Technical Disclosure Commons

Defensive Publications Series

December 2019

Safeguarding Biometric Authentication Systems from Fingerprint Spoof Attacks

Firas Sammoura

Jean-Marie Bussat

Safeguarding Biometric Authentication Systems from Fingerprint Spoof Attacks

Abstract:

This publication describes processes, methods, and techniques of safeguarding biometric authentication systems from fingerprint spoof attacks. A fingerprint spoof attack occurs when a malicious party attempts to access a user's computing device by mimicking/replicating biometric identifiers inherent to the user's fingerprint. To prevent fingerprint spoof attacks, an improved biometric authentication protocol includes, in a first step, verifying a user's identity and, in a second step, determining if the finger presented is alive. As a result, fingerprint spoof attacks can be rejected when it is determined that the spoof finger does not exhibit characteristics of life.

Keywords:

Fingerprint biometric recognition systems, fingerprint sensor, spoof detection, spoof fingerprint, fake fingerprint, biometrics, machine-learned model, ML, artificial intelligence, AI, authentication, fingerprint matcher, convolution neural network (CNN), binary statistical image features (BSIF), local binary pattern (LBP), local phase quantization (LPQ), feature extraction

Background:

Biometric recognition systems afford computing device users personalized and convenient means by which to access their device. Popular biometric identifiers include facial patterns, voice, and fingerprints. Fingerprints are the most common, and generally most convenient, means by which to authenticate oneself to a biometric recognition system. Unfortunately, biometric recognition systems are vulnerable to fingerprint spoof attacks. Fingerprint spoof attacks entail

the tricking of biometric sensors by utilizing a false finger (spoof finger) that mimics/replicates the biometric identifiers found in the authentic fingerprint. Consequently, a biometric recognition system may mistakenly permit access to the computing device if a spoof finger can trick a biometric recognition system sensor. Fingerprint spoof attacks, therefore, pose a serious security and privacy concern.

Therefore, it is desirable to improve biometric authentication protocols, such that spoof fingers cannot gain access to the computing device. To this end, determining the liveness of a spoof finger may thwart fingerprint spoof attacks.

Description:

This publication describes processes, methods, and techniques of safeguarding biometric authentication systems from fingerprint spoof attacks. More specifically, a computing device, such as a smartphone or tablet, can detect spoof fingers through implementation and execution of these processes, methods, and techniques.

When initializing or altering authentication security settings (enrollment) on a computing device, a user may be prompted to provide biometric information by placing a finger on a biometric recognition system sensor (*e.g.*, a fingerprint sensor); to the end that the fingerprint sensor can capture an image of the fingerprint and/or liveness features of the finger. When the fingerprint is captured during enrollment, it is referred to as a template image. Immediately after capturing a fingerprint and/or liveness features of the finger, a machine-learned model (fingerprint matcher) divides the template image into "M" number of smaller images, referred to as patches, with a sliding distance of one pixel.

As a user attempts to access the computing device, they may be prompted to provide biometric authentication. Biometric authentication includes, in a first step, verifying a user's identity and, in a second step, determining if the user is alive (liveness). In the same manner by which the user enrolled the device, they may be prompted to authenticate themself to the device by placing their finger on the fingerprint sensor.

In the first step, the fingerprint sensor captures a fingerprint and/or liveness features. When the fingerprint is captured during authentication, it is referred to as a verify image. Immediately after capturing the fingerprint and/or liveness features of the finger, the fingerprint matcher divides the verify image into "M" number of patches with a sliding distance of one pixel and compares the verify image patches to the template image patches. If a verify image patch matches a template image patch, then both patches are extracted.

Figure 1 illustrates an exemplary verify image and template image, along with an extracted matching patch.

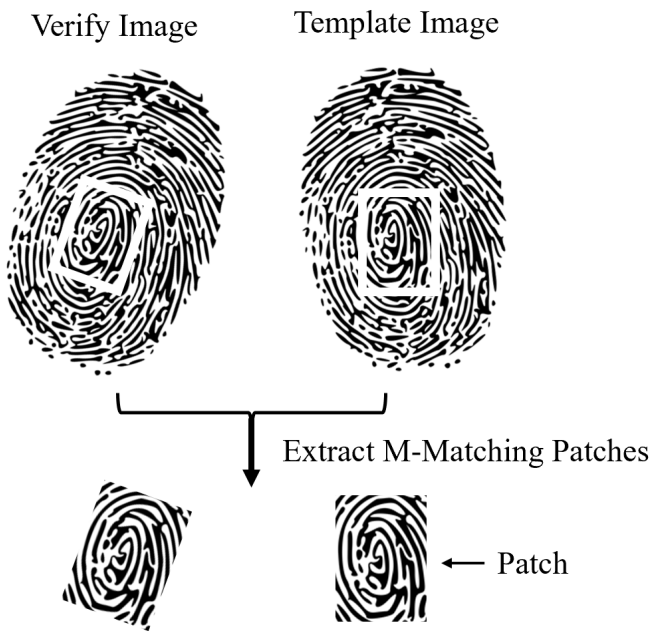

Figure 1

As illustrated in Figure 1, a fingerprint sensor captured images of a user's fingerprint. The fingerprint sensor captured the template image during enrollment, while the verify image was captured during authentication. Immediately after capturing either of the images and/or liveness features, a fingerprint matcher divides the image into "M" number of patches with a sliding distance of one pixel. During authentication, the fingerprint matcher analyzes the patches from both images and extracts the matching patches. For illustrative purposes, Figure 1 only presents one extracted matching patch.

The fingerprint matcher then generates a "Yes-or-No" outcome based on a predetermined threshold number of matching patches. In other words, the fingerprint matcher first verifies the user's identity. If the fingerprint matcher determines that the verify image contains biometric identifiers differing from the biometric identifiers contained in the template image, then the biometric authentication system can cease further analysis and reject authentication.

In a second step, the biometric recognition system determines liveness. Any combination of the following features utilized to determine liveness are referred to herein as liveness features.

Texture Feature Extraction

In one aspect, the described processes, methods, and techniques incorporate the utilization of texture feature extraction approaches, such as binarized statistical image features (BSIF), local binary pattern (LBP), local phase quantization (LPQ), invariant gradients, wavelet features, and/or touch features. These approaches are different methods by which to calculate the perceived texture of an image. In a first step, during enrollment and authentication, these approaches can extract texture features from the template image and verify image, respectively. Take for instance, a nefarious troublemaker who attempts to access a computing device by authenticating himself with

an image of the owner's fingerprint on a white piece of paper. A biometric recognition system that utilizes texture feature extraction approaches can compare and compute a concatenated difference vector of the verify image's texture and the template image's texture.

In a second step, a machine-learned model, such as a support vector machine (SVM) or a deep neural network (DNN), can then analyze the liveness feature difference vector to determine individual patch liveness scores. In a final step, the liveness scores can be fused to generate a total liveness score. In continuing the example, a machine-learned model on the computing device can analyze the vector and detect that the texture of the spoof finger is different than that of the template image. As a result, the biometric recognition system can reject authentication.

Convolution Neural Network Feature Extraction

In another aspect, the described processes, methods, and techniques incorporate the utilization of a convolution neural network (CNN) feature extraction approach to determine liveness. In a first step, the CNN compares liveness features of the verify image, patch-by-patch, to liveness features of the template image. The CNN may be iteratively trained, off-device, to compare and match two types of patch pairs: live-to-live and spoof-to-live. For example, a live-to-live matching patch pair includes both a verify image and a template image generated from the same live user's finger. Conversely, a spoof-to-live matching pair includes both a verify image and a template image generated from a spoof finger and a live user, respectively. Thus, the CNN can be trained to recognize live-to-live pairs and spoof-to-live pairs, as well as computing a liveness feature difference vector. After sufficient training, the CNN can be deployed to the CRM of a computing device.

In a second step, a machine-learned model, such as an SVM or an DNN, can then analyze the liveness feature difference vector to determine individual patch liveness scores. In a final step, the liveness scores can be fused to generate a total liveness score.

Dynamic Feature Extraction

In another aspect, the described processes, methods, and techniques incorporate the utilization of red, green, and blue light illuminated on a finger to determine liveness. Since optical properties of human tissue are amplified and/or detected through the illumination of various colors, a spoof finger can be distinguished from a live finger by illuminating the proposed finger with red, green, and/or blue light.

In a first step, during enrollment, a user may be prompted to situate their finger above the computing device's display, while the display sequentially emits blue, red, and green light. An imaging sensor (*e.g.*, a front-facing RGB camera) integrated into the computing device can then capture images of the finger for each successive color emitted. Lastly, liveness features are extracted from the captured image.

In a second step, during authentication, a user may be prompted to situate their finger above the computing device's display. The display can then emit a pattern of red, green, and blue light on the finger. While the pattern is illuminating the finger, the imaging sensor can capture an image of the finger.

Figure 2A and Figure 2B illustrate the differing illumination methods for enrollment and authentication.

Enrollment

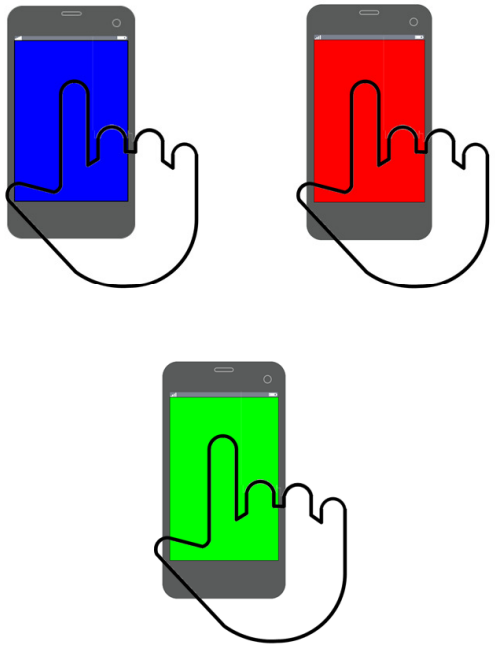

Figure 2A

Authentication

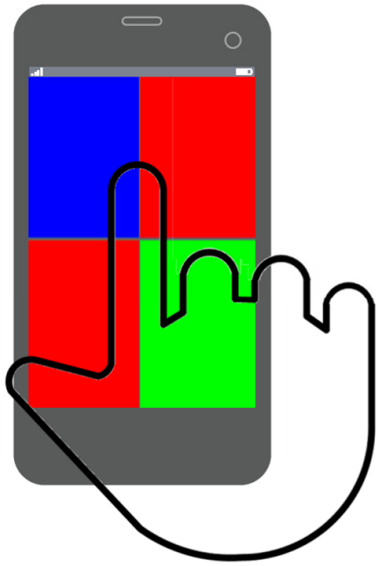

Figure 2B

Figure 2A illustrates conditions during enrollment. More specifically, a user's finger is situated above the device display while the display sequentially emits blue, red, and green light. For each color, an imaging sensor can capture images of the finger. Figure 2B illustrates conditions during authentication. More specifically, a user's finger is situated above the device display while the display presents a pattern of colors, such as blue, red, red, and green. The imaging sensor can capture a single image of the finger when illuminated by the pattern.

In a third step, liveness features can be extracted from the authenticated image and compared to the liveness features extracted from the enrollment image. The comparison results in a liveness feature difference vector. Next a machine-learned model, such as an SVM or an CNN, can analyze the vector to determine individual patch liveness scores. In a final step, the liveness scores can be fused to generate a total liveness score.

In an alternative third step, a machine-learned model, such as a generative adversarial network (GAN), can be trained with images of live, color illuminated fingers. As a result, the GAN can be trained to distinguish spoof fingers from live fingers by comparing the optical properties of the live finger in enrollment images to the live or spoof finger in the authentication image.

In all aspects, the improved biometric authentication protocol includes two main steps and at least two machine-learned models. Figure 3 illustrates all aspects of the improved biometric authentication protocol.

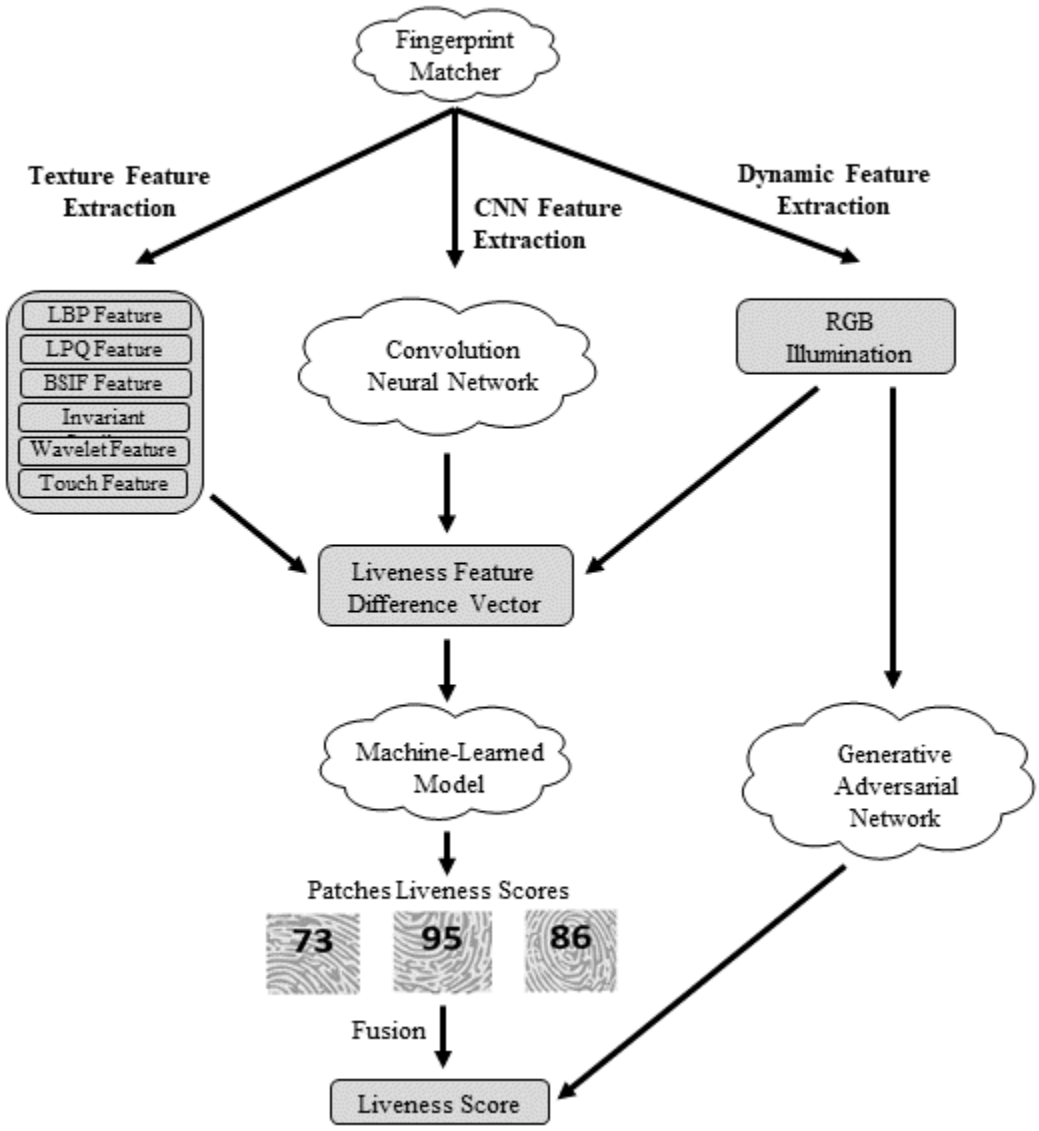

Figure 3

As illustrated, the first step of the improved biometric authentication protocol is to execute the fingerprint matcher. In short, the fingerprint matcher verifies the user identity. If the biometric recognition system verifies the user identity, then the second step is to determine the liveness of the finger. The biometric recognition system can determine liveness by any of the following techniques: texture feature extraction, CNN feature extraction, or dynamic feature extraction.

Through utilization of any, or all, of the feature extraction techniques, the biometric recognition system can calculate a total liveness score. If the liveness score exceeds a predetermined threshold, then the user can access the computing device; otherwise, if the liveness score is too low, then the biometric recognition system can reject the finger as a spoof finger.

The processes, methods, and techniques as described herein, cooperatively operating on a computing device to reject spoof fingers can be appreciated in the following example. During enrollment, a user (Jane) places her finger on a fingerprint sensor. The fingerprint sensor captures a template image and texture features (*i.e.*, liveness features). A fingerprint matcher generates an "M" number of template image patches. Later, Jane attempts to authenticate herself to the computing device by placing her finger on the fingerprint sensor. The fingerprint sensor captures a verify image and texture features. The fingerprint matcher generates an "M" number of verify image patches and confirms that the fingerprint contains biometric identifiers similar to Jane's finger. Next, the biometric authentication system compares, patch-by-patch, the texture features measured in the verify image to the template image. The comparison generates a difference vector that is analyzed by an SVM. The SVM scores each matching verify image patch for liveness. The individual scores are fused together and produce a liveness score above a predetermined threshold; thus, Jane can access her computing device.

In another example instance, during enrollment, a user (Jake) places his finger on a fingerprint sensor. The fingerprint sensor captures a template image. The computing device then prompts Jake to situate his finger above the computing device's display. Next, the display sequentially emits blue, red, and green light. For each emitted color, an imaging sensor captures images of Jake's finger. Lastly, liveness features are extracted from the enrollment images. Later, a nefarious troublemaker attempts to access Jake's computing device using a molded spoof finger.

The molded spoof finger may register as containing biometric identifiers similar to Jake's finger, but when observed under color illuminated conditions, the detected optical properties differ from a living finger. As a result, a CNN calculates individual liveness scores which, when fused together, generate a total liveness score that is too low to access the device and authentication is rejected.

In conclusion, the described processes, methods, and techniques can safeguard biometric authentication systems from fingerprint spoof attacks by, first, verifying a user's identity and, second, determining liveness.

References:

[1] Patent Publication: US 20180165508 A1. Systems and methods for performing fingerprint based user authentication using imagery captured using mobile devices. Priority Date: December 8, 2016.

[2] Patent Publication: US 20180018501 A1. Systems and methods for performing fingerprint based user authentication using imagery captured using mobile devices. Priority Date: February 6, 2015.

[3] Patent Publication: US 20180012006 A1. Method and apparatus for verifying user using multiple biometric verifiers. Priority Date: July 11, 2016.

[4] Patent Publication: US 20170200054 A1. Multi-stage liveness determination. Priority Date: September 5, 2014.

[5] Sammoura, Firas and Bussat, Jean-Marie, "Spoof Detection for Fingerprint Sensors," Technical Disclosure Commons. Published: November 05, 2019. https://www.tdcommons.org/dpubs_series/2648.

[6] Sammoura, Firas; Sengupta, Kuntal; and Bussat, Jean-Marie, "Fingerprint-Matching Algorithms," Technical Disclosure Commons. Published: May 29, 2019. https://www.tdcommons.org/dpubs_series/2228.

[7] L. Ghiani, A. Hadid, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection using Binarized Statistical Image Features," in the proceedings of the 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), Arlington, VA, USA, September 2013. https://ieeexplore.ieee.org/document/6712708.

[8] T. Ojala, M. Pietikäinen, and Topi Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 971-987, vol. 24, no. 7, July 2002. https://ieeexplore.ieee.org/document/1017623.