image is more likely to be accurately matched to its photographing device. In practice, the fingerprint quality of an image is normally estimated through calculating the PCE value between the image's noise residue and a high quality reference fingerprint of the photographing device. Fig. 4.5(c) plots the PCE values between each test image and a strong reference fingerprint estimated from five images. Please note that, although we use five images to extract the reference fingerprint for the purpose of a clear illustration, one image is enough for our camera-based authentication system *CIM*. It can be observed that, the fingerprint quality of an image is significantly and positively correlated with the instantaneous velocity of the

photographing device. This is because the images captured at high movement speed are more blurry than images captured at low movement speed. Camera fingerprints extracted from blurred images are less distorted by the image content and therefore have higher quality. We refer to the correlation between the fingerprint quality and the camera movement as the **fingerprint-movement correlation**.

Correlation with the noise residue: The second feature affected by the camera movement is the Peak to Autocorrelation Energy (PAE) of the image's noise residue. The noise residue of an image consists of the camera fingerprint and random noise components (e.g., environmental noise and image content). Given a noise residue **W** of size m*n, the PAE can be calculated as:

$$\begin{aligned} PAE(\mathbf{W}) &= PCE(\mathbf{W},\mathbf{W}) \\ &= \frac{(\mathbf{W}-\overline{\mathbf{W}})\cdot(\mathbf{W}(s_{peak})-\overline{\mathbf{W}(s_{peak})})}{\frac{1}{mn-|N|}\sum_{s\notin N}(\mathbf{W}-\overline{\mathbf{W}})\cdot(\mathbf{W}(s)-\overline{\mathbf{W}(s)})}, \end{aligned} \tag{4.4}$$

where $(\mathbf{W}-\overline{\mathbf{W}})\cdot(\mathbf{W}(s)-\overline{\mathbf{W}(s)})$ is the dot product between $\mathbf{W}-\overline{\mathbf{W}}$ and $\mathbf{W}(s)-\overline{\mathbf{W}(s)}$ circularly shifted by vector s. N is a small neighborhood around the peak (Goljan, Fridrich and Filler, 2009). Because the peak for autocorrelation is always occurs as $s_{peak}=[0,0]$, the equation 4.4 can be rewritten as

$$PAE(\mathbf{W}) = \frac{Var(\mathbf{W})}{\frac{1}{mn-|N|}\sum_{s\notin N}(\mathbf{W}-\overline{\mathbf{W}})\cdot(\mathbf{W}(s)-\overline{\mathbf{W}(s)})}, \tag{4.5}$$

where $Var(\mathbf{W})$ is the variance of the noise residue **W**. Fig. 4.5(d) plots the PAE value of each test image. It can be observed that, the noise PAE of an image is negatively correlated with the instantaneous velocity of the photographing device as well as the fingerprint quality. The reason are twofold. Firstly, the

noise residue of images captured at high movement speed contain less image content than normal images, which decreases the variance of the noise residue. Secondly, the movement of the smartphone introduced periodic signals into the captured image and increased the denominator in equation 4.5. We refer to the correlation between the noise PAE and the camera movement as the **noise-movement correlation**.

Universality of correlations with movement: We now demonstrate the universality of the fingerprint-movement correlation and the noise-movement correlation. We test two types of images for comparison: 1) Moving image: a burst image captured with a moving camera. 2) Static image: a burst image captured with a stationary camera. We test 200 moving images and 200 static images for each smartphone model under investigation.

Fig. 4.6 shows the distributions of the noise PAE and the fingerprint quality of the tested images. It can be observed that, in most cases, the fingerprint quality of a moving image is significantly higher than the fingerprint quality of a static image, and the opposite is true for the noise PAE of these images. The fingerprint-movement correlation and the noise-movement correlation are both universal for all tested smartphone models. For several smartphone models, there exist a small overlap between the distributions of moving images and static images. This is because both the fingerprint quality and the noise PAE vary from one image to another. Fortunately, such overlap only exists in the distributions constructed across different burst series.

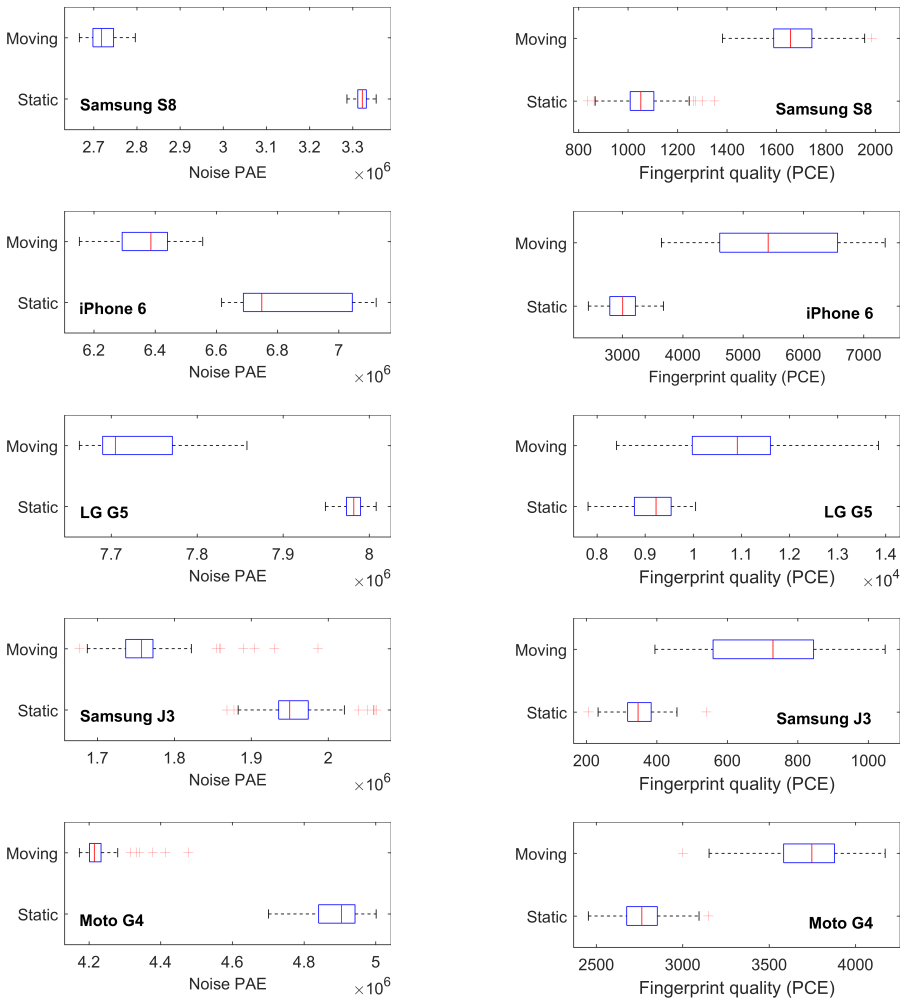


Figure 4.6. Noise-movement correlation and fingerprint-movement correlation. The fingerprint quality is estimated using a reference fingerprint extracted from five burst images.

4.4.4 Sensitivity to Fingerprint Forgery Attacks

To illustrate the impact of fingerprint forgery operations on burst images, we construct three series of burst images for comparison: 1) *Normal burst series*: an unmodified burst series containing 33 images captured in a row. We use the burst series shown in Fig. 4.5. 2) *Injection burst series*: a forged burst series fabricated through the quick injection strategy described in section 4.3.4. 3) *Replacement burst series*: a forged burst series fabricated through the fingerprint re-

placement strategy described in section 4.3.4. We first generate a sanitized burst series through removing the inherent fingerprints of the normal burst series. We then inject a foreign fingerprint into a sanitized burst series. The injection strength (α) in Equation 4.2 is set to be 0.03 so that the forged image will have a victim fingerprint within the normal quality range. For fingerprint removal, the strength factor (β) in Equation 4.3 is set to be 0.02 to completely eliminate the image's inherent fingerprint.

Each victim fingerprint is extracted from a training set of five images. The victim fingerprints injected into different images are always be extracted from different and non-overlapping training sets. To explain why, consider an adversary using two training sets $S1$ and $S2$ that both contain an image $\mathbf{I}$. In a forgery attack, the adversary fabricates two forged images $\mathbf{J}_1'$ and $\mathbf{J}_2'$ using respectively the victim fingerprint $\mathbf{K}_{A1}$ extracted from S_1 and the victim fingerprint $\mathbf{K}_{A2}$ extracted from S_2. Because an extracted fingerprint will also carry the random noise components of the applied victim images. $\mathbf{K}_{A1}$ and $\mathbf{K}_{A2}$ will both carry the random noise components of $\mathbf{I}$, making the PCE between $\mathbf{J}_1'$ and $\mathbf{J}_2'$ higher than a normal value. As a result, the adversary is exposed. To demonstrate this phenomenon, we conduct experiments with image sets of five different sizes: 5, 10, 15, 20, and 30. For each setting, we construct 3 image sets $S1$, $S2$ and $S3$ using images captured by the same iPhone 6. $S1$ and $S2$ are non-overlapping image sets. $S2$ and $S3$ have 20% common images. We then extract $\mathbf{K}_{A1}$, $\mathbf{K}_{A2}$ and $\mathbf{K}_{A3}$ from the three image sets and calculate $PCE(\mathbf{K}_{A1}, \mathbf{K}_{A2})$ and $PCE(\mathbf{K}_{A3}, \mathbf{K}_{A2})$. Figure 4.7 plots the PCE values obtained from different settings. It can be observed that, due to the existence of common images, the PCE value between $\mathbf{K}_{A3}$ and $\mathbf{K}_{A2}$ is always significantly higher than that between $\mathbf{K}_{A2}$ and $\mathbf{K}_{A1}$.

Noisechain: We construct connected image pairs from each burst series and

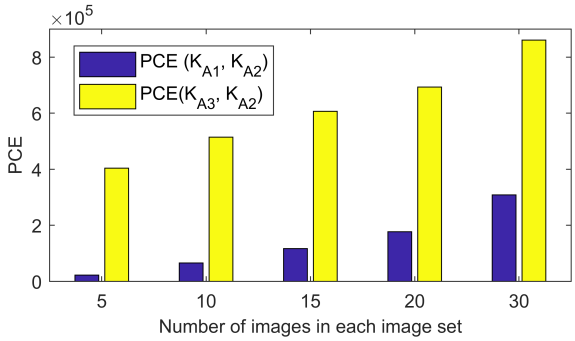


Figure 4.7. The impact of common images on the similarity between fingerprint estimations. $\mathbf{K}_{A1}$ and $\mathbf{K}_{A2}$ are estimated from non-overlapping image sets. The image sets applied to estimate $\mathbf{K}_{A2}$ and $\mathbf{K}_{A3}$ have 20% common images

calculate their PCE values. Fig. 4.8 shows the heat map of the PCE values obtained from the first ten images of each burst series. It can be seen that the heat map of the normal burst series has an *attenuation pattern*, in which the PCE values gradually decrease with the increasing distance. For the injection burst series and the replacement burst series, there is no such attenuation pattern in their heat maps. This is because the fingerprints injected into different images are derived from different image sets and thus have non-uniform fingerprint quality. The injection of these non-uniform fingerprints severely distorts the *attenuation pattern* of the target burst series' heat map.

The second observation is that the PCE values obtained from forged burst series are significantly higher than the PCE values obtained from the normal burst series. For the injection burst series, this phenomenon is caused by the extra camera fingerprint carried in forged images. For the replacement burst series, this phenomenon is caused by the introduction of a new noise component. Following the fingerprint replacement strategy, the adversary removes the inherent fingerprints of target images before injecting the collected fingerprints, in the hope of decreasing the PCE values between forged images. However, as will

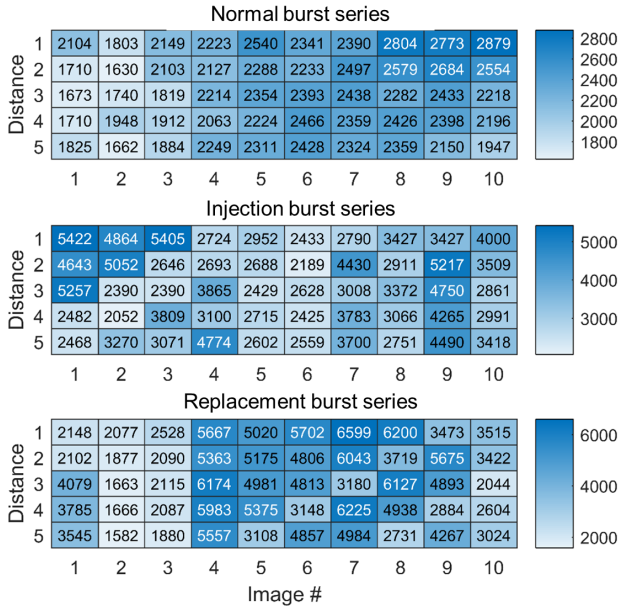
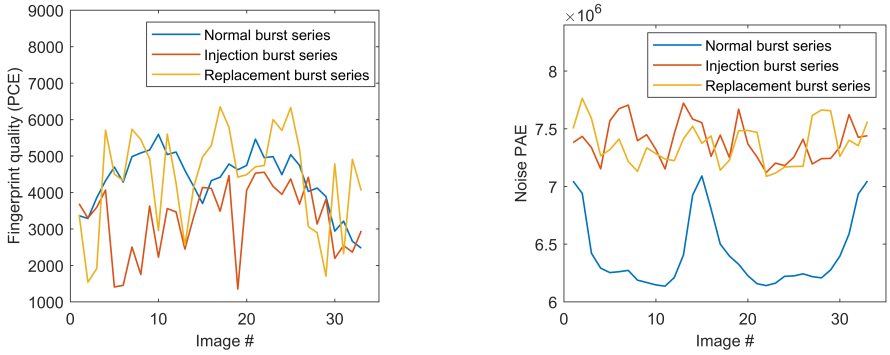


Figure 4.8. The PCE values of connected image pairs selected from different burst series. For each image pair, the image # is the sequential number of the first image, and sum of the image # and the distance is the sequential number of the second image.

be shown in Section 4.6.5, removing the fingerprint of an image using adaptive-denoising will introduce the negation of that image's short-term noise into the sanitized image. This noise component is shared between forged images and causes the PCE values between forged images to be significantly higher than that between normal images.

Our last observation is that the noisechain is also affected by the movement of the photographing device. A comparison between Fig. 4.8 and Fig. 4.3 reveals that the standard deviation of the PCE values in Fig. 4.8 is much lower than the standard deviation in Fig. 4.3. This is because the former experiment involves camera movement. According to our experiment results, the standard deviation of the strength of the short-term noise sharply decreases with the increasing instantaneous velocity of the camera. In this paper, we refer to this phenomenon as the **noisechain-movement correlation**. As will be shown in



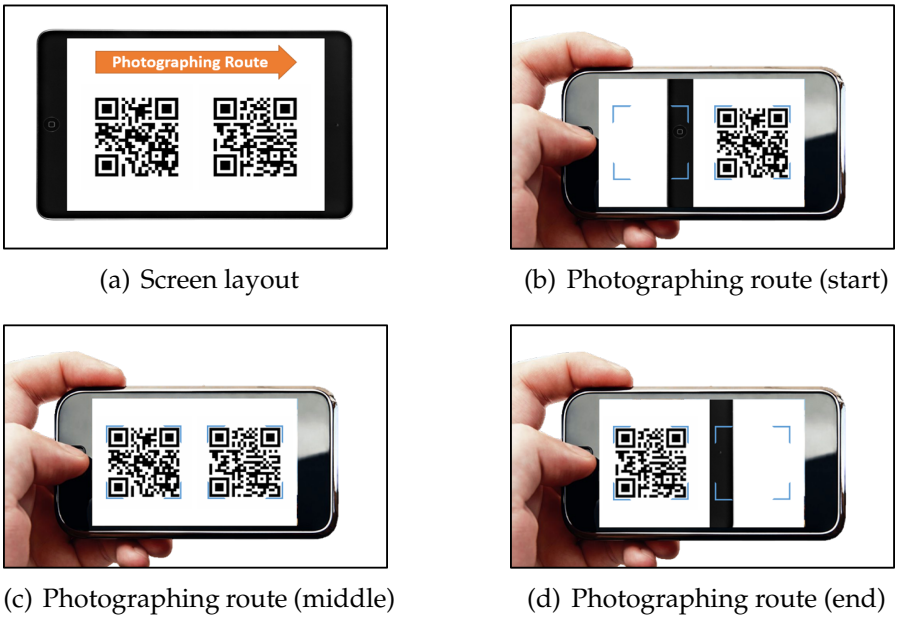
(a) Fingerprint quality (b) Noise PAE

Figure 4.9. Correlations with movement. The fingerprint quality is estimated using a reference fingerprint extracted from five burst images.

Section 4.6.4, this correlation can be used to improve the performance of the noisechain-based forgery detector.

The correlations with movement: Fig. 4.9 plots the fingerprint quality and the noise PAE calculated from each burst series. For the injection burst series and the replacement burst series, both the fingerprint-movement correlation and the noise-movement correlation are severely distorted. These distortions, like the distortion of the attenuation pattern, are primarily caused by the injection of non-uniform camera fingerprints. Due to the fact that the fingerprints extracted from different image sets are normally rather different in terms of fingerprint quality and noise PAE, it is difficult for an adversary to preserve both correlations at the same time.

Another major difference between the normal burst series and modified burst series is the distribution of their noise PAE. As shown in Fig. 4.9(b), the noise PAE of a normal image is significantly lower than the noise PAE of a modified image. This is intuitive because both the injection burst series and replacement burst series involve a fingerprint injection process. Because the camera fingerprint is also a white Gaussian noise, the fingerprint injection process actu-



(a) Screen layout (b) Photographing route (start)

(c) Photographing route (middle) (d) Photographing route (end)

Figure 4.10. Screen layout and photographing route.

ally introduces extra noise into the target image and will inevitably increase the noise PAE of the forged image.

4.5 The Proposed System

In this section, we introduce the *CIM* protocol with the proposed forgery detectors in detail. The architecture of our system is the same as the one in Fig. 4.1 while the underlying protocol is different. *CIM* allows a user to submit either single or multiple images for the purpose of authentication.

4.5.1 CIM Protocol

The proposed authentication system involves two phases: the registration phase and the authentication phase. During the registration phase, the verifier requires the user to submit a user name, the smartphone model, and a burst se-

ries captured by his/her smartphone. The reason for requiring the smartphone model is that several thresholds used by *CIM* are different for different smartphone models. After receiving these data, the verifier extracts a reference fingerprint ($\hat{\mathbf{K}}$) from the received burst images and constructs a profile <user name, reference fingerprint, smartphone model> for this user.

In the authentication phase, the user sends out a transaction along with the user name to the verifier. The verifier then authenticate the user through following procedures:

Step 1: the verifier generates two fresh and different QR codes ($\mathbf{I}^1_{QR}, \mathbf{I}^2_{QR}$) and displays them on its interface simultaneously. Each of the QR codes is encoded with the metadata of the ongoing transaction, a time stamp and a random string.

Step 2: the user captures the QR codes following the specified route shown in Fig. 4.10(a). The user first points the smartphone camera to the first QR code and initiates the photographing process in burst mode. While photographing, the user moves the smartphone along the route and stops at the second QR code as shown in Fig. 4.10(d). The measurements of the accelerometer are recorded during the photographing process. The user then uploads single or multiple captured images ($\{\mathbf{I}_1, ..., \mathbf{I}_n\}$) along with the corresponding accelerometer readings (Acc) to the verifier through a wireless channel.

Step 3: upon receiving the burst series uploaded by the user, the verifier conducts liveness detection, fingerprint matching, and forgery detection to verify the user's smartphone.

Liveness detection: The verifier detects the required QR codes in the received burst series. The authentication request will be rejected unless all the images in the burst series carry at least one of the required QR codes.

Fingerprint matching: The verifier first extracts the noise residue of each re-

ceived image using a denoising filter. It then calculates the PCE values between the reference fingerprint of the legitimate smartphone and each of the noise residues. The obtained PCE values are stored in a vector V_F. Finally, it compares the minimum value of V_F with a predefined threshold in order to detect if all the images uploaded by the user carry the fingerprint of the legitimate device.

Forgery detection: This process differs depending on the number of images submitted by the user. If the user submits **multiple images** for authentication, the verifier detects fingerprint forgery attacks using both the *noisechain-based forgery detector* and *movement-based forgery detector* introduced below. In the case that the user submits a **single image** for authentication, *CIM* requires the user to submit the image covering both QR codes (Fig. 4.10(c)). This image is normally the one captured at the highest movement speed, i.e., the image carrying the *maximum fingerprint quality* and *minimum noise PAE* of the captured burst series. Upon receiving this particular image, the verifier detects fingerprint forgery attacks through checking the *noise PAE* of the received image. Because the minimum noise PAE is the most powerful feature in detecting fingerprint forgery attacks (will be shown in Section 4.6), *CIM* can still provide reliable detection results.

In practice, the selection of the number of images to be submitted by the user depends on the specific application scenario and the verifier's security requirements. In the single-image authentication mode, both the transmission overhead and the computational overhead are minimized. The verifier checks a single feature for the detection of fingerprint forgery attacks. So far, we have yet to find any practical attacks against this detection mechanism. In the multiple-images authentication mode, the transmission overhead and the computational

overhead increase. However, the verifier is able to take advantage of the two forgery detectors, which make the authentication system highly secure.

4.5.2 Noisechain-based Forgery Detector

In order to detect the integrity of the noisechain, we construct two correlation matrices for the received burst series $\{\mathbf{I}_1, ..., \mathbf{I}_n\}$. The left correlation matrix $\mathbf{C}_L$ is an $(n-5) \times 5$ matrix. Its entry $\mathbf{C}_L[i,j]$ is the PCE value between the noise residues of $\mathbf{I}_i$ and $\mathbf{I}_{i-j}$. The right correlation matrix $\mathbf{C}_R$ is an $(n-5) \times 5$ matrix. Its entry $\mathbf{C}_R[i,j]$ is the PCE value between the noise residues of $\mathbf{I}_i$ and $\mathbf{I}_{i+j}$. After obtaining these correlation matrices, the verifier checks three features for the detection of fingerprint forgery attacks: the mean, the standard deviation, and the attenuation pattern.

The reason for choosing the mean and standard deviation is because all forged burst series contain extra noise components. As discussed in Section 4.4.4, a burst series fabricated through the quick injection strategy carries the fingerprint of the adversarial device, and a burst series fabricated through the fingerprint replacement strategy carries the negation of the short-term noise. Because of these extra noises, the correlation matrices of a forged burst series always has a significantly higher mean and standard deviation than those of a legitimate burst series. Here we use the mean and standard deviation of the concatenated matrices ($\mathbf{C}_L$ and $\mathbf{C}_R$). The authentication request is rejected if any of the two values is larger than a predefined threshold.

If both the mean and the standard deviation are within the normal range, the detector will further check the attenuation pattern of the correlation matrices. In the ideal case, the value of $\mathbf{C}_L[i,j]$ and $\mathbf{C}_R[i,j]$ should both gradually decrease

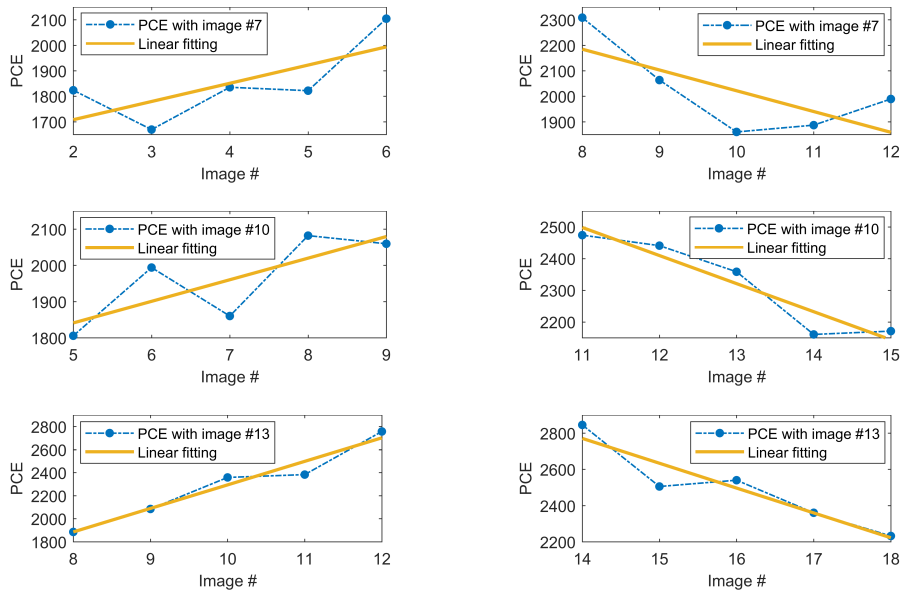


Figure 4.11. The correlation between a burst mode image and nearby images. The x-axis is the sequential number of nearby images. For instance, in the first figure, the PCE at the image #2 refers to the PCE value between image #7 and image #2.

as j increases. In practice, few matrices can strictly follow such pattern. For instance, the heat map shown in Fig. 4.8 is apparently more "noisy" than the ideal case. This is because the strength of the camera fingerprint and the short-term noise usually vary from one image to another. This inevitably introduces uncertainty into the obtained correlation matrices and makes it difficult to conduct pattern matching.

In order to overcome the above issue, we design a pattern matching algorithm named *Slope Counting*. The key observation behind slope counting is, although most correlation matrices are unlikely to strictly follow the attenuation pattern, their attenuation trends are surprisingly robust. To illustrate, we pick three images from a burst series and plot their attenuation trends using linear fitting. As shown in Fig. 4.11, their correlations to nearby images always tend to increase on the left side and to decrease on the right side. Leveraging this

observation, *Slope Counting* calculates the percentage of fitting lines that match the desired trends and use thresholding to determine if the noisechain has been distorted.

Specifically, the *Slope Counting* algorithm works as follows. Given a burst series $\{\mathbf{I}_1, ..., \mathbf{I}_n\}$ submitted by the query smartphone, *Slope Counting* first generates the left correlation matrix $\mathbf{C}_L$ and the right correlation matrix $\mathbf{C}_R$. Then, for both $\mathbf{C}_L$ and $\mathbf{C}_R$, the algorithm generates linear fitting lines for each row and record their slopes. The slopes obtained from the tth row of the left correlation matrix and of the right correlation matrix are respectively represented as m_t^L and m_t^R. The algorithm then compares the obtained slopes with predefined thresholds (ω^L and ω^R) in order to determine if the fitting lines match the desired trends. Finally, *Slope Counting* counts the number of matching slopes and outputs a matching ratio η that represents the percentage of matching slopes. The details of *Slope Counting* are shown in Algorithm 4.

Algorithm 4 Slope Counting

F1 **function** Slope_Counting ($\{\mathbf{I}_1, ..., \mathbf{I}_n\}, i, j$)
1. $(\mathbf{C}_L, \mathbf{C}_R) \leftarrow CorrelationMatrix(\{\mathbf{I}_1, ..., \mathbf{I}_n\})$
2. **for** t:= 1 **to** n-5 **do**
3. $\left(m_t^L, m_t^R\right) \leftarrow LinearFit\left(\mathbf{C}_L\left[t, :\right], \mathbf{C}_R\left[t, :\right]\right)$
4. **If** ($m_t^L > \omega^L$) **then**
5. $Count \leftarrow Count + 1$
6. **end if**
7. **If** ($m_t^R < \omega^R$) **then**
8. $Count \leftarrow Count + 1$
9. **end if**
10. **end for**
11. **Return** $\frac{Count}{2(n-5)}$

end function

4.5.3 Movement-based Forgery Detector

The movement-based forgery detector builds upon the observation that introducing foreign fingerprints into a burst series will inevitably increase their noise PAE and will distort their correlation with the movement pattern. In particular, the detector uses three features to differentiate forged burst series from legitimate ones: *the minimum noise PAE*, *the maximum noise PAE*, and *the correlation with movement*.

During the authentication process, the detector first constructs a noise vector V_N, which stores the noise PAE of each query image. For image $\mathbf{I}_i$, its noise PAE $V_N[i]$ is calculated using $PCE(\mathbf{W}_i, \mathbf{W}_i)$, where $\mathbf{W}_i$ is the noise residue of $\mathbf{I}_i$. The detector then finds the maximum (V_N^{max}) and the minimum (V_N^{min}) of V_N and compares them with predefined thresholds (ω_N^{max} and ω_N^{min}). Because all received burst series are captured following the movement pattern shown in Fig. 4.10, the maximum of V_N is always obtained at an image captured at the stationary stage and the minimum of V_N is always obtained at an image captured at the high movement speed. This makes the range of the noise PAE remarkably robust for legitimate burst series. Therefore, we choose a ω_N^{min} that is slightly higher than the lower limit of the normal range and directly use the upper limit as ω_N^{max}. The authentication request will be rejected if either V_N^{max} or V_N^{min} is higher than the corresponding threshold.

If the noise PAE of the target burst series are all within the normal range, the detector further checks the fingerprint-movement correlation and the noise-movement correlation. Here we use the linear correlation coefficient as our similarity metrics. It outputs a value ranging from -1 to 1, where 1 represents total positive linear correlation, 0 represents no linear correlation, and -1 represents

total negative linear correlation. The verification process is as follows: 1) Calibrate the accelerometer readings Acc through eliminating gain and offset errors unique to individual smartphones (Das, Borisov and Caesar, 2016). 2) Constructs a vector V_V that stores the instantaneous velocity corresponding to each image in the burst series. Since the photographing process starts at a static stage, this vector V_V can be easily calculated using the calibrated accelerometer readings. 3) Calculates $correlation\,(V_N, V_V)$ and $correlation\,(V_F, V_V)$ and compares their absolute values with a predefined threshold ω_C. The burst series will be identified as forged if either of them is lower than ω_C.

4.6 Attack Detection

In this section, we first introduce our experimental methodology. We then discuss the security of the proposed protocol through examining its resistance against the replay attack, the fingerprint forgery attacks described in Section 4.3, and advanced adversaries who know the detailed setting of our detection mechanism.

4.6.1 Experimental Methodology

Camera Fingerprinting: The analysis of camera fingerprints is conducted using Matlab on a Windows system. All of the images evaluated in this section are captured in the indoor environment because an authentication is seldom carried out in an outdoor environment. For fingerprint extraction and matching, we use the code published by the digital data embedding laboratory (Goljan, Fridrich and Filler, 2009). Specifically, we use the wavelet-based denoising filter

described in (Fridrich, 2009*b*) to extract the noise residue of an image. The noise residue is then processed by a zero-mean filter to remove linear pattern and a Wiener Filter in Fourier domain to remove periodical patterns. These procedures reduce the impact of the artifacts common to same-model cameras and the artifacts caused by JPEG compression. For fingerprint forgery attacks, all forged images in this paper are saved in the JPEG format with a quality factor of 95. This quality factor controls the compression ratio of the JPEG compression process. In matlab, it is a scalar in the range 0 to 100, where 0 indicates the highest compression and 100 indicates highest quality. Intuitively, the adversary may want to use a quality factor of 100 to preserve the quality of the injected fingerprint. However, according to our experimental results, both kinds of fingerprint forgery attack can significantly increase the file size of the forged image. A forged image stored with a quality factor of 100 is normally at least twice the file size of a legitimate image, and thus can easily be detected. Therefore, the quality factor is set to 95 in our experiment. Under this setting, the forged images have a similar file size as a legitimate one and will preserve most of the injected camera fingerprint.

Devices: We employ 22 smartphones of 5 different models for evaluation: i) 10 iPhone 6; ii) 3 Samsung Galaxy S8; iii) 3 LG G5; iv) 3 Samsung J3; v) 3 Moto G4. The technical specifications of the smartphones are shown in Table 4.1.

Metrics: False Acceptance Rate (FAR) and *True Acceptance Rate* (*TAR*) measure the likelihood that a illegitimate/legitimate burst series is wrongly/correctly accepted by the verifier. FAR and TAR vary depending on the chosen thresholds. In practice, the thresholds are often determined by setting an upper bound for the FAR, which is set to 0% in this paper. *Peak to Correlation Energy (PCE)* measures the similarity degree between two noise residues. In this paper, it is used

not only to measure the correlation between a query image's noise residue and the reference fingerprint, but also to indicate the fingerprint quality of a burst series. *Box plot* is a graphical plot that displays the distribution of the obtained PCE values based on five values: minimum value, first quartile, median, third quartile, and maximum. It is used to compare the PCE distribution of different experimental settings.

4.6.2 Replay Attacks

Under replay attacks, our system can achieve 100% detection rate. This is intuitive because the verifier generates fresh QR codes for each authentication request and never reuse them. Because even the lowest version of QR code can support up to 5.7×10^{45} different images (QR code Model 2, Version 1, ECC L (*Information capacity and versions of the QR Code*, N.d.)), it is hardly possible for the attacker to bypass the liveness detection mechanism within limited trials.

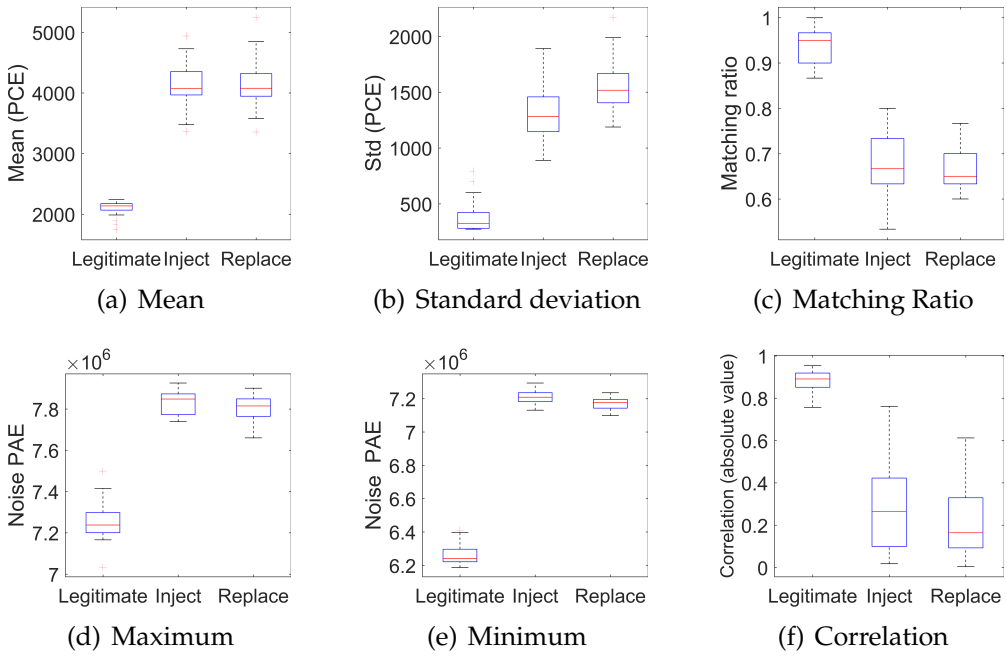
4.6.3 Fingerprint Forgery attacks

The fingerprint forgery attack is the most challenging attack which enables the adversary to fabricate arbitrary images carrying the victim's camera fingerprint. As introduced in section 4.3.4, there are two strategies to fabricate forged images: 1) *Quick Injection*, the adversary directly injects a victim's camera fingerprint into an image captured by an adversarial device. 2) *Fingerprint Replacement*, the adversary removes the inherent fingerprint of the target image before injecting the victim's camera fingerprint.

To evaluate the performance of the proposed detectors in detecting these forgery attacks, we construct three kinds of burst series for comparison: 1) *Le-*

gitimate burst series: a series of unmodified burst images. This is the burst series submitted by legitimate users. 2) *Injection burst series*: a forged burst series fabricated through the quick injection strategy. Specifically, to fabricate a forged image, we first extract a reference fingerprint $\hat{\mathbf{K}}_V$ through averaging the noise residues of five burst images captured by the victim smartphone. We then embed $\hat{\mathbf{K}}_V$ into the target image $\mathbf{J}$ using $\mathbf{J}' = \left(1 + \alpha\hat{\mathbf{K}}_V\right)\mathbf{J}$. The reference fingerprints injected into different images are always extracted from different and non-overlapping image sets. 3) *Replacement burst series*: a forged burst series fabricated through the fingerprint replacement strategy. In this attack, we first extract the noise residue $\mathbf{W}$ of the target image $\mathbf{J}$ using the denoising filter implemented in (Goljan, Fridrich and Filler, 2009). We then remove $\mathbf{W}$ from $\mathbf{J}$ using $\mathbf{J}'^0 = (1 - \beta\mathbf{W}_A)\,\mathbf{J}$. Finally, we embed a victim fingerprint into the sanitized image $\mathbf{J}'^0$ using the same approach as the quick injection attack. The strength factors (α and β) for each smartphone model are empirically derived. The length of the burst series is set to be 20 images since the attenuation pattern and the correlation coefficient requires long burst series to achieve robust detection rate. The impact of the burst series length will be evaluated in section 4.6.4.

We construct 60 burst series of each type for the smartphones listed in Table 4.1. For each smartphone model, we calculate the proposed features from each burst-series and examine the distributions of the obtained values. The obtained distributions are very similar across all tested smartphones, except for the range of PCE values. To illustrate the performance of each feature, Fig. 4.12 shows a plot of the distributions obtained from iPhone 6. It can be observed that, for most features, there is no overlap between the distributions of legitimate burst series and that of forged burst series. The mean of the correlation matrix and the minimum noise PAE are particularly effective. For the correlation coefficient,



(a) Mean (b) Standard deviation (c) Matching Ratio

(d) Maximum (e) Minimum (f) Correlation

Figure 4.12. The distributions of each feature

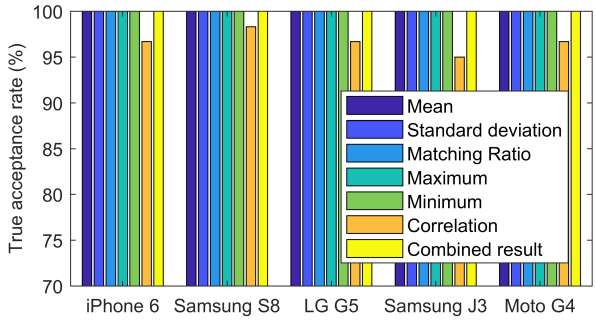


Figure 4.13. Detection of forgery attacks: the TAR of each feature at a FAR of 0%. The combined result is obtained through a bagged decision tree.

there is a small overlap between the distributions of different burst series. This is because the accelerometers on smartphones are very noisy. The performance of this feature could be improved if advanced denoising methods are adopted. Several possible calibration approaches are discussed in Section 4.8. Fig. 4.13 summarizes the TAR of each feature at a FAR of 0%. The proposed forgery detectors achieve 100% TAR in detecting forgery attacks.

4.6.4 Impact of Burst Series Length

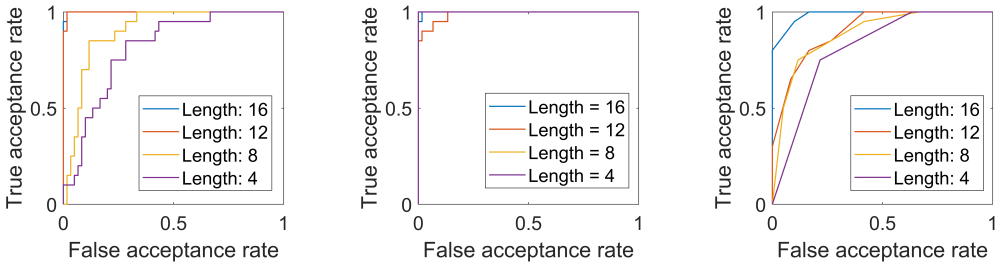
We now investigate the impact of the burst series length on the proposed system. Here we consider the scenario where the verifier still follows the challenge-response process in Section 4.5.1, except that it only requires the user to submit part of the captured burst series for authentication.

We conduct the evaluation with burst series of five different lengths: 16, 12, 8, 4, and 1. To construct a burst series of length n, we first capture a burst series of length 20 following the photographing route in Fig. 4.10(a). We then select n consecutive images from the captured burst series and use them to construct the shorter burst series. In particular, if n is greater than 10, we select the first n images from the captured burst series. The constructed burst series contains images captured at the static stage as well as at the moving stage. If n is less than 10, we select the n images around the middle of the burst series. This ensures that the constructed burst series will always contain images captured at the high movement speed. For each burst series length, we construct 60 legitimate burst series, 60 injection burst series and 60 replacement burst series and evaluate the proposed forgery detectors. Table 4.2 shows the TAR of each feature at 0% FAR.

For the movement-based forgery detector, the maximum and the minimum of the noise PAE can always achieve 100% TAR at 0% FAR. This is because all tested burst series contain images captured at the high movement speed. Due to the noise-movement correlation, the minimum noise PAE is fairly stable across burst series of different lengths. For the burst series of length 1, the minimum noise PAE is just the noise PAE of the single image. For the correlation coefficient, the TAR drops rapidly when the burst series length drops below 10. This is because, under the short and simple movement pattern, the correlation coef-

Table 4.2. The TAR of each feature at a FAR of 0%

Length	16	12	8	4	1
Correlation	95%	90%	0%	10%	n/a
Maximum	100%	100%	100%	100%	n/a
Minimum	100%	100%	100%	100%	100%
Mean	100%	100%	100%	100%	n/a
Std	95%	85%	100%	100%	n/a
Matching ratio	80%	30%	0%	0%	n/a
Overall result	100%	100%	100%	100%	100%



(a) Correlation (b) Standard deviation (c) Attenuation pattern

Figure 4.14. ROC curves. When calculating the matching ratio, if the length is set to 8 and 4, each slop is calculated from 3 and 2 neighbors respectively.

ficient calculated from forged burst series can also be very high. However, if the upper bound of the FAR is raised, this feature can also provide the verifier with valuable information (as shown in Fig 4.14(a)).

For the noisechain-based forgery detector, the mean value of the correlation matrices can always achieve 100% TAR at 0% FAR. For the standard deviation, the TAR approaches 100% when the burst series length drops below 10. This is because the burst series with a length of less than 10 images will only contain images captured in the moving stage. Due to the noisechain-movement correlation (introduced in Section 4.4), a legitimate burst series containing only moving stage images will always have a fairly low standard deviation, which makes it easy to differentiate forged burst series and legitimate ones. For the attenuation pattern, the TAR drops quickly with the decreasing of the burst series length.

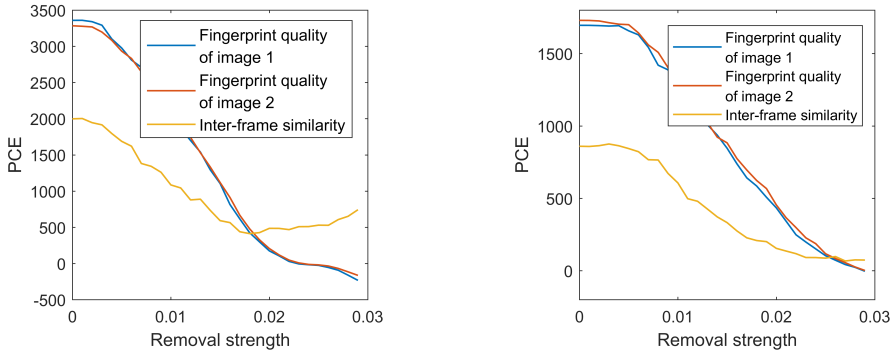
But once again, through raising the upper bound of the FAR, the attenuation pattern can also provide valuable information (as shown in Fig. 4.14(c)).

4.6.5 Detection of Advanced Adversary

We now discuss advanced adversaries who are capable of collecting large number of images captured by the victim smartphone and knowing the detailed setting of our defending mechanism. In order to bypass the forgery detectors of *CIM*, the burst series fabricated by the adversary should meet following requirements:

Proper inter-frame similarity: the similarity value between the noise residues of continuously captured images should lie in the normal range. Conventionally, this is achieved through removing the inherent fingerprint of the target images. Since normal images share only the camera fingerprint of the photographing device, the fingerprint removal process can easily reduce their correlation to a negligible level and thereby balance out the similarity gain caused by the subsequent fingerprint injection process. In our setting, due to the use of burst images, the fingerprint removal process will also affect the short-term noise shared between adjacent images. The adversary needs to find a removing approach that can balance out the similarity gain caused by the fingerprint injection process and can retain the attenuation pattern of the noisechain. Specifically, *the adversary needs to remove the adversarial device's camera fingerprint without distorting the relative strength of the short-term noises*.

We first evaluate the classical fingerprint removal approach. As shown in Fig. 4.15(a), due to the existence of the short-term noise, the classical fingerprint removal method can no longer reduce the inter-frame similarity to a negligible



(a) One round removal (b) Two round removal

Figure 4.15. The impact of fingerprint removal. The fingerprint quality is estimated using a reference fingerprint extracted from five burst images. In the two round removal, the removal factor for the first round is set to be 0.12 in order to minimize the noise PAE of the sanitized images.

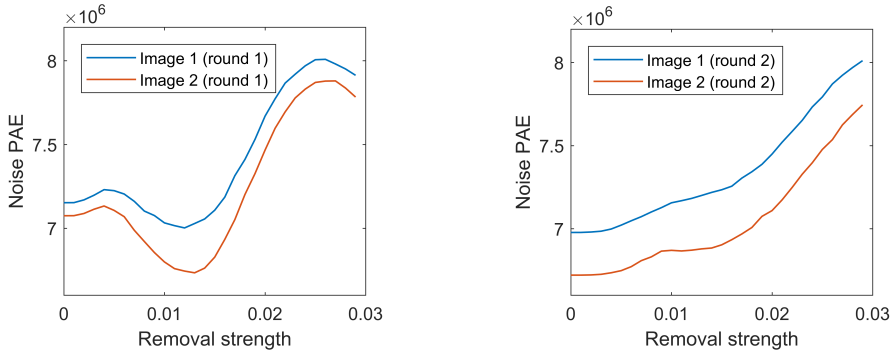
level. The sanitized images will still have considerable similarity even when the adversarial device's fingerprint is completely removed ($\beta = 0.02$). This is because a new common component is introduced into the sanitized images due to over-removing, which refers to the situation that the applied removal factor is higher than the strength of the short-term noise carried in the target images. Denote the target image's short-term noise as $\mathbf{W}_s$ and its strength as β_s. Conducting fingerprint removal with a β of 0.02 is similar to introducing a new noise component $(-\mathbf{W}_s)$ with a strength of $(0.02 - \beta_s)$.

Since the ineffectiveness of the classical fingerprint removal approach is caused by over-removing the short-term noise, we further test an iterative fingerprint removal approach. In this approach, the target images go through multiple rounds of classical fingerprint removal with small strength factors. Fig. 4.15(b) plots the results obtained from a two-round removal. It can be observed that the iterative removal approach can reduce the inter-frame similarity to a very low degree, making it possible for the adversary to balance out the similarity gain caused by fingerprint injection. However, the iterative removal ap-

proach will weaken and even distort the attenuation pattern of the noisechain, which makes it harder for the adversary to preserve the attenuation pattern in the forged burst series. Moreover, as will be shown later, the iterative fingerprint removal approach will also make the noise PAE of the target images far exceeds the normal range.

Proper noise PAE: the noise PAE of all forged images should lie in the normal range. To meet this requirement, the adversary needs to find the proper removal strength that could balance out the extra noise introduced by the subsequent fingerprint injection process. Fig. 4.16 shows the relationship between the removal strength and the noise PAE. We make following observations: 1) The fingerprint removal process can reduce the noise PAE of the target image, but only when the removal strength is sufficiently small (lower than 0.012). 2) The removal strength that achieves the lowest noise PAE can not reduce the inter-frame similarity to the required degree. 3) Regardless of the removal method or the removal factor, the reduction of the noise PAE can hardly balance out the extra noise introduced by the subsequent fingerprint injection process. So far, the best strategy to reduce the PAE gain caused by fingerprint injection is to use high victim quality fingerprints and low injection factors.

Attenuation pattern: the forged burst series should have the attenuation pattern introduced in Section 4.4.4. Generally, there are two ways to implement this requirement: 1) Preserve the attenuation pattern of the captured burst series: due to the fact that the attenuation pattern is mainly distorted by the non-uniform fingerprints injected into the burst series, the adversary could try to preserve the attenuation pattern by carefully selecting the injection strength for each image of the burst series. However, it could take a large number of iterations for the adversary to find the proper factors, which makes this strategy



(a) The first round of removal (b) The second round of removal

Figure 4.16. The impact of iterative removal on the noise PAE of two images. In (b), the removal strength for round 1 is set to 0.012.

extremely time-consuming. 2) Falsify a fake attenuation pattern through injecting a forged noisechain. The injected noisechain should be strong enough to overwhelm the non-uniformity introduced by the fingerprint injection process. Although this strategy is more efficient than the previous one, it will inevitably increase the inter-frame similarity and the noise PAE of the fabricated burst series.

Correlations with movement: the forged burst series should preserve the fingerprint-movement correlation and the noise-movement correlation. Implementing this requirement also requires the adversary to carefully select the injection strength for each image of the captured burst series. It is particularly difficult and time-consuming to find the proper injection factors that can preserve the attenuation pattern, the fingerprint-movement correlation, and the noise-movement correlation at the same time.

In conclusion, the best strategy for fingerprint injection is to embed the captured images with high quality victim fingerprints using carefully selected injection factors. The best strategy for fingerprint removal is to conduct multiple rounds of denoising using low removal factors. To understand the performance

of these strategies, we conducted a quick injection attack and a fingerprint replacement attack using 600 victim images captured by a Samsung S8. During the photographing of these images, we target the smartphone camera at the same background with minor angle change. In this experiment, we first estimate 20 victim fingerprints from these images (each from 30 images). We then construct a 20-image burst series using another Samsung S8 and conduct the quick injection attack and the fingerprint replacement attack. We use carefully selected injection factors for fingerprint injection and employ three-round fingerprint removal.

Table 4.3 lists the value of each feature before and after each kind of attack. For the fingerprint injection attack, due to the application of high quality fingerprints and low injection factor, the PAE gain caused by fingerprint injection is relatively low, though still detectable. The correlation with movement and the attenuation pattern were partially preserved due to the carefully selected injection factors. However, because of the existence of the adversarial device's camera fingerprint, the mean of the injection burst series is significantly higher than that in the legitimate burst-series. For the fingerprint replacement attack, the mean is significantly decreased due to the removal of the adversarial device's fingerprint. However, the fingerprint removal process increased the Noise PAE of the forged images and further distorted the attenuation pattern as well as the correlation with movement. Therefore, it is difficult for the adversary to generate forged burst series satisfying all above requirements. The noisechain-based forgery detector and the movement-based forgery detector significantly raise the bar for fingerprint forgery attacks.

Table 4.3. Advanced adversary with 600 images

Feature	Legitimate	Injection	Replacement
Correlation	0.81	0.68	0.61
Maximum	3.5×10^6	3.6×10^6	3.8×10^6
Minimum	2.8×10^6	3.0×10^6	3.2×10^6
Mean	1383	4034	2405
Std	992	1232	1627
Matching ratio	73.3%	63.3%	56.7%

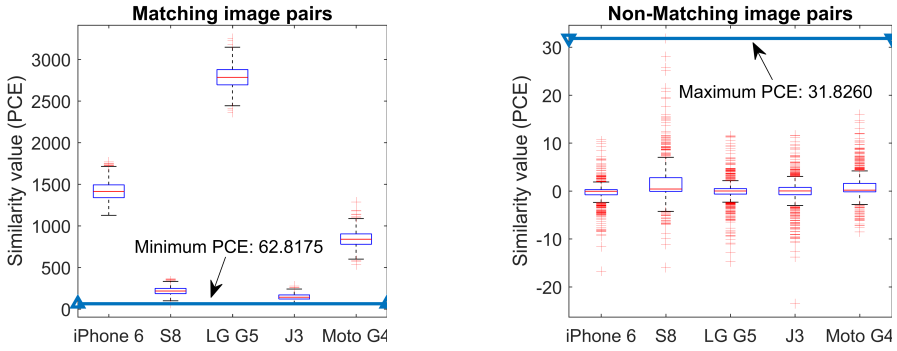


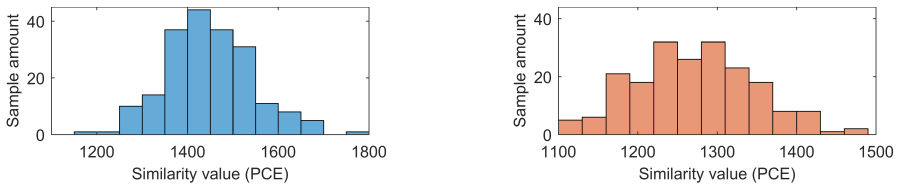
Figure 4.17. PCE distribution of burst images. For matching image pairs, the PCE value is mainly determined by the strength of the camera fingerprint.

4.7 Performance Evaluation

In this section, We present results validating the usability of our camera-based smartphone authentication system.

4.7.1 Smartphone Identification via Burst Images

We first demonstrate the feasibility of using burst images in camera identification. For each smartphone model, we construct 800 matching image pairs and 800 non-matching image pairs. Recall that a matching image pair contains two burst images captured by the same smartphone. The two images of an image pair are always selected from different burst series in order to eliminate the interference from noisechains. A non-matching image pair contains two burst



(a) QR set (b) NonQR set

Figure 4.18. The impact of QR code on fingerprint quality

images captured by different smartphones of the same model. Each of the test image is randomly selected from a burst series.

Fig. 4.17 shows the PCE distributions obtained from each smartphone model. It can be observed that, the PCE values obtained from matching image pairs are significantly higher than the PCE values obtained from non-matching image pairs. This indicates that the fingerprints carried in burst images are strong enough to differentiate off-the-shelf smartphones (with 100% TAR at 0% FAR). Another worthy observation is that, for iPhone 6, the PCE values obtained from burst images are ten times higher than the PCE values obtained from normal images that are not captured in burst mode (Fig. 4.2). This is likely because shooting in burst mode will lead to considerable computational cost. iPhone 6 may disable some post-processing in order to allow a faster burst rate. Most post-processing operations suppress the camera fingerprint of the target image. The take away message is that burst images are at least as good as normal images in differentiating smartphones.

4.7.2 The Impact of QR code on Smartphone Identification

Because the texture of an image can significantly affect the quality of the fingerprint extracted from that image, we now evaluate the impact of QR code on the camera fingerprint. We construct two image sets for comparison. 1) *QR set*: 200

burst images constructed through photographing two QR codes shown on an iPad Pro. We use the version 2 QR codes with data correction level M. The first QR code contains 35 alphanumeric characters and the second QR code contains 60 numeric values. 2) *NonQR set*: 200 burst images captured in the same room without the presence of the verifier's interface (i.e., the iPad Pro). We calculate the PCE between the images in each set with a reference fingerprint extracted from a burst image. The reference image is captured without the presence of the QR codes. Fig. 4.18 plots the distribution of the PCE values obtained from each set. The results are counter-intuitive because the PCE value obtained from the QR set is even higher than that from the Non-QR set. We found that this is because the intensity of the ambient light is increased due to the presence of the iPad Pro. The take away message is that it is feasible to use QR code images for camera identification.

4.7.3 Time Overhead

The overhead of a camera-based smartphone authentication system comes from two aspects: transmitting the images to the verifier and verifying the camera fingerprint. With LTE, the time overhead of the transmission stage is acceptable in most application scenarios. For instance, four images captured by Samsung J3 are around 1.5 MB in total. Using T-Mobile LTE (16 Mbps (Cassavoy, 2018)), the user is able to upload the images to the verifier within one second. Fig. 4.19 shows the computation overhead of *CIM* under different settings. The verification is conducted on a laptop with an 8-core CPU running at 2.8 GHz. Each overhead value is an average of computation time for 20 times of authentication. It can be observed that the computation overhead of *CIM* is relatively large and

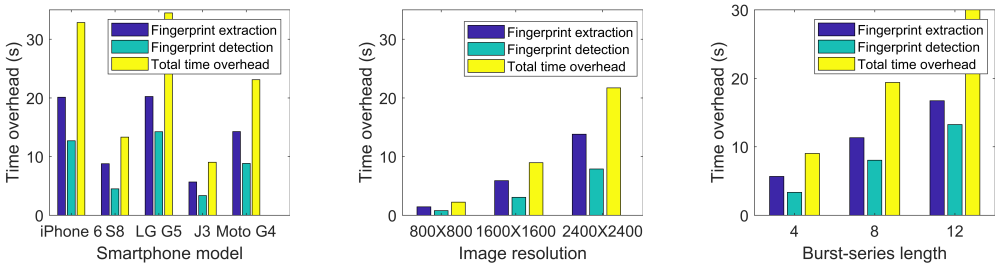


Figure 4.19. (a) The time overhead of different smartphones. (b) The impact of image resolution. (c) The impact of burst series length. The verifier uses the parallel pool of Matlab with four workers. For (a) and (b), the burst series length is four. For (c), the images are captured by a Samsung J3.

Table 4.4. Usability Study

	Fingerprint Matching	Liveness Detection	Forgery Detection
TAR	100 %	96.7 %	100 %

increases quickly with the burst series length and the image resolution. The reason is twofold. First, although different images are processed in parallel, there are only four workers in the parallel pool of Matlab. Second, the code for fingerprint extraction and matching has not been optimized for efficiency. Utilizing GPU computing and parallel computing, the efficiency of the system can be greatly improved. Another promising direction to reduce the computation overhead is to use downscaled images. On the one hand, because of the strong fingerprint of smartphone cameras, downscaled images can also provide reliable identification results. On the other hand, because the downscaling process has an uniform impact (proportional to the scaling ratio) on the noise components of all target images (Ba et al., 2018), the downscaled images will preserve the noisechain and the correlations with movement.

4.7.4 Usability Study

To evaluate the usability of the proposed system, we set up an iPad Pro as the verifier and involve ten male participants and ten female participants to use the proposed scheme with their preferred angle and speed normally. According to (Nielsen, 2012), testing 20 users is sufficient to get statistically significant results. During each test, we first provide an introduction to the system and conduct an authentication for the demonstration purpose. Each participant is then required to carry out three rounds of authentication using an iPhone 6. We then upload the captured images to a laptop and conduct liveness detection (QR code matching), fingerprint matching and forgery detection. Table 4.4 lists the TAR of each detection process. For fingerprint matching and forgery detection, our system employed 20 images for each authentication attempt and achieved 100% TAR. For liveness detection, the system failed to detect the QR codes in several images of two burst captured by a user. However, this is because one of the position markers in those images is blocked by the reflection of a lamp. After adjusting the angle of the iPad Pro, the images captured by that user also succeeds in passing the liveness detection process. Therefore, the individual differences among users do not affect the accuracy of the system.

4.8 Points of Discussion

Accelerometer calibration. Accelerometers on smartphones are known to be noisy and unreliable. In *CIM*, the accelerometer readings are calibrated through removing the gain and offset errors incurred by manufacturing imperfection (Das, Borisov and Caesar, 2016). Because these two errors are unique to the

user's smartphone, the accelerometer readings submitted by an adversary may not be calibrated correctly, which further reduces the chance for the adversary to bypass the forgery detector. In practice, there are many other calibration approaches that can be used to improve the accelerometer readings. For instance, verifiers can remove the noise from accelerometer readings using various denoising filters. They can also crosscheck the accelerometer readings with the measurements of the smartphone's built-in gyroscope. Another promising direction is to reuse the built-in speaker in the verifier's interface as a Doppler radar (Zhang, Tan and Yang, 2017). The verifier emits acoustic signals and collects reflections using its microphones. It then uses the Doppler shifts of the collected reflections to obtain the movement speed.

Man in the Middle Attacks. In practice, the adversary may try to position himself/herself between the user and verifier, and performs various MITM attacks. For instance, an adversary may intercept the user's request (transaction) and sends out a malicious one to the verifier. To address this attack, the verifier could embed an abstract of the user's request into the QR codes shown on its interface. In this way, the user is able to verify the information of the transaction before taking pictures. The user can terminate the authentication process if the information is different from what he/she requested.

Selection of thresholds. *CIM* uses one threshold for fingerprint matching and six thresholds for forgery detection. For fingerprint matching, because images captured in burst mode have high quality fingerprint, we use the same threshold for all smartphone models. For forgery detection, four of the applied features are influenced by smartphone models: mean, standard deviation, maximum PAE, and minimum PAE. However, because of the large gap between legitimate burst series and forged burst series, the thresholds for these features

can be set during the registration process in three steps: 1) Extract camera fingerprints from the images uploaded by the user and use them to construct several forged burst series. 2) Extract the four features from each forged burst series and find the the minimum of each feature. 3) For each feature, set the threshold to the average of the obtained minimum and the feature extracted from the legitimate burst series. These thresholds can be further optimized when more images are collected from the same smartphone model.

4.9 Conclusion

In this paper, we first present two novel observations of smartphones taking pictures: 1) There exists a noisechain embedded in continuously captured burst images. 2) The camera fingerprint and noise components of an image are correlated with the movement of the photographing device. We explore these two observations, design two reliable forgery detectors for the detection of forgery attacks against PRNU-based camera fingerprinting, and propose *CIM*, a camera-based smartphone authentication system. *CIM* is practical since it leverages universal sensors (camera and accelerometer) in a smartphone and a user just needs to take pictures in burst mode while moving the smartphone along a simple route. Extensive experiments are conducted to validate the effectiveness of *CIM* against fingerprint forgery attacks. A user can submit either multiple or one image for authentication. In both cases, *CIM* achieves 100% TAR at 0% FAR in both fingerprint matching and forgery detection.

Chapter 5

Preventing Camera Fingerprint Leakage via Obfuscation-based Fingerprint Concealment

5.1 Introduction

With the prevalence of high-quality smartphone cameras and the fast growing of social networks, a large number of digital images are being captured and shared on social platforms exponentially. For example, 350 Million photos are uploaded to Facebook every day (Omnicore, 2018*a*). 50 billion photos have been shared on Instagram until September 2018 (Omnicore, 2018*b*). Meanwhile, the increasing concerns on image privacy is along with the data explosion (De Choudhury et al., 2009; Yeung et al., 2009; Pesce et al., 2012; Zerr et al., 2012; Christin et al., 2013). In a recent survey (Handley, 2018), 40% of survey respondents said that they have deleted at least one social account in the past year due to privacy concerns. In the literature, an enormous amount of research has been

carried out in an attempt to identify and address the privacy issues incurred by the contents and tags of shared images (Yu et al., 2017; Wang et al., 2013; Nov, Naaman and Ye, 2009; Squicciarini, Xu and Zhang, 2011; Wang, Xu and Grossklags, 2011). However, the noise components of an image could also lead to security and privacy issues.

As reported in (Lukas, Fridrich and Goljan, 2006; Fridrich, 2009*a*; Khanna, Mikkilineni and Delp, 2009), all images captured by digital cameras contain a device-specific noise component originated from the Photo Response Non-Uniformity of image sensors. This noise is particularly effective for image-to-camera matching and has been recognized as the most reliable hardware fingerprint of digital cameras. Because of the stability of the camera fingerprint, an adversary with a handful of the images collected from a user's social network can easily extract the fingerprint of her photographing device.

In this paper, we highlight and evaluate two critical attacks caused by camera fingerprint leakage: 1) *Identity Linking*: the adversary tries to find the association among anonymous social accounts through matching the camera fingerprints extracted from their posted images. 2) *Identity Forgery*: the adversary extracts the victim's camera fingerprint from her social accounts and embed the obtained fingerprint into foreign images captured by other devices. The forged image can be applied to frame the innocent victim (Goljan, Fridrich and Chen, 2011) or to bypass camera-based smartphone authentication system (Ba et al., 2018). The effectiveness of these two attacks are evaluated on four social platforms. Our results show that, with around 5 images from each social account, the identity linking attack can achieve a True Positive Rate (TPR) over 90% at negligible False Positive Rate (FPR), regardless of whether the accounts are within the same social platform. For identity forgery attacks, the probability

of a forged image being identified as a legitimate one is higher than 95%.

In order to counter the above attacks, we propose CFP, an intermediary between smartphone cameras and social medias. It enables the beneficial applications of camera fingerprints while concealing the camera fingerprint of the image of interest before uploading the image to the Internet. In the literature, the concealing of the camera fingerprint can be achieved through reducing the image quality (Rosenfeld and Sencar, 2009), flat-fielding (Gloe et al., 2007; Böhme and Kirchner, 2013), seam curving(Bayram, Sencar and Memon, 2013; Dirik, Sencar and Memon, 2014), adaptive fingerprint subtraction (Li, Chang and Li, 2009), and adaptive denoising (Dirik and Karaküçük, 2014; Karaküçük and Dirik, 2015). However, these approaches either remove only a small portion of the fingerprint or are extremely time-consuming. Moreover, removing the fingerprint also disables beneficial forensic tasks such as copyright protection and integrity verification.

To prevent malicious applications of the camera fingerprint without hindering the beneficial ones, we explore obfuscation-based fingerprint concealment. The idea is to alter the composition of the image's noise components through embedding an obfuscating noise. This noise is designed to be irremovable and remains constant for all photographing devices. In identity linking attacks, this noise serves as an obfuscator that confuses the adversary's fingerprint matching mechanism. Because all the images obfuscated by our system will carry a similar obfuscating noise, the similarity value obtained from the matching algorithm is always high, whether or not the images are captured by the same device. In identity forgery attacks, the obfuscating noise serves as a reliable probe for detecting forged images. Because the adversary can only access to obfuscated images, fingerprints extracted by the adversary will always carry

a significant amount of the obfuscating noise, thereby exposing the adversary. Meaning while, because our system does not remove the camera fingerprint, the smartphone owner would still be able to prove her ownership or the integrity of an obfuscated image. The beneficial applications of camera fingerprint are preserved.

Our major contributions are summarized as follows:

1. We highlight two critical identity attacks caused by camera fingerprint leakage and demonstrate their effectiveness on current image sharing practices. The impact of a comprehensive set of image post-processing techniques are also evaluated.

2. We propose a real-time fingerprint concealment system. It explores obfuscation-based fingerprint concealment and is able to prevent malicious applications of camera fingerprints without hindering beneficial ones.

3. We conduct extensive experiments to confirm the effectiveness of the proposed CFP system. The results show that the proposed design reduces the TPR of identity linking by 86% on average and enabled a forgery detection mechanism that could achieve 100% detection rate.

The rest of this paper is organized as follows. We introduce the background knowledge in Section 5.2. Section 5.3 highlights two specific attacks and evaluates their effectiveness on current image sharing practices. Section 5.4 discusses the failure of fingerprint removal and gives the design of CFP in detail. Section 5.5 evaluates the proposed system with extensive experiments. Section 5.6 concludes this paper.

5.2 Background

5.2.1 Photo Response Non-Uniformity

Photo Response Non-Uniformity (PRNU) (Lukas, Fridrich and Goljan, 2006; Fridrich, 2009*a*; Khanna, Mikkilineni and Delp, 2009) is a hardware fingerprint of digital cameras that originates from the non-uniform light-sensitivity (Janesick et al., 2001) of millions of pixels. It works as a multiplicative factor to the actual optical view during the image acquisition process (Nakamura, 2016). Denoting the fingerprint of a camera as $\mathbf{K}$, an image captured by that camera can be represented as:

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

where $\mathbf{I}^0$ and Θ respectively represents the actual optical view and other noise components. The following describes the extraction and matching of camera fingerprints.

5.2.1.1 Fingerprint Extraction

Because the camera fingerprint behaves like a white Gaussian noise (Lukas, Fridrich and Goljan, 2006; Chen, Fridrich and Goljan, 2007), the fingerprint of an image is normally extracted using a denoising filter (Caldell et al., 2010; Lukas, Fridrich and Goljan, 2006; Chen et al., 2008). The extracted noise residue is an estimate of the camera fingerprint and can be represented as

$$\mathbf{W} = \mathbf{I}^0\mathbf{K} + \Xi,$$

where Ξ is a combination of other Gaussian noise (Chen et al., 2008). A fingerprint estimated in this way is particularly noisy because Ξ is much higher in energy than the camera fingerprint. In the case where a high quality camera fingerprint is required, we can extract the noise residues of multiple images and use maximum likelihood estimation to get a better estimate of the camera fingerprint (Cain, Hayat and Armstrong, 2001).

5.2.1.2 Fingerprint Matching

To determine if two estimated fingerprints belong to the same camera, the most common method is to calculate their Peak to Correlation Energy (PCE) (Goljan, 2008). The PCE value between two estimated fingerprints increases with the energy of their common noise components. Because the noise component Ξ is different for different images, a high PCE value normally means that the two estimated fingerprints are carrying the same $\mathbf{K}$. In practice, the verifier use pre-trained threshold to differentiate matching and non-matching fingerprints.

5.2.2 Beneficial Applications of PRNU

In the literature, the PRNU of digital cameras has been demonstrated to be effective in various digital forensics tasks. The following illustrates two primary applications of PRNU:

5.2.2.1 Copyright Protection

Leveraging PRNU's superior performance in image-to-camera matching, a user can easily demonstrate the copyright of an image through presenting the photographing device (Pathways, 2010). In particular, given an image with disputes

over ownership, the verifier can extract the camera fingerprint of each candidate camera and conduct fingerprint matching to determine the origin of the target image.

5.2.2.2 Integrity Verification

Given a digital image and the camera that captured it, a verifier can identify certain types of tampering operations through checking the integrity of the PRNU (Chen et al., 2008). In particular, the verifier first estimates a high quality reference fingerprint of the photographing device. She then divides the image of interest into multiple disjoint blocks and calculate the PCE value between each obtained block and the reference fingerprint. A low PCE indicates the lacking of PRNU in the corresponding block, which implies that the image content of that block has been modified. With this approach, the camera owner was able to demonstrate if an image containing her camera fingerprint has been maliciously tampered (e.g., copy-move (Bravo-Solorio and Nandi, 2011) and splicing (Farid, 2009)).

5.3 Security & Privacy Risks

Large number of images are posted on Internet due to the popularity of image sharing, exposing the camera fingerprints of their photographing device directly to the public. These fingerprints, however, can be used not only for benevolent purposes, but also for malicious ones. In this section, we highlight the offensive potential of PRNU, demonstrate the effectiveness of PRNU-based attacks on current practices, and present the impact of post-processing techniques.

5.3.1 Identity Linking Attacks

The first malicious application of PRNU is identity linking on social networks. Users are prone to create multiple social accounts within the same social network and across different ones. Some of the accounts created are public ones and may contain the user's identity information, while some are private ones (normally under an assumed name) reserved for specific readers. Identity linking enables the adversary to find the association among all these accounts created by the same user, which poses serious threat to user privacy.

The basic idea behind this attack is to analyze the images posted on social accounts and utilize their camera fingerprints to determine which accounts belong to the same user. Specifically, given several social accounts, the adversary first estimates a camera fingerprint for each social account using a number of images posted on it. She then calculates the PCE values between every two estimated fingerprints and compares the obtained values with a predefined threshold. If the PCE value between two fingerprints is higher than a threshold, it will be considered that the two corresponding social accounts are belong to the same user.

We now demonstrate the feasibility of this attack on four representative social networks: Facebook, Wechat, Weibo and Flickr. These platforms are chosen because of their different image processing strategies. Flickr, Facebook and Wechat convert each user image to JPEG format with low, medium, and high compression, respectively. Weibo embed each received image with a constant noise component. We employ 6 smartphones of 3 different models to represent six users, including 2 Samsung S8, 2 HUAWEI P10, and 2 HUAWEI P20 Pro. For each user, we create 10 social accounts on each platform and upload 10 images

to each account. We use the built-in camera APP to capture images for Samsung S8 and HUAWEI P10 and use a third-party APP (B612) to capture images for HUAWEI P20 Pro. Overall we have 240 accounts across four platforms.

Two kinds of identity linking are conducted on these social accounts: (i) *Intra-platform identity linking*. This attack tries to find the association among different social accounts within the same platform. We divide the 240 accounts into four groups based on platforms and conduct identity linking on each group. The threshold applied for Facebook, Flickr and Wechat are respectively 48, 19, and 14. For Weibo, because its image compression process introduced a similar noise component into all images, the PCE value between all estimated fingerprints are increased and the threshold of this platform increases with the number of images applied to estimate the fingerprint of an account. In this experiment, the threshold is 1100 when 3 images are applied and is 2330 when 5 images are applied. Table 5.1 summarizes the True Positive Rate (TPR) and False Positive Rate (FPR) for each smartphone model under different settings. (ii) *Inter-platform identity linking:* This attack tries to establish links between social accounts across different platforms. Images from different platforms are down-sampled to the same resolution in order to conduct this attack. The results of this attack are summarized in Table 5.2.

It can be observed that, for all platforms under investigation, the adversary can determine with a high degree of accuracy whether two anonymous accounts are created by the same user, regardless of whether the two accounts are within the same platform. The performance of both attacks improve with the number of images applied to estimate the fingerprint of an account. Another observation is that Wechat and Weibo requires more images to achieve a better TPR. For Wechat, this is because the images posted on this platform suffer more in-

Table 5.1. Intra-Platform Identity Linking [%]: N is the number of images applied to estimate the fingerprint of an account

Platform/N	S8		P10		P20 Pro	
	FPR	TPR	FPR	TPR	FPR	TPR
Facebook/1	0	83	4	74	0	94
Facebook/3	0	100	0	100	1	100
Flick/1	0	100	0	100	4	91
Flick/3	0	100	0	100	0	100
Weibo/3	0	100	0	77	7	100
Weibo/5	0	100	0	92	2	100
Wechat/3	2	100	1	62	4	94
Wechat/5	0	100	1	89	0	100

Table 5.2. Inter-Platform Identity Linking [%]: 3/3/3/5 means the number of images applied in Facebook, Flickr, Weibo and Wechat are respectively 3,3,3, and 5. ω is the threshold

Setting	ω	S8		P10		P20b	
		FPR	TPR	FPR	TPR	FPR	TPR
3/3/3/5	15	0	85	0	100	1	92
5/5/5/7	25	0	94	0	100	0	99

formation loss during the image compression process. It requires more images to estimate a high quality fingerprint for each account. For Weibo, it requires more images because the camera fingerprint is affected by the additional noise component introduced during the image compression process The take away message is that, although uploading an image to social networks could weaken its camera fingerprint, the remaining fingerprint in the posted image can still enable the adversary to find the association among different accounts that belong to the same user.

5.3.2 Identity Forgery Attacks

The second malicious application of PRNU is identity forgery (Goljan, Fridrich and Chen, 2011; Caldelli, Amerini and Novi, 2011; Rao et al., 2013). Given a

number of images captured by a victim smartphone, an adversary can generate an estimation ($\hat{\mathbf{K}}_v$) of the victim's camera fingerprint and embed it into an image ($\mathbf{J}$) come from another device using:

$$\mathbf{J}' = \mathbf{J}(1 + \alpha\hat{\mathbf{K}}_v), \tag{5.1}$$

where α is the strength factor, $\mathbf{J}'$ is a forged image that carries the victim's camera fingerprint. With this approach, the adversary can fabricate arbitrary images that carry the victim's camera fingerprint and fool a third party into believing that the forged images are captured by the victim. Although a number of methods have been proposed to detect forged images (Ba et al., 2018; Goljan, Fridrich and Chen, 2011; Valsesia et al., 2017; Quiring and Kirchner, 2015), most of the detection mechanisms are either impractical or have security flaws.

We now demonstrate the feasibility of fabricating forged images using victim images downloaded from social networks. To do so, we first implemented a camera-based smartphone authentication system (Ba et al., 2018), in which a verifier uses the camera fingerprint to authenticate smartphones. During the authentication process, the verifier first challenges the user to provide a freshly captured image. She then estimates a camera fingerprint from the received image and calculates the PCE value between the estimated fingerprint and the reference fingerprint of the legitimate device. The authentication is successful if the PCE value is higher than a threshold. In this implementation, we use ten images to extract the reference fingerprint of a legitimate device and use device-specific thresholds to authenticate different devices. The thresholds for Samsung S8, Huawei P10 and Huawei P20 Pro are 9.32, 13.35 and 28.98 respectively. Under this setting, the authentication system achieved 100% TPR at 0%

Table 5.3. Success Rate of identity forgery attacks [%]:

Platform	S8		P10		P20pb	
	α/N	SR	α/N	SR	α/N	SR
Facebook	10/5	95	5/3	100	10/1	100
Flickr	3/1	100	3/1	100	3/1	100
Weibo	10/1	100	3/1	100	3/1	100
Wechat	10/3	100	5/5	100	5/3	100

FPR in identifying smartphones.

We then conduct identity forgery attacks on this system using images downloaded from the social accounts created in section 5.3.1. We extract the camera fingerprint of a victim device from her online images and embed the obtained fingerprint into an image captured by another device. The objective of this attack is to fool the verifier into believing that the image from the foreign camera is captured by the victim device. Table 5.3 lists the Success Rate (SR) of this attack under different settings. N is the number of images applied to estimate the victim's camera fingerprint and α is the embedding strength in equation 5.1. These two parameters are determined by the image compression strategy of the social platform as well as the fingerprint strength of the smartphone model. It can be observed that, using images downloaded from social platforms, the adversary can easily fabricate forged images and bypass the camera-based authentication system with high success rate.

5.3.3 The Impact of Post-processing

In practice, most users would prefer to post-process their images before uploading them to the social networks. We now evaluate the robustness of camera fingerprint against three major groups of post-processing approaches:

- Filter: this operation alters the shades and colors of an image at the pixel

Table 5.4. The impact of Filter: remaining camera fingerprint [%]

Filter	S8	P10	P20 Pro
Daily	69.34-84.84	49.38-69.68	35.87-60.22
Aqua	58.36-77.04	43.95-65.26	29.44-69.17
Jelly	82.46-93.01	58.33-71.42	34.24-62.10
Milk	63.76-76.61	52.01-67.76	31.48-59.35
Nature 1	49.61-65.72	40.82-52.57	28.22-49.77
Nature 2	47.11-69.61	37.60-54.44	21.64-43.95
Nature 3	49.14-69.36	37.18-53.00	23.34-53.49
Analog 1	39.96-67.93	40.02-53.08	26.34-51.86
Analog 2	52.60-77.90	40.35-51.67	26.86-55.63
Analog 3	41.89-63.18	42.33-52.24	23.86-48.13
Analog 4	63.88-78.58	49.99-68.08	28.38-57.19
Loveletter	42.91-64.82	31.63-45.94	18.73-55.65
Alight	62.25-80.04	48.95-64.55	26.92-55.30
Gleam	60.38-77.51	46.03-60.56	29.25-58.31
Pure	50.03-67.49	42.03-55.90	25.47-61.00

level in some manner (He, Sun and Tang, 2013; Gastal and Oliveira, 2011; Xu et al., 2011). It is normally being used to adjust the brightness and contrast of an image as well as to add a wide variety of tones and special effects to an image.

- Beautify: this operation is specially designed for portrait processing (Mawale and Chaugule, 2016; Batool and Chellappa, 2014; Ohchi, Sumi and Arakawa, 2010; Arakawa, 2004). It covers a rich set of tools for color correcting and shape fitting. Makeup editing is also allowed to beauty the appearance and physique of portrait photos.

- Add-on: this operation covers an image with user-defined stickers (e.g., emotion, graffiti and watermark). It is used to enrich and explain the images adequately.

We selected 15 filters, 15 beautifying operations and 9 add-ons from a phe-

nomenal camera APP named B612. The reason for choosing this APP is because it has a large group of users and allows a comprehensive set of post-processing techniques. Please note that, although the post-processing operations could be different across different APPs, their impact on camera fingerprint are very similar.

To evaluate the impact of a specific operation, we first capture an image using a specific smartphone. We then calculate the PCE value between the fingerprint of that image and a reference fingerprint of the device. The obtained PCE value indicates the relative fingerprint strength of that image. Next, we conduct the post-processing operation on that image and calculate the PCE value between the camera fingerprint of the post-processed image and the reference fingerprint. Finally, we divide the later PCE value by the former one. The obtained value indicates the percentage of camera fingerprint remained in the post-processed image.

For each operation under investigation, we repeat the above process with five different images. The range of the percentage value obtained from filter, beautify and add-ons operations are respectively listed in Table 5.4, 5.5 and 5.6. We make following observations: 1) Although most post-processing operations have reduced the camera fingerprint of the target image, none of these operations have completely eliminated the camera fingerprint of an image. This means that post-processed images can still be applied to conduct identity linking and identity forgery attacks. 2) The impact of post-processing operations varies slightly among different smartphone models. This is because different smartphone models normally have different fingerprint strength. According to our experimental results, the stronger the camera fingerprint is, the higher the percentage will be. 3) Beautify operations have the least impact on the camera

Table 5.5. The impact of Beautify: remaining camera fingerprint [%]

Beautify	S8	P10	P20 Pro
Skin	88.52-98.84	66.44-73.62	53.13-70.52
Brighten	69.02-88.16	53.70-67.84	37.85-63.07
Lift	79.53-94.84	61.44-72.94	41.71-70.81
Slim	83.45-97.71	64.47-76.55	49.27-72.76
Length	84.10-96.55	60.55-75.79	48.07-72.65
Chin	85.45-95.53	64.85-72.47	55.13-70.06
Wrinkles	88.24-99.07	67.78-78.74	51.66-75.09
Brow	87.20-98.76	67.85-79.10	54.49-77.68
Clarity	88.27-99.11	67.71-78.60	51.74-75.52
Enlarge	88.44-98.52	66.69-79.66	51.56-75.42
Narrow	88.11-98.61	65.01-78.20	48.65-76.47
Mouth	88.14-98.52	66.70-77.61	48.75-73.82
Contour	88.21-99.03	67.71-79.08	51.78-74.27
Blush	88.56-98.89	66.69-79.27	54.70-74.43
Lip color	88.24-98.99	67.59-79.01	51.61-74.96

Table 5.6. The impact of Add-ons: remaining camera fingerprint [%]

Add-ons	S8	P10	P20
Brush	39.11-53.33	29.22-46.51	16.85-42.49
Graffiti	42.42-60.61	36.12-54.79	21.21-52.05
Stickers	42.59-60.91	35.97-55.65	21.67-50.74
Mask	74.20-93.32	58.04-73.88	35.65-65.85
Glasses	35.92-61.13	35.49-48.66	18.31-52.94
Headwear	78.45-93.55	60.26-74.13	44.94-67.50
Cap	73.23-93.14	57.10-68.98	39.80-69.24
Background	70.29-86.93	53.49-72.08	42.82-65.25

fingerprint of the target image. This is because most Beautify operations only modify a small area of the image.

5.4 Camera Fingerprint Concealment

In this section, we present the detailed design of our fingerprint concealment system. It defeats PRNU-based identity linking and identity forgery attacks without hindering the beneficial applications of PRNU.

Figure 5.1. System Architecture.

5.4.1 Privacy-Preserving Architecture for Image Sharing

In order to protect users' camera fingerprints from being leaked, we propose to introduce an intermediary between smartphone cameras and social medias (as shown in Fig. 5.1). It can either be a plug-in for the camera or a separate app locally deployed on the smartphone. During the image-sharing process, the user first captures an image using the camera app or selects one from her photo album. She then conceals her camera fingerprint using our CFP system and uploads the anonymized image to the social network.

With the full consideration of practicability, we envision three design goals for the fingerprint concealment system:

- **Prevent malicious uses of PRNU**: adversaries who can only access to anonymized images should not be able to successfully conduct PRNU-based identity linking or identity forgery attacks.

- **Preserve beneficial uses of PRNU**: concealing the camera fingerprint of an image should not hinder the beneficial uses of PRNU. Specifically, a user in possession of a smartphone should be able to demonstrate whether an anonymized image is captured by this smartphone. Given an anonymized image and the smartphone that captured it, one should be able to demonstrate whether the image has been tampered.

- **Enable real-time fingerprint concealment**: the system should be able to conceal the camera fingerprint of an image in real-time.

5.4.2 Failure of Fingerprint Removal

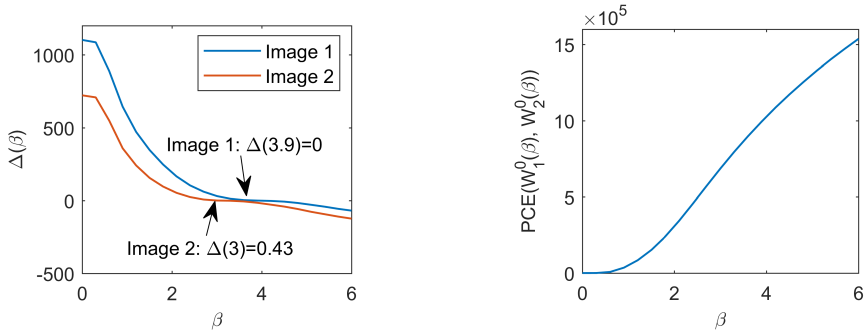
In the literature, image source anonymization is normally achieved through removing the camera fingerprint of the target image. If the removal approach can perfectly eliminate the camera fingerprint, an adversary with sanitized images will no longer be able to conduct identity linking or fabricate forged images. Therefore, before presenting our full-fledged system, we first discuss the feasibility of two state-of-the-art fingerprint removal approaches. The first approach can defeat identity forgery attacks but cannot address identity linking. The second approach can prevent all malicious uses of PRNU. However, both of the approaches are time-consuming and hinder the beneficial uses of PRNU due to the elimination of the camera fingerprint.

5.4.2.1 Adaptive Subtraction

This approach considers the camera fingerprint as a constant matrix added to the actual optical view. Given an input image $\mathbf{I}$, it first estimates a reference fingerprint $\hat{\mathbf{K}}_1$ from several images captured by the same device. Next, it calculates a sanitized image using

$$\mathbf{I}^0(\beta) = \mathbf{I} - \beta\hat{\mathbf{K}}_1,$$

where β is a strength factor for magnitude adjustment. In this round, the applied β is normally an empirical value and may not completely eliminate the fingerprint of the target image. Therefore, this approach then measures the



(a) Fingerprint strength

(b) PCE between sanitized images

Figure 5.2. The performance of Adaptive Subtraction: (a) The strength of the camera fingerprint on sanitized images decreases with the increasing β. The value of $\Delta(\beta)$ becomes negative when β is too large. (b) The similarity between the two sanitized images increases with β

strength of the remaining fingerprint in $\mathbf{I}^0(\beta)$ through

$$\Delta(\beta) = PCE(\mathbf{W}^0(\beta), \hat{\mathbf{K}}_2),$$

where, $\mathbf{W}^0(\beta)$ is the noise residue of $\mathbf{I}^0(\beta)$, $\hat{\mathbf{K}}_2$ is another reference fingerprint of the photographing device. Please note that $\hat{\mathbf{K}}_1$ and $\hat{\mathbf{K}}_2$ should be estimated from non-overlapping image sets. This is to ensure that $\mathbf{W}^0(\beta)$ and $\hat{\mathbf{K}}_2$ only share the fingerprint of the photographing device. Finally, Adaptive Subtraction adjusts the value of β based on $\Delta(\beta)$ and repeats the above process until $\Delta(\beta)$ approaches 0. The fingerprint-free image is the $\mathbf{I}^0(\beta)$ obtained in the last iteration.

To illustrate the performance of *Adaptive Subtraction* in removing the camera fingerprint, we captured two images using Samsung Galaxy S8 and conducted *Adaptive Subtraction*. Fig. 5.2(a) shows the value of $\Delta(\beta)$ obtained at different β. It can be seen that, with an appropriate β, it is possible to suppress the camera fingerprint of both images to a negligible amount. An adversary using such

sanitized images will not be able to extract the user's camera fingerprint and thus can not conduct identity forgery attacks.

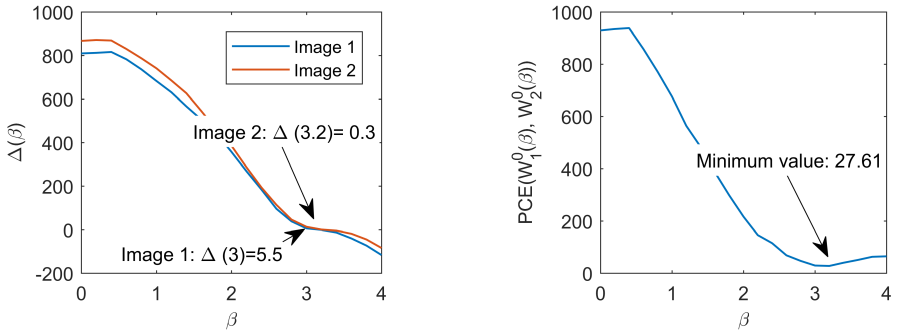
In this approach, it is important to ensure that the applied β can fully eliminate the camera fingerprint of the target image. Otherwise, the adversary might be able to estimate the user's fingerprint through analyzing multiple sanitized images. However, because the strength of camera fingerprint varies from one image to another, it normally requires multiple iterations to find the β for an image, which makes this approach time-consuming.

Another main issue with *Adaptive Subtraction* is that it can not prevent identity linking. The PCE values between sanitized images are actually increased after subtracting $\hat{\mathbf{K}}_1$. This is because $\hat{\mathbf{K}}_1$ is a combination of the camera fingerprint $\mathbf{K}$ and other noise components Ξ. Subtracting $\hat{\mathbf{K}}_1$ from an image is equivalent to introducing $-\hat{\mathbf{K}}_1$ into it. As a result, every image sanitized with the same $\hat{\mathbf{K}}_1$ will carry a significant amount of $-\Xi$, resulting in high PCE values between their noise residues. As shown in Fig. 5.2(b), the PCE value between image #1 and image #2 increases rapidly with increasing β. The increase of PCE actually makes it even easier for the adversary to conduct identity linking.

One obvious way to optimize *Adaptive Subtraction* is to generate new $\hat{\mathbf{K}}_1$ for each image to be anonymized. However, the generation of a new $\hat{\mathbf{K}}_1$ always requires the user to provide fresh reference images captured by the smartphone – that is, the user needs to take several new images whenever she tries to anonymize an image, which is not user-friendly.

5.4.2.2 Adaptive Denoising

This approach considers the camera fingerprint as a noise component and removes it using denoising filters. Given an image $\mathbf{I}$, *Adaptive Denoising* first ex-



(a) Fingerprint strength

(b) PCE between sanitized images

Figure 5.3. The performance of Adaptive Denoising: (a) The strength of the camera fingerprint on sanitized images. (b) The similarity between the sanitized image #1 and the sanitized image #2

tracts its noise components using a denoising filter F (e.g., Wiener filter). It then removes the extracted noise residue $\mathbf{W}$ from the image using:

$$\mathbf{I}^0(\beta) = \mathbf{I} - \beta\mathbf{W}$$

Here $\mathbf{W}$ is multiplied by a strength factor β because the denoising filter can hardly extract all the fingerprint components of the image. In practice, the value of β is always greater than one. After obtaining the sanitized image $\mathbf{I}^0(\beta)$, as with the *Adaptive Subtraction* method, *Adaptive Denoising* measures the remaining camera fingerprint in $\mathbf{I}^0(\beta)$ and adjusts the β accordingly until $\Delta(\beta)$ approaches 0.

The *Adaptive Denoising* approach has two major advantages over *Adaptive Subtraction*. First, it does not increase the PCE value between sanitized images. In *Adaptive Denoising*, an image $\mathbf{I}$ is anonymized through subtracting its own noise residue $\mathbf{W} = \mathbf{K} + \mathbf{\Xi}$, meaning that the random noise component $\mathbf{\Xi}$ is different for different images. Therefore, although removing $\mathbf{W} = \mathbf{K} + \mathbf{\Xi}$ from $\mathbf{I}$

will still introduce $-\Xi$ into the sanitized image, such $-\Xi$ will not affect the PCE value between different sanitized images. To illustrate, Fig. 5.3 plots the performance of *Adaptive Denoising* on two images captured by Samsung Galaxy S8. It can be observed that the similarity between sanitized images can be decreased to a very low level. Second, *Adaptive Denoising* normally requires less iterations to find the appropriate β for an image. This is because $\mathbf{W}$ is a noise residue extracted from the target image $\mathbf{I}$. The amount of camera fingerprint in $\mathbf{W}$ is roughly proportional to that in $\mathbf{I}$, which narrows the search for the appropriate β.

Unfortunately, although *Adaptive Denoising* is able to prevent malicious uses of PRNU, it has the following drawbacks: 1) It cannot guarantee real-time fingerprint concealment. Although *Adaptive Denoising* is a bit more time-efficient than *Adaptive Subtraction*, it still requires more than three iterations on average to find the appropriate β. As will be shown in section 5.5.4, it takes the tested computer tens of seconds to finish three iterations of adaptive denoising. 2) It cannot preserve beneficial uses of PRNU. Due to the removal of the camera fingerprint, a user can no longer use the camera fingerprint to prove the copyright or determine the integrity of an image.

5.4.3 Obfuscation-based Fingerprint Concealment

Since removing the camera fingerprint of an image is time-consuming and cannot preserve beneficial uses of PRNU, we propose to conceal the camera fingerprint of an image through fingerprint obfuscation. Specifically, our system tries to "cover up" the camera fingerprint of the target image through embedding it with an obfuscating noise $\mathbf{O}$. Given an image $\mathbf{I}$, the anonymized image $\mathbf{I}'$ can be

generated as:

$$\mathbf{I}' = \mathbf{I} + \alpha\mathbf{O},$$

The noise $\mathbf{O}$ remains constant for all photographing devices, while the strength factor α varies from one image to another. In this way, all obfuscated images will have a common noise component regardless of whether the images are captured by the same photographing device. This common component not only renders the identity linking attack ineffective but also serves as a probe signal that defeats the identity forgery attack. We now explain the rational behind and how the design goals are achieved.

In identity linking attacks, the adversary estimates a camera fingerprint for each social account and checks the PCE values between the estimated fingerprints. Because the PCE value between two matching fingerprints is almost always higher than that between two non-matching fingerprints, the adversary can use thresholding to determine if two fingerprints belong to the same photographing device. In obfuscation-based fingerprint concealment system, all images uploaded to social accounts are embedded with the same obfuscating noise with varying strength. The PCE value between two estimated fingerprints is not only affected by the camera fingerprint, but also by the obfuscating noise. To illustrate the impact of the obfuscating noise, Fig. 5.4 shows the PCE distributions of 200 matching fingerprint pairs and 200 non-matching fingerprint pairs before and after the obfuscation process. It can be seen that, after the obfuscation process, the PCE distributions of matching and non-matching fingerprint pairs have become remarkably similar. This is because the obfuscating noise is much higher in energy than the camera fingerprint, thus becoming the dominant factor affecting the PCE value between two fingerprints. As a result, the

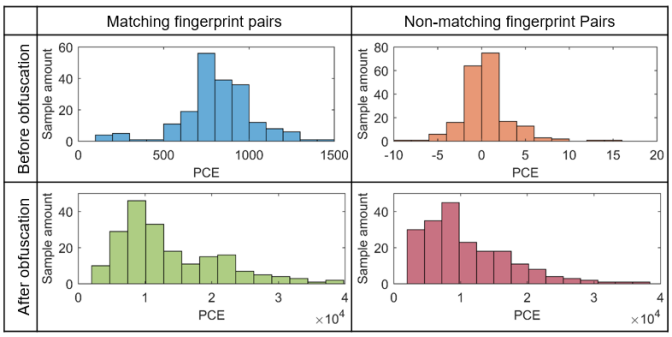


Figure 5.4. The impact of an obfuscating noise on the PCE distributions of 200 matching fingerprint pairs and 200 non-matching fingerprint pairs. The obfuscating noise is a Gaussian noise with mean 0 and variance 1. The strength factor α for each image is a randomly selected floating point number between 1 and 2.

adversary can no longer use thresholding to determine if two estimated fingerprints belong to the same device.

In identity forgery attacks, the adversary extracts a victim's camera fingerprint from her online images and injects the obtained fingerprint into an image captured by another device, in the hope of fooling a third party into believing that the forged image is captured by the victim's camera. However, in the case that any of the images collected by the adversary are obfuscated ones, the extracted fingerprint will carry a significant amount of the obfuscating noise, and the third party can easily detect forged images through checking the existence of the obfuscating noise.

For beneficial uses of PRNU, a user in possession of the photographing device can still demonstrate whether an obfuscated image is captured by her device or whether the image has been tampered. This is because obfuscation-based fingerprint concealment does not remove the camera fingerprint of the target image, and the device owner can provide "clean" images that do not contain the obfuscating noise. Because the PCE value between an obfuscated image and a clean one will only be affected by the camera fingerprint, this PCE value

can still be used to conduct image-to-camera matching and integrity verification. More details will be presented in section 5.5.3.

5.4.4 Defeating De-Obfuscation Attacks

Since the obfuscating noise remains constant for all photographing devices, it is likely that an adversary could obtain this noise and try to remove it from an obfuscated image to recover the original one. In particular, because the obfuscating noise is of the same type as a camera fingerprint (i.e., Gaussian noise), the recovery of the original image can be achieved using the Adaptive Subtraction approach introduced in section 5.4.2. We refer to this attack as the *De-Obfuscation Attack*.

Given an obfuscated image $\mathbf{I}'$ and the obfuscating noise $\mathbf{O}$, the adversary first calculates an estimation $\mathbf{I}^E(\beta)$ of the original image through

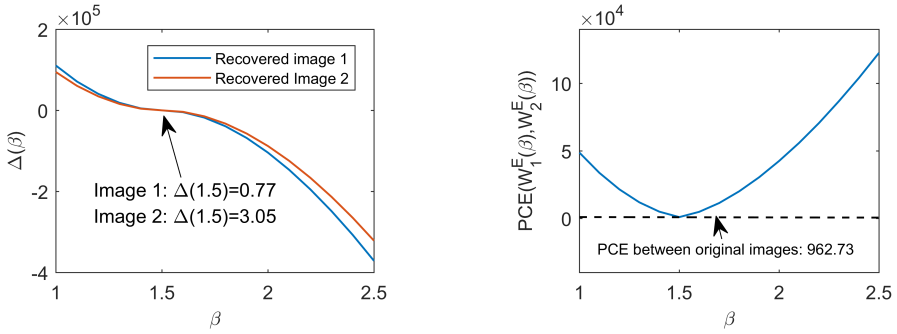
$$\mathbf{I}^E(\beta) = \mathbf{I}' - \beta\mathbf{O},$$

where β is an estimation of the embedding strength α. She then measures the remaining $\mathbf{O}$ in $\mathbf{I}^E(\beta)$ through

$$\Delta(\beta) = PCE(\mathbf{W}^E(\beta), \mathbf{O}),$$

where $\mathbf{W}^E(\beta)$ is the noise residue of $\mathbf{I}^E(\beta)$. Finally, the adversary adjusts the value of β based on $\Delta(\beta)$ and repeats the above process until $\Delta(\beta)$ approaches 0. The original image is the $\mathbf{I}^E(\beta)$ obtained in the last iteration.

Fig. 5.5 illustrates the performance of *Adaptive Subtraction* on two obfuscated images $\mathbf{I}'_1$ and $\mathbf{I}'_2$. For both images, the obfuscating noise is a Gaussian noise with



(a) The strength of the obfuscation noise in recovered images

(b) PCE between recovered images

Figure 5.5. The performance of Adaptive Subtraction on images obfuscated through the basic obfuscation function: $\mathbf{I}' = \mathbf{I} + \alpha\mathbf{O}$

mean 0 and variance 0.25, and the strength factor α is set to 1.5. It can be seen that, when β is equal to α, the obfuscating noise $\mathbf{O}$ in the estimated image becomes negligible (Fig. 5.5(a)), and the PCE value between the recovered images is very similar to the PCE value between the original images (Fig. 5.5(b)). The original images was successfully recovered.

The reason why this attack is possible is because the basic obfuscating function $\mathbf{I}' = \mathbf{I} + \alpha\mathbf{O}$ is invertible and the search space of α is small. To address this issue, we propose the robust obfuscation function

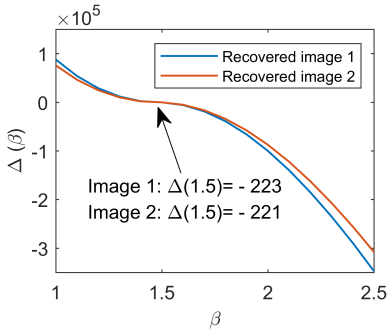
$$\mathbf{I}' = \mathbf{I} + \mathbf{A} \circ \mathbf{O},$$

where $\mathbf{A}$ is a onetime matrix with the same resolution as $\mathbf{O}$ and $\mathbf{A} \circ \mathbf{O}$ is the Hadamard product of the two matrices. With this design, the obfuscating noise will be distorted by the onetime matrix $\mathbf{A}$ and the following integer conversion process. The noise $\mathbf{O}'$ embedded in $\mathbf{I}'$ is similar but different from the original obfuscating noise $\mathbf{O}$. Because $\mathbf{A}$ is a onetime matrix consisting of millions of elements (for most images), the adversary will not be able to obtain $\mathbf{O}'$ and can

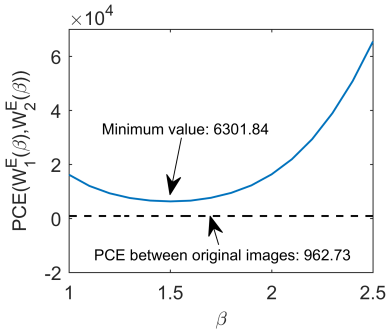
only use $\mathbf{O}$ to conduct adaptive subtraction. However, as has been shown in section 5.4.2, the adaptive subtraction approach fails if the subtracted matrix contains noise components that are not present in the target image. Denoting the common components between $\mathbf{O}'$ and $\mathbf{O}$ as $\mathbf{O}_c$, the components that only present in $\mathbf{O}$ as $\mathbf{O}_\delta$, the noise residue of a recovered image can be rewrite as:

$$\begin{aligned} \mathbf{W}^E(\beta) &= \gamma\mathbf{O}_c - \beta(\mathbf{O}_c + \mathbf{O}_\delta) + \Theta, \\ &= (\gamma - \beta)\mathbf{O}_c + \beta(-\mathbf{O}_\delta) + \Theta, \end{aligned} \tag{5.2}$$

where γ represents the strength of $\mathbf{O}_c$ in $\mathbf{I}'$, Θ represents other noise components (such as random noise components, camera fingerprint, etc.). Although $\mathbf{O}_c$ and $\mathbf{O}_\delta$ could be different for different images, these noise components will still affect the PCE value between recovered images significantly. To illustrate, Fig. 5.6 shows the performance of Adaptive Subtraction on two images constructed using the robust obfuscating function. The matrix $\mathbf{A}$ for each image is a Gaussian noise with mean 1.5 and variance 0.25. It can be seen that, although $\Delta(\beta)$ is close to zero when β equals to the mean of $\mathbf{A}$, the PCE value between the recovered images are always significantly higher than the PCE value between original images. The noise component $(\alpha - \beta)\mathbf{O}_c + \beta(-\mathbf{O}_\delta)$ is always the dominant factor that affects the PCE value between two fingerprints, which renders the adversary unable to recover the original images that only share the camera fingerprint of the photographing device. Please note that the Adaptive Denoising approach also does not work because it will eliminate the camera fingerprint of the target image. The recovered images cannot be used for identity linking or identity forgery.



(a) PCE between the recovered image and the obfuscating noise



(b) PCE between recovered images

Figure 5.6. The performance of Adaptive Subtraction on images obfuscated through the robust obfuscation function: $\mathbf{I}' = \mathbf{I} + \mathbf{A} \circ \mathbf{O}$

5.4.5 System Design

Using the above mechanisms as building blocks, we now present the full fledged fingerprint concealment system. The system is installed on a smartphone with a Gaussian noise that remains constant for all photographing devices. This noise has a high resolution and is used to generate obfuscating noises of different resolutions. Upon receiving an image that needs to be anonymized, the system conceals its fingerprint using following steps:

1) Noise Generation. The system generates two matrices with the same resolution as the target image. The first matrix $\mathbf{O}$ is an obfuscating noise generated through resampling the Gaussian noise stored in the system. The second matrix $\mathbf{A}$ is a onetime Gaussian noise generated using the mean and variance randomly selected within a particular range. The mean value determines the strength of the embedded noise.

2) Noise Embedment. The system calculates the Hadamard product of $\mathbf{A}$ and $\mathbf{O}$ and inject the obtained matrix into the target image using:

$$\mathbf{I}' = \mathbf{I} + \mathbf{A} \circ \mathbf{O}.$$

Table 5.7. Intra-Platform Identity Linking using obfuscated images: N is the number of images applied to estimate the fingerprint of an account. ω is the threshold. TPR is calculated at 5% FPR.

Platform/N	ω [10^4]			TPR [%]		
	S8	P10	P20	S8	P10	P20
Facebook/3	4	6	18	11.32	5.79	7.37
Facebook/5	9	10	30	10.00	7.63	7.89
Flickr/3	71	74	79	7.89	6.32	3.42
Flickr/5	108	114	107	11.05	7.89	6.84
Weibo/3	76	56	76	10.79	6.05	5.53
Weibo/5	115	74	108	14.47	5.26	8.95
Wechat/3	4	1	5	11.32	5.53	5.97
Wechat/5	9	3	9	10.00	7.63	6.32

Table 5.8. Inter-Platform Identity Linking using obfuscated images: N is the number of images applied to estimate the fingerprint of an account. ω is the threshold. TPR is calculated at 5% FPR.

N	ω [10^4]			TPR [%]		
	S8	P10	P20	S8	P10	P20
3	8.29	2.84	9.06	11.19	11.47	10.19
5	14.78	5.24	15.42	13.66	12.28	12.07

The obtained $\mathbf{I}'$ is a matrix of type double, and its elements are not necessarily in the range 0 to 255.

3) Integer Conversion. The system converts all the elements of $\mathbf{I}'$ to 1-byte unsigned integers. Elements within the range of 0 to 255 are rounded to the nearest integer and elements outside are mapped to the nearest endpoint.

4) Image Export. The system exports the obtained matrix $\mathbf{I}'$ to a user-desired format such as JPEG and PNG. Please note that, for image formats that employ lossy compression, the exported image will contain less fingerprint than $\mathbf{I}'$ but will be more robust against de-obfuscation attacks.

5.5 Performance Evaluation

5.5.1 Experimental Methodology

Configuration: The evaluation is conducted using Matlab on a laptop with an 8-core CPU running at 2.8 GHz. The algorithm for camera fingerprinting is based on the code published by the digital data embedding laboratory (Goljan, Fridrich and Filler, 2009). The obfuscating noise **O** is a constant Gaussian noise with mean 0 and variance 0.25. For the onetime matrix **A**, its mean is a randomly selected floating number between 2 and 5 and its variance is 1.

Devices and Platforms: we employ 6 smartphones of 3 different models: i) 2 Samsung S8; ii) 2 HUAWEI P10; iii) 2 HUAWEI P20 Pro. We evaluate images downloaded from four social platforms: Facebook, Wechat, Weibo, and Flickr.

Metric: True Positive Rate (TPR) and False Positive Rate (FPR) measures the likelihood that a positive/negative sample is correctly/wrongly detected as a positive sample.

5.5.2 Preventing Malicious Applications of PRNU

We now demonstrate the effectiveness of obfuscation-based fingerprint concealment in defeating identity linking and identity forgery attacks.

5.5.2.1 Identity Linking Attacks

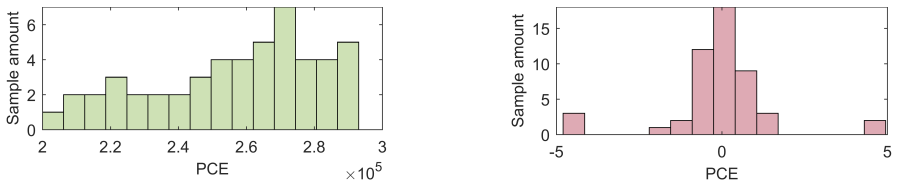
We repeat the identity linking experiment in section 5.3.1 with obfuscated images. For each user (smartphone), we created 20 social accounts on each platform and uploaded 5 obfuscated images to each account. We then download images from these accounts and conduct Intra-platform identity linking and

Inter-platform identity linking. Unlike the previous experiments in section 5.3.1, this experiment considered the worst case scenario where the adversary already knew the *smartphone model* of each account. In this case, the adversary can use model-specific thresholds to differentiate individual smartphones and achieve better matching results.

Table 5.7 and 5.8 list the TPR under different settings at 5% FPR. It can be observed that, because of the existence of the obfuscating noise, the adversary can hardly determine whether two anonymous accounts belong to the same users. The threshold varies significantly across different settings because the strength of the obfuscating noise in an estimated fingerprint is affected by the many different factors, such as the number of applied images, the compression algorithm, and the image resolution. If the adversary does not know in advance the smartphone model of each account, the identity linking results could be even worse. Moreover, there was no significant increase in the TPR when the number of images increases. This is because increasing the number of images will increase not only the strength of the camera fingerprint but also the strength of the obfuscating noise.

5.5.2.2 Identity Forgery Attacks

Due to the preservation of the camera fingerprint, obfuscated images can still be applied to conduct identity forgery attacks. However, the forged images will inevitably carry a significant amount of obfuscating noise, and thus can be easily detected. In particular, upon receiving an image, the verifier could extract its camera fingerprint and calculate the PCE value between the estimated fingerprint and the obfuscating noise $\mathbf{O}$. If the PCE value is higher than a threshold, the image must carries the obfuscating noise and thus can be determined as



(a) Forged image set (b) Genuine image set

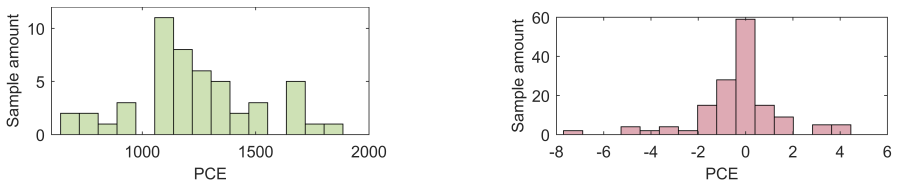
Figure 5.7. Forgery detection: the PCE values obtained from forged images are significantly higher than the PCE value obtained from genuine images

forged.

To evaluate the effectiveness of the forgery detection mechanism, we integrated it into the authentication system implemented in section 5.3.1 and tested the system with two sets of images: 1) genuine set: 50 images captured by the victim's smartphone (a Huawei P10); 2) forged set: 50 foreign images embedded with the victim's camera fingerprint. Each victim fingerprint is estimated from 10 obfuscated images downloaded from the victim's Flickr account. Fig. 5.7 shows the distributions of the PCE values obtained from the forgery detection mechanism. It can be observed that, because of the existence of the obfuscating noise, the PCE values obtained from forged images are significantly higher. By setting the threshold at 50, the authentication system was able to detect forged images with 100% TPR at 0% FPR.

5.5.3 Preservation of Beneficial Applications of PRNU

Because obfuscated images still carry the camera fingerprint, the beneficial applications of PRNU are preserved. The following illustrates how to conduct *Copyright Protection* and *Integrity Validation* on obfuscated images.



(a) The user's images (b) Foreign images

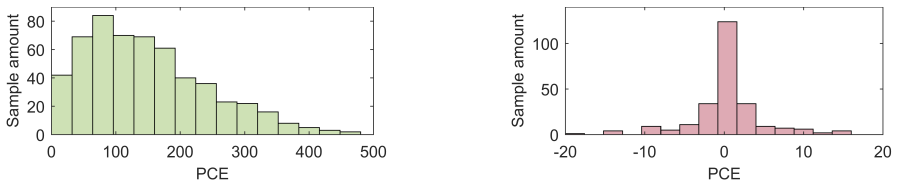
Figure 5.8. Copyright Protection: the PCE values obtained from the user's images are significantly higher than that obtained from foreign images

5.5.3.1 Copyright Protection

To demonstrate the copyright of an obfuscated image, the user first captures several unobfuscated images using her smartphone. She then estimates a reference fingerprint from the captured images and calculate the PCE value between the reference fingerprint and the obfuscated image. Because the reference fingerprint does not contain the obfuscation noise, a high PCE means the obfuscated image carries the camera fingerprint of the user's smartphone. To demonstrate the feasibility of this approach, we assume a user with a Samsung S8 and construct 200 obfuscated images, of which 50 are captured by the user's smartphone, the resting are captured by five other devices. We then calculate the PCE value between the noise residue of each image and a reference fingerprint estimated from 5 unobfuscated images. Fig.5.8 plots the PCE distributions of different images. It can be observed that the obtained PCE values are not affected by the obfuscating noise. Using thresholding, the user can demonstrate her copyright of the images with 100% TPR at 0% FPR.

5.5.3.2 Integrity Verification

To validate the integrity of an obfuscated image, the user first estimates a high quality reference fingerprint from multiple unobfuscated images captured by



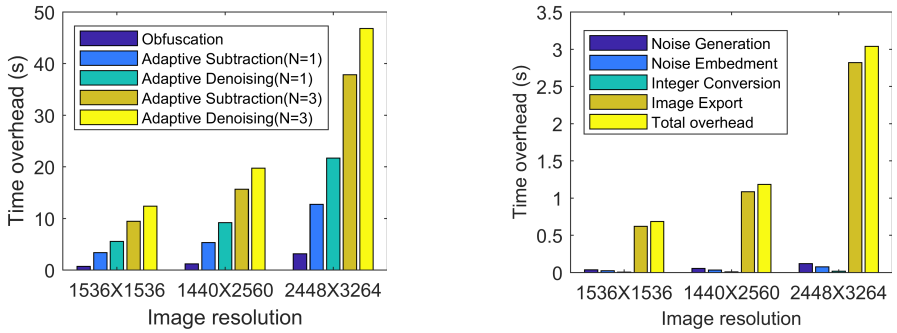
(a) Unmodified blocks (b) Modified blocks

Figure 5.9. Integrity Verification: the PCE values obtained from modified blocks are significantly lower than that obtained from unmodified blocks

her smartphone. She then divides the target image and the reference fingerprint into N blocks and calculates the PCE value between corresponding blocks of the target image and the reference fingerprint. After that, she use thresholding to determine if a block has been modified. To demonstrate the feasibility of this approach, we constructed 50 obfuscated images, divided each image into 16 blocks, and modified randomly selected blocks. We then conduct integrity verification using a reference fingerprint estimated from 20 images. Fig.5.9 shows the PCE distribution of modified and unmodified blocks. We chose 50 as the threshold and achieved a 98% TPR at 0% FPR in detecting modified blocks.

5.5.4 Time Overhead

We now demonstrate the efficiency of the obfuscation-based approach. Fig. 5.10(a) compares the time overhead of *Adaptive Subtraction*, *Adaptive Denoising*, and *Obfuscation-based Fingerprint Concealment* under three common image resolutions. Each overhead value is an average of 20 times of anonymization. It can be observed that, the obfuscation-based approach is much more time efficient compared with other approaches. For *Adaptive Subtraction* and *Adaptive Denoising*, the latency is mainly caused by the fingerprint extraction and matching operations involved in their iterative procedure. The time overhead



(a) Time overhead of different anonymization approaches

(b) Time overhead of individual procedures in image obfuscation

Figure 5.10. Time overhead. N is the number of times the iterative procedure is executed during the anonymization process. Anonymized images are exported to the PNG format.

increases rapidly with the image resolution and the number of times the iterative procedure is executed. The reason why *Adaptive Denoising* is more time-consuming than *Adaptive Subtraction* under the same setting is because it needs to extract the noise residue of the target image before the iterative process. For the obfuscation-based approach, the latency is mainly caused by the image export process (as shown in Fig. 5.10(b)). The generation and embedment of the obfuscation noise can always be finished within 250 millisecond for all images under investigation. It is easy for modern smartphones to obfuscate images in real-time.

5.6 Conclusion

Posting images on social platforms has become an important part of modern life. Most people, however, don't realize that every image posted on the Internet could expose the unique hardware fingerprint of their photographing device to the public. In this paper, we highlight the security and privacy issues

caused by this fingerprint leakage and propose a practical fingerprint concealment system to address them. As demonstrated in this paper, with a handful of images downloaded from current image sharing platforms, an adversary can easily conduct identity linking and identity forgery attacks with high success rate. Our system, CFP, seeks to address these attacks through embedding the target image with an obfuscating noise. By carefully designing this noise component, we demonstrate that such noise can significantly reduce the success rate of identity linking attacks and can serve as a reliable probe for the detection of identity forgery attacks. Moreover, because our obfuscation-based mechanism does not remove the camera fingerprint of the target image, obfuscated images can still be utilized for beneficial forensic tasks (e.g., copyright protection and integrity verification). The proposed system enables smartphone users to enjoy image sharing without risking privacy.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this dissertation, we investigated hardware-rooted device authentication and carried out an in-depth study on a powerful hardware fingerprint named Photo Response Non-Uniformity. We summarize our results as follows:

In chapter 2, we study the feasibility of hardware-rooted device authentication through analyzing a variety of hardware fingerprinting approaches. We first describe the architecture of hardware-rooted authentication systems, focusing on the players involved and the communication channels. Two kinds of challenge-response schemes are presented to collect the output signals of different transducers. We then analyze the security threats underlying these schemes and list several desirable properties for a usable hardware fingerprinting method. After that, we study several existing fingerprinting methods and discuss their performance under replay attacks and fingerprint forgery attacks.

In chapter 3, we study a specific hardware fingerprint named Photo Response Non-Uniformity (PRNU) and explore the feasibility of utilizing the

PRNU as a smartphone's unique fingerprint to implement 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 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.

In chapter 4, we discuss a limitation of the ABC system's forgery detection mechanism and present new premitives for the forgery detection of PRNU. We present two novel observations of smartphones taking pictures: 1) There exists a noisechain embedded in continuously captured burst images. 2) The camera fingerprint and noise components of an image are correlated with the movement of the photographing device. We explore these two observations, design two reliable forgery detectors for the detection of forgery attacks against PRNU-based camera fingerprinting, and propose *CIM*, a camera-based smartphone authentication system. *CIM* is practical since it leverages universal sensors (camera and accelerometer) in a smartphone and a user just needs to take pictures in burst mode while moving the smartphone along a simple route. Extensive experiments are conducted to validate the effectiveness of *CIM* against fingerprint forgery attacks. A user can submit either multiple or one image for authentication. In both cases, *CIM* achieves 100% TAR at 0% FAR in both fingerprint matching and forgery detection.

In chapter 5, we report the problem of camera fingerprint leakage in current image sharing systems and propose a privacy-preserving architecture to address it. As demonstrated in this paper, with a handful of images downloaded from current image sharing platforms, an adversary can easily conduct identity linking and identity forgery attacks with high success rate. Our system, CFP, seeks to address these attacks through embedding the target image with an obfuscating noise. By carefully designing this noise component, we demonstrate that such noise can significantly reduce the success rate of identity linking attacks and can serve as a reliable probe for the detection of identity forgery attacks. Moreover, because our obfuscation-based mechanism does not remove the camera fingerprint of the target image, obfuscated images can still be utilized for beneficial forensic tasks (e.g., copyright protection and integrity verification). The proposed system enables smartphone users to enjoy image sharing without risking privacy.

6.2 Future Work

As IoT systems involve a wide variety of devices, networks, gateways, applications, and services, there is a wide range of potential vulnerabilities with multiple attack surfaces, making IoT security a rich topic with diverse possible avenues of investigation. We identify following issues for future work:

Smartphone-Centric IoT Attacks and Defenses: In the context of IoT security, the smartphone plays a very intriguing dual role. On the one hand, it could be used as a low-cost attacking device. Modern smartphones are equipped with rich on-board sensors and are able to access various side channels of IoT devices. An adversary using a smartphone can inconspicuously launch his attack

because of the portability and pervasiveness of smartphones. My research in this direction will focus on revealing the vulnerability of current IoT devices against smartphone-based side channel attacks and on developing countermeasures. On the other hand, due to its sufficient computational power, various device connection capabilities, and convenient user interface, the smartphone could also be used as a security hub to provide the first line of defense for IoT devices. Using the smartphone as the security hub not only empowers the users to enforce security policy across devices, but also eases the way users manage their personal information. I will investigate techniques under diverse hardware and cryptographic limitations to support the device management tasks with rigorous security and privacy guarantees.

Artificial Intelligence Enabled IoT Security: The battle filed of IoT security is rapidly shifting due to the accelerated development of Artificial Intelligence (AI). On the one hand, AI techniques offer new tools to help organizations secure IoT systems from malicious incursions. It has been exploited to analyze and recognize patterns of security vulnerabilities and are expected to react more effectively to new threats than traditional approaches. On the other hand, AI techniques magnify existing IoT vulnerabilities and bring in new ones. Hackers may employ AI techniques themselves to develop increasingly sophisticated attacks. For instance, adversarial machine learning use intentionally designed adversarial samples to cause an AI model to make a mistake, resulting in a mismatch between the physical and cyber world. In safety-critical applications like medical devices and automobiles, such mismatch could bring great danger to human lives. My research in this direction will focus on exploring the implication of Artificial Intelligence on physical-layer IoT security. In particular, I will explore the novel ideas and techniques to strengthen robustness and reliance

for hardware-rooted device identification, detect and recover forged/modified sensor measurements fabricated through AI technqiues (e.g., Generative Adversarial Network), and identify and mitigate AI enabled cyber physical attacks against IoT systems.

Bibliography

Amerini, Irene, Rudy Becarelli, Roberto Caldelli, Alessio Melani and Moreno Niccolai. 2017. "Smartphone fingerprinting combining features of on-board sensors." *IEEE Transactions on Information Forensics and Security* 12(10):2457–2466.

Arakawa, Kaoru. 2004. Nonlinear digital filters for beautifying facial images in multimedia systems. In *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*. Vol. 5 IEEE pp. V–V.

Armstrong, Martin. 2017. "Smartphone Life Cycles Are Changing." https://www.statista.com/chart/8348/.

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

Ba, Zhongjie, Sixu Piao and Kui Ren. 2017. Defending against Speaker Fingerprinting Based Device Tracking for Smartphones. In *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE pp. 188–189.

Ba, Zhongjie, Sixu Piao, Xinwen Fu, Dimitrios Koutsonikolas, Aziz Mohaisen and Kui Ren. 2018. ABC: Enabling Smartphone Authentication with Built-in Camera. In *NDSS*.

Baldini, Gianmarco and Gary Steri. 2017. "A survey of techniques for the identification of mobile phones using the physical fingerprints of the built-in components." *IEEE Communications Surveys & Tutorials* 19(3):1761–1789.

Batool, Nazre and Rama Chellappa. 2014. "Detection and inpainting of facial wrinkles using texture orientation fields and Markov random field modeling." *IEEE transactions on image processing* 23(9):3773–3788.

Bayram, Sevinç, Husrev T Sencar and Nasir D Memon. 2013. Seam-carving based anonymization against image & video source attribution. In *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*. IEEE pp. 272–277.

Böhme, Rainer and Matthias Kirchner. 2013. Counter-forensics: Attacking image forensics. In *Digital Image Forensics*. Springer pp. 327–366.

Bojinov, Hristo, Yan Michalevsky, Gabi Nakibly and Dan Boneh. 2014. "Mobile device identification via sensor fingerprinting." *arXiv preprint arXiv:1408.1416* .

Bravo-Solorio, Sergio and Asoke K Nandi. 2011. "Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics." *Signal Processing* 91(8):1759–1770.

Breach Level Index H1 2018 Infographic. N.d. https://safenet.gemalto.com/resources/data-protection/breach-level-index-2018-h1/.

Brik, Vladimir, Suman Banerjee, Marco Gruteser and Sangho Oh. 2008. Wireless device identification with radiometric signatures. In *MobiCom*. ACM pp. 116–127.

Cain, Stephen C, Majeed M Hayat and Ernest E Armstrong. 2001. "Projection-based image registration in the presence of fixed-pattern noise." *IEEE transactions on image processing* 10(12):1860–1872.

Caldell, Roberto, Irene Amerini, Francesco Picchioni, Alessia De Rosa and Francesca Uccheddu. 2010. Multimedia forensic techniques for acquisition device identification and digital image authentication. In *Handbook of Research on Computational Forensics, Digital Crime, and Investigation: Methods and Solutions*. IGI Global pp. 130–154.

Caldelli, Roberto, Irene Amerini and Andrea Novi. 2011. An analysis on attacker actions in fingerprint-copy attack in source camera identification. In *Information Forensics and Security (WIFS), 2011 IEEE International Workshop on*. IEEE pp. 1–6.

Cassavoy, Liane. 2018. "How Fast Is 4G LTE Wireless Service?" https://www.lifewire.com/how-fast-is-4g-wireless-service-577566.

Chen, Dajiang, Ning Zhang, Zhen Qin, Xufei Mao, Zhiguang Qin, Xuemin Shen and Xiang-Yang Li. 2017. "S2M: A lightweight acoustic fingerprints-based wireless device authentication protocol." *IEEE Internet of Things Journal* 4(1):88–100.

Chen, Dajiang, Xufei Mao, Zhen Qin, Weiyi Wang, Xiang-Yang Li and Zhiguang Qin. 2015. Wireless device authentication using acoustic hardware fingerprints. In *International Conference on Big Data Computing and Communications*. Springer pp. 193–204.

Chen, Mo, Jessica Fridrich and Miroslav Goljan. 2007. Digital imaging sensor identification (further study). In *Security, Steganography, and Watermarking of Multimedia Contents IX*. Vol. 6505 International Society for Optics and Photonics p. 65050P.

Chen, Mo, Jessica Fridrich, Miroslav Goljan and Jan Lukás. 2008. "Determining image origin and integrity using sensor noise." *IEEE Transactions on Information Forensics and Security* 3(1):74–90.

Chen, Si, Kui Ren, Sixu Piao, Cong Wang, Qian Wang, Jian Weng, Lu Su and Aziz Mohaisen. 2017. You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE pp. 183–195.

Christin, Delphine, Pablo Sánchez López, Andreas Reinhardt, Matthias Hollick and Michaela Kauer. 2013. "Share with strangers: Privacy bubbles as user-centered privacy control for mobile content sharing applications." *information security technical report* 17(3):105–116.

Danev, Boris, Davide Zanetti and Srdjan Capkun. 2012. "On physical-layer identification of wireless devices." *ACM Computing Surveys (CSUR)* 45(1):6.

Danev, Boris, Heinrich Luecken, Srdjan Capkun and Karim El Defrawy. 2010. Attacks on physical-layer identification. In *Proceedings of the third ACM conference on Wireless network security*. ACM pp. 89–98.

Danev, Boris, Thomas S Heydt-Benjamin and Srdjan Capkun. 2009. Physical-layer Identification of RFID Devices. In *Usenix Security Symposium*. pp. 199–214.

Das, Anupam, Nikita Borisov and Edward Chou. 2018. "Every Move You Make: Exploring Practical Issues in Smartphone Motion Sensor Fingerprinting and Countermeasures." *Proceedings on Privacy Enhancing Technologies* 2018(1):88–108.

Das, Anupam, Nikita Borisov and Matthew Caesar. 2014*a*. Do you hear what i hear?: Fingerprinting smart devices through embedded acoustic components. In *CCS*. ACM pp. 441–452.

Das, Anupam, Nikita Borisov and Matthew Caesar. 2014*b*. "Fingerprinting smart devices through embedded acoustic components." *arXiv preprint arXiv:1403.3366* .

Das, Anupam, Nikita Borisov and Matthew Caesar. 2015. "Exploring ways to mitigate sensor-based smartphone fingerprinting." *arXiv preprint arXiv:1503.01874* .

Das, Anupam, Nikita Borisov and Matthew Caesar. 2016. Tracking Mobile Web Users Through Motion Sensors: Attacks and Defenses. In *NDSS*.

De Choudhury, Munmun, Hari Sundaram, Yu-Ru Lin, Ajita John and Doree Duncan Seligmann. 2009. Connecting content to community in social media via image content, user tags and user communication. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*. IEEE pp. 1238–1241.

Dey, Sanorita, Nirupam Roy, Wenyuan Xu, Romit Roy Choudhury and Srihari Nelakuditi. 2014. AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable. In *NDSS*.

Dirik, Ahmet Emir and Ahmet Karaküçük. 2014. "Forensic use of photo response non-uniformity of imaging sensors and a counter method." *Optics express* 22(1):470–482.

Dirik, Ahmet Emir, Hüsrev Taha Sencar and Nasir Memon. 2014. "Analysis of seam-carving-based anonymization of images against PRNU noise pattern-based source attribution." *IEEE Transactions on Information Forensics and Security* 9(12):2277–2290.

Edman, Matthew and Bülent Yener. 2009. "Active attacks against modulation-based radiometric identification." *Rensselaer Institute of Technology, Technical report* pp. 09–02.

Enterprise, Verizon. 2017. "2017 Data breach investigations report.".

Fang, Song, Yao Liu and Peng Ning. 2016. "Mimicry attacks against wireless link signature and new defense using time-synched link signature." *IEEE Transactions on Information Forensics and Security* 11(7):1515–1527.

Farid, Hany. 2009. "Image forgery detection." *IEEE Signal processing magazine* 26(2):16–25.

Fridrich, Jessica. 2009*a*. "Digital image forensics." *IEEE Signal Processing Magazine* 26(2).

Fridrich, Jessica J. 2009*b*. "Digital Image Forensics Using Sensor Noise." *IEEE Signal Processing Magazine* pp. 26–37.

Gastal, Eduardo SL and Manuel M Oliveira. 2011. Domain transform for edge-aware image and video processing. In *ACM Transactions on Graphics (ToG)*. Vol. 30 ACM p. 69.

Gloe, Thomas, Matthias Kirchner, Antje Winkler and Rainer Böhme. 2007. Can we trust digital image forensics? In *Proceedings of the 15th ACM international conference on Multimedia*. ACM pp. 78–86.

Gohshi, Seiichi, Haruyuki Nakamura, Hiroshi Ito, Ryousuke Fujii, Mitsuyoshi Suzuki, Shigenori Takai and Yukari Tani. 2005. 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*. Springer pp. 1099–1107.

Goljan, Miroslav. 2008. Digital camera identification from images–estimating false acceptance probability. In *International Workshop on Digital Watermarking*. Springer pp. 454–468.

Goljan, Miroslav, Jessica Fridrich and Mo Chen. 2011. "Defending against fingerprint-copy attack in sensor-based camera identification." *IEEE Transactions on Information Forensics and Security* 6(1):227–236.

Goljan, Miroslav, Jessica Fridrich and Tomáš Filler. 2009. Large scale test of sensor fingerprint camera identification. In *Media Forensics and Security*. Vol. 7254 International Society for Optics and Photonics p. 72540I.

Handley, Lucy. 2018. "Four in 10 people have deleted a social media account in the past year due to privacy worries, study says." https://www.cnbc.com/2018/06/18/people-are-deleting-social-media-accounts-due-to-privacy-worries.html.

Hao, Tian, Ruogu Zhou and Guoliang Xing. 2012. COBRA: color barcode streaming for smartphone systems. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM pp. 85–98.

He, Kaiming, Jian Sun and Xiaoou Tang. 2013. "Guided image filtering." *IEEE transactions on pattern analysis & machine intelligence* (6):1397–1409.

Hupperich, Thomas, Davide Maiorca, Marc Kührer, Thorsten Holz and Giorgio Giacinto. 2015. On the robustness of mobile device fingerprinting: Can mobile users escape modern web-tracking mechanisms? In *Proceedings of the 31st Annual Computer Security Applications Conference*. ACM pp. 191–200.

Image Sensor Relative Size Comparison Tool. N.d. http://cameraimagesensor.com/size/.

Index, Breach Level. N.d. "DATA BREACH STATISTICS." http://breachlevelindex.com/.

Information capacity and versions of the QR Code. N.d. http://www.qrcode.com/en/about/version.html.

Jana, Suman and Sneha K Kasera. 2010. "On fast and accurate detection of unauthorized wireless access points using clock skews." *IEEE Transactions on Mobile Computing* 9(3):449–462.

Janesick, James R et al. 2001. *Scientific charge-coupled devices*. Vol. 117 SPIE press Bellingham.

Jiang, Zhiping, Jizhong Zhao, Xiang-Yang Li, Jinsong Han and Wei Xi. 2013. Rejecting the attack: Source authentication for wi-fi management frames using csi information. In *INFOCOM, 2013 Proceedings IEEE*. IEEE pp. 2544–2552.

Karaküçük, Ahmet and Ahmet Emir Dirik. 2015. "Adaptive photo-response non-uniformity noise removal against image source attribution." *Digital Investigation* 12:66–76.

Khanna, Nitin, Aravind K Mikkilineni and Edward J Delp. 2009. "Scanner identification using feature-based processing and analysis." *IEEE Transactions on Information Forensics and Security* 4(1):123–139.

Li, Chang-Tsun, Chih-Yuan Chang and Yue Li. 2009. On the repudiability of device identification and image integrity verification using sensor pattern noise. In *International Conference on Information Security and Digital Forensics*. Springer pp. 19–25.

Liu, Ming-Wei and John F Doherty. 2008. Specific emitter identification using nonlinear device estimation. In *2008 IEEE Sarnoff Symposium*. IEEE pp. 1–5.

Lukas, Jan, Jessica Fridrich and Miroslav Goljan. 2006. "Digital camera identification from sensor pattern noise." *IEEE Transactions on Information Forensics and Security* 1(2):205–214.

Mawale, Ashwini and Archana Chaugule. 2016. "Facial Wrinkles Detection Techniques and its Application." *International Journal of Computer Applications* 134(7):5–8.

Nakamura, Junichi. 2016. *Image sensors and signal processing for digital still cameras*. CRC press.

Nielsen, Jakob. 2012. "How Many Test Users in a Usability Study?" https://www.nngroup.com/articles/how-many-test-users/.

Nov, Oded, Mor Naaman and Chen Ye. 2009. Motivational, Structural and Tenure Factors that Impact Online Community Photo Sharing. In *ICWSM*.

Ohchi, Shuji, Shinichiro Sumi and Kaoru Arakawa. 2010. A nonlinear filter system for beautifying facial images with contrast enhancement. In *Communications and Information Technologies (ISCIT), 2010 International Symposium on*. IEEE pp. 13–17.

Omnicore. 2018*a*. "Facebook by the Numbers: Stats, Demographics & Fun Facts." https://www.omnicoreagency.com/facebook-statistics/.

Omnicore. 2018*b*. "Instagram by the Numbers: Stats, Demographics & Fun Facts." https://www.omnicoreagency.com/instagram-statistics/.

Patel, Hiren. 2015. Non-parametric feature generation for RF-fingerprinting on ZigBee devices. In *CISDA, 2015 IEEE Symposium on*. IEEE pp. 1–5.

Pathways, Forensic. 2010. "Digital Signature Identification." http://www.forensic-pathways.com/forensic-image-analyser/.

Pesce, João Paulo, Diego Las Casas, Gustavo Rauber and Virgílio Almeida. 2012. Privacy attacks in social media using photo tagging networks: a case study with Facebook. In *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*. ACM p. 4.

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

Polak, Adam C and Dennis L Goeckel. 2015. "Wireless device identification based on RF oscillator imperfections." *IEEE Transactions on Information Forensics and Security* 10(12):2492–2501.

Polak, Adam C, Sepideh Dolatshahi and Dennis L Goeckel. 2011. "Identifying wireless users via transmitter imperfections." *IEEE Journal on selected areas in communications* 29(7):1469–1479.

Quiring, Erwin and Matthias Kirchner. 2015. Fragile sensor fingerprint camera identification. In *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*. IEEE pp. 1–6.

Rao, Quanquan, Haodong Li, Weiqi Luo and Jiwu Huang. 2013. Anti-forensics of the triangle test by random fingerprint-copy attack. In *Computational Visual Media Conference*. pp. 1–6.

Rehman, Saeed Ur, Kevin W Sowerby and Colin Coghill. 2014. "Analysis of impersonation attacks on systems using RF fingerprinting and low-end receivers." *Journal of Computer and System Sciences* 80(3):591–601.

Remley, KA, Chriss A Grosvenor, Robert T Johnk, David R Novotny, Paul D Hale, MD McKinley, A Karygiannis and E Antonakakis. 2005. Electromagnetic signatures of WLAN cards and network security. In *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.* IEEE pp. 484–488.

Rosenfeld, Kurt and Husrev Taha Sencar. 2009. A study of the robustness of prnu-based camera identification. In *Media Forensics and Security*. Vol. 7254 International Society for Optics and Photonics p. 72540M.

Sharaf-Dabbagh, Yaman and Walid Saad. 2016. On the authentication of devices in the Internet of Things. In *2016 IEEE 17th International Symposium on WoWMoM*. IEEE pp. 1–3.

Son, Yunmok, Juhwan Noh, Jaeyeong Choi and Yongdae Kim. 2018. "GyrosFinger: Fingerprinting Drones for Location Tracking Based on the Outputs of MEMS Gyroscopes." *ACM Transactions on Privacy and Security (TOPS)* 21(2):10.

Squicciarini, Anna C, Heng Xu and Xiaolong Zhang. 2011. "CoPE: Enabling collaborative privacy management in online social networks." *Journal of the American Society for Information Science and Technology* 62(3):521–534.

Steinebach, Martin, Huajian Liu, Peishuai Fan and Stefan Katzenbeisser. 2010. Cell phone camera ballistics: attacks and countermeasures. In *Multimedia on Mobile Devices 2010*. Vol. 7542 International Society for Optics and Photonics p. 75420B.

Toonstra, J and W Kinsner. 1996. A radio transmitter fingerprinting system ODO-1. In *Proceedings of 1996 Canadian Conference on Electrical and Computer Engineering*. Vol. 1 IEEE pp. 60–63.

Valsesia, Diego, Giulio Coluccia, Tiziano Bianchi and Enrico Magli. 2017. "User Authentication via PRNU-Based Physical Unclonable Functions." *IEEE Transactions on Information Forensics and Security* 12(8):1941–1956.

Van Goethem, Tom, Wout Scheepers, Davy Preuveneers and Wouter Joosen. 2016. Accelerometer-based device fingerprinting for multi-factor mobile authentication. In *International Symposium on Engineering Secure Software and Systems*. Springer pp. 106–121.

Wang, Cong, Bingsheng Zhang, Kui Ren and Janet M Roveda. 2013. "Privacy-assured outsourcing of image reconstruction service in cloud." *IEEE Transactions on Emerging Topics in Computing* 1(1):166–177.

Wang, Na, Heng Xu and Jens Grossklags. 2011. Third-party apps on Facebook: privacy and the illusion of control. In *Proceedings of the 5th ACM symposium on computer human interaction for management of information technology*. ACM p. 4.

Wang, Wenhao, Zhi Sun, Kui Ren and Bocheng Zhu. 2016. User capacity of wireless physical-layer identification: An information-theoretic perspective. In *2016 IEEE International Conference on Communications (ICC)*. IEEE pp. 1–6.

Xiong, Jie and Kyle Jamieson. 2013. Securearray: Improving wifi security with fine-grained physical-layer information. In *MobiCom*. ACM pp. 441–452.

Xu, Li, Cewu Lu, Yi Xu and Jiaya Jia. 2011. Image smoothing via L 0 gradient minimization. In *ACM Transactions on Graphics (TOG)*. Vol. 30 ACM p. 174.

Yeung, Ching-man Au, Lalana Kagal, Nicholas Gibbins and Nigel Shadbolt. 2009. Providing Access Control to Online Photo Albums Based on Tags and Linked Data. In *AAAI Spring Symposium: Social Semantic Web: Where Web 2.0 Meets Web 3.0*. pp. 9–14.

Yu, Jun, Baopeng Zhang, Zhengzhong Kuang, Dan Lin and Jianping Fan. 2017. "iPrivacy: image privacy protection by identifying sensitive objects via deep multi-task learning." *IEEE Transactions on Information Forensics and Security* 12(5):1005–1016.

Zanetti, Davide, Boris Danev et al. 2010. Physical-layer identification of UHF RFID tags. In *MobiCom*. ACM pp. 353–364.

Zeng, Kai, Kannan Govindan, Daniel Wu and Prasant Mohapatra. 2011. Identity-based attack detection in mobile wireless networks. In *INFOCOM, 2011 Proceedings IEEE*. IEEE pp. 1880–1888.

Zeng, Kai, Kannan Govindan and Prasant Mohapatra. 2010. "Non-cryptographic authentication and identification in wireless networks [security and privacy in emerging wireless networks]." *IEEE Wireless Communications* 17(5).

Zerr, Sergej, Stefan Siersdorfer, Jonathon Hare and Elena Demidova. 2012. Privacy-aware image classification and search. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM pp. 35–44.

Zhang, Bingsheng, Kui Ren, Guoliang Xing, Xinwen Fu and Cong Wang. 2016. "SBVLC: Secure barcode-based visible light communication for smartphones." *IEEE Transactions on Mobile Computing* 15(2):432–446.

Zhang, Linghan, Sheng Tan and Jie Yang. 2017. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *CCS*. ACM pp. 57–71.

Zhou, Zhe, Wenrui Diao, Xiangyu Liu and Kehuan Zhang. 2014. Acoustic fingerprinting revisited: Generate stable device id stealthily with inaudible sound. In *CCS*. ACM pp. 429–440.

Zhuang, Zhou, Xiaoyu Ji, Taimin Zhang, Juchuan Zhang, Wenyuan Xu, Zhenhua Li and Yunhao Liu. 2018. FBSleuth: Fake Base Station Forensics via Radio Frequency Fingerprinting. In *ASIACCS*. ACM pp. 261–272.