Critical Segment Based Real-time E-Signature for Securing Mobile Transactions

Yanzhi Ren[1], Chen Wang[1], Yingying Chen[1], Mooi Choo Chuah[2], Jie Yang[3]
[1]Department of ECE, Stevens Institute of Technology, Hoboken, NJ 07030
{yren2, cwang42, yingying.chen}@stevens.edu
[2]Department of CSE, Lehigh University, Bethlehem, PA 18015
chuah@cse.lehigh.edu
[3]Department of CS, Florida State University, Tallahassee, FL 32306
jyang5@fsu.edu

***Abstract*—The explosive usage of mobile devices enables conducting electronic transactions involving direct E-signature on such devices. Thus, user signature verification becomes critical to ensure the success deployment of online transactions such as approving legal documents and authenticating financial transactions. Existing approaches mainly focus on user verification targeting the unlocking of mobile devices or performing continuous verification based on a user's behavioral traits. Few studies provide efficient real-time user signature verification. In this work, we propose a critical segment based online signature verification system to secure mobile transactions on multi-touch mobile devices. Our system identifies and exploits the segments which remain invariant within a user's signature to capture the intrinsic signing behavior embedded in each user's signature. Our system further extracts useful features from a user's signature that describe both the geometric layout of the signature as well as physiological characteristics in the user's signing behavior. Our experimental evaluation of 25 subjects over six months time period shows that our system is highly accurate in provide signature verification and robust to signature forging attacks.**

I. Introduction

Mobile devices such as smartphones and tablets have become increasingly popular and play important roles in our daily lives. In particular, mobile devices equipped with touch screens and various communication interfaces have been used for supporting anytime-anywhere mobile services, e.g. e-commerce and online banking. For example, the "Go Mobile" trend has boosted the mobile transactions with 118% average growth every year in the United States, and the percentage increase is even higher in Asia-pacific region [1]. As e-commerce related applications become more prevalent, more and more sensitive information, such as financial data and credit card information are sent through transactions using mobile devices. Real-time user signature verification on mobile devices for online transactions becomes crucial for preventing unauthorized access, approving legal documents, or authenticating financial transactions.

Recent development focuses on providing secure methods to unlock mobile devices by utilizing a user's finger movement patterns or finger gestures captured on the touch screens [2]–[4] as opposed to traditional password based authentication. These methods are not suitable for securing mobile transactions. The online handwritten signature is one of the legally accepted mechanisms used to support real-time transactions. Previous work [5], [6] involves acquiring the user's signature from traditional digitizers with a stylus. As the multi-touch screen captures the user's signing behavior during the signature signing process, it could provide a richer set of information than the traditional digitizer. Sae-Bae Napa et.al [7] shows the initial success of verification on signatures directly signed by the user's finger on touch screens. This approach utilizes machine learning mechanisms to extract features from the whole signature trajectory but it does not consider signatures written by multiple fingers.

When using his/her finger to sign on touch screens, each individual has an intrinsic signature signing behavior that has not been studied in the previous work. Such a signing behavior, if captured, could largely increase the accuracy of signature verification and effectively combat adversarial signature forging activities. Toward this direction, we design a signature verification system that has the capability to identify the critical segments in each individual user's signature and extract unique features to describe a person's intrinsic signing behavior. Several challenges need to be addressed when developing such a system: First, the user's signature is signed on mobile devices with fingers instead of a stylus, resulting in extensive variations in the signatures. Second, the system should be resilient to signature imitation attacks as an adversarial can observe and mimic a legitimate user's signature. Third, the system should be robust to various signature sizes and orientations since the same user can write his signature with such variations under different scenarios.

Our proposed system consists of three main components to cope with these challenges: *Critical Segment Extraction*, *Feature Extraction* and *Signature Normalization and Interpolation*. Given a signature collected from the touch screen, *Signature Normalization and Interpolation* is performed to reduce the impact of signature geometric distortions caused by different writing sizes and orientations on touch screens. *Critical Segment Extraction* is the core component that captures the intrinsic user signing behaviors by identifying the segments which remain stable within the user's genuine signature. The extracted critical segments are typically invariant even in the presence of extensive variations in the user's signature, and hence hard for adversarials to imitate. During *Feature Extraction*, our system extracts useful features to describe both the geometric layout of the signature as well as the user's signing behavior by leveraging the rich set of information enabled by touch screens. In addition, our system supports signature signing using multiple fingers. We utilize two physiological features when signatures are signed with multiple fingers to improve accuracy of signature verification.

978-1-4673-7876-5/15/$31.00 ©2015 IEEE

We summarize our main contributions as follows:

- We design a signature verification system leveraging the multi-touch screen for mobile transactions.
- We propose to extract critical segments which capture a user's intrinsic signing behavior for accurate signature verification.
- Our system exploits not only the geometric layout of the user's signature but also his behavior and physiological characteristics, such as writing speed, touch pressure, distance pattern and correlation between fingers for attack-resilient signature verification.
- We show that our system is robust to adversarial behaviors of forging a user's signature including knowing how his signature looks like or even having the ability to observe and imitate his signature.
- We evaluate our system with 25 subjects over six-month time period. The results show that our system is highly accurate and robust in signature verification under various scenarios.

The rest of the paper is organized as follows. We first present the related work in Section II. We then describe the system model and attack model in Section III. The core component of critical segment identification is presented in Section IV. Next, we present feature extraction and signature normalization in Section V and Section VI respectively. In Section VII, we validate the feasibility and performance of our proposed system through real experiments. Finally, we conclude our work in Section VIII.

II. Related Work

Existing methods for user authentication on mobile devices can be categorized into two major classes: schemes for unlocking mobile devices and schemes for continuous user authentication. To unlock screens, some mobile devices rely on manual entry of a secret password or PIN number. This method is insufficient as many people only go through this process once when the device is switched on [8]. There has also been active work in using biometric information for user authentication on mobile devices such as fingerprints [9]. However, fingerprint scanners are not always available on smartphones, making it less suitable for user authentication on mobile devices. Several gesture based user authentication schemes have also been proposed [10], [11]. The basic idea of these schemes is users can be authenticated by making a gesture in the air while holding the mobile phone. Such gesture is captured through the accelerometer embedded in the phone for user authentication. However, the gesture based authentication is vulnerable to replay attacks in which attackers can observe and replicate the authentication information. Along this direction, recent work proposes to use users' finger gestures captured on touch screens for user authentication [2], [4], [12], [13]. These methods rely on the finger gesture patterns on the touch screens as authentication information. However, all of the above schemes consider unique features in simple gestures for either unlocking mobile devices or performing user authentication, and they cannot be easily applied in signature verification scenario.

Furthermore, there are schemes on continuous user authentication systems relying on finger movements [3] or gait patterns [14]. The purpose of such systems is to continuously authenticate users during the whole process of system execution. However, these behaviors did not happen frequently on mobile devices, because people hardly use mobile devices while walking or do heavy texting on them. Thus, it is difficult to collect a sufficient number of behavior traits for online transaction verification.

Our work focuses on the aspect of providing accurate online signature verification. Since signature is critical in many online transactions, several work have been proposed [5], [6], [15]. In these papers, a user utilizes a stylus to sign his signature on the touch screen of mobile devices. The information of the signature is then captured and compared with a pre-constructed template. Moving forward, new development enables the user directly sign his signature using finger on touch screens without the requirement of a stylus pen [7]. This approach is computationally efficient for verifying signatures. It represents the online signature with a feature vector derived from attribute histograms extracted from each signature. This system treats the user's signature as a whole and does not consider the intrinsic signing behavior of the user, which results in stable critical segments in signatures of the same user.

Our work is different in that we develop a secure and robust online signature verification system by capturing the intrinsic user signing behaviors through identifying the segments which remain invariant within the user's signatures. Moreover, our system verifies signatures not only based on the geometric layout of a signature but also based on the user's behavior and physiological characteristics. Additionally, signing signatures using multiple fingers (e.g., two fingers) are also considered to further improve the verification accuracy.

III. Framework Overview

In this section, we first introduce the challenges and design goals of our signature verification system. We then present the adversary model and provide an overview of our signature verification system.

A. Challenge and Design Goals

The goal of our system is to perform online signature verification on multi-touch mobile devices for securing mobile transactions. In particular, our system is designed to meet the following requirements:

Robust to Signature Variations. When a user signs his signature on touch screen with his finger instead of a stylus, the resulting signature could suffer from extensive variations under various signing conditions such as holding the device in-hand or putting the device on the table when signing. Our system aims to identify and extract the stable segments embedded in the signature to capture the user's intrinsic signing behavior for accurate signature verification.

High Accuracy. An adversary can observe and imitate a user's signature to pass online transaction verification. Our system should be robust under such adversarial behaviors by accepting the legitimate user's signature while rejecting forged signatures.

Adaptive to Various Writing Sizes or Orientations. Users may write their signatures with different sizes and orientations on the touch screen. Our system should be scalable to handle these scenarios for effective signature verification.

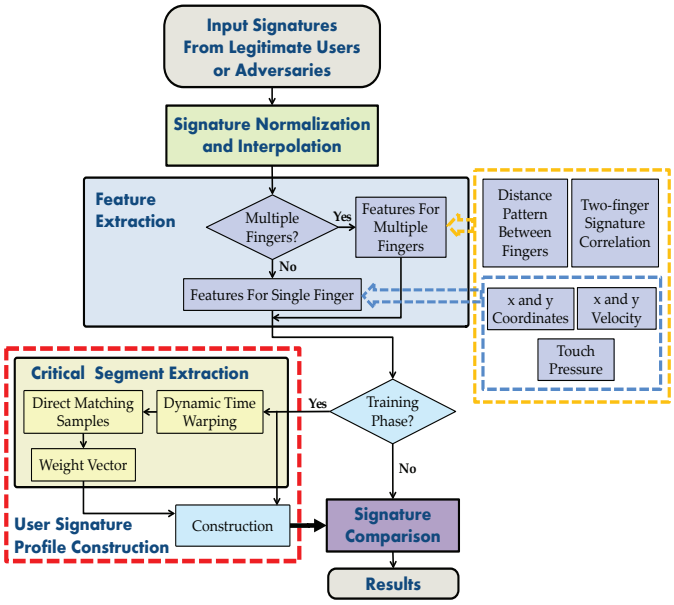


Fig. 1. Overview of our signature verification system

Computationally Feasible on Smartphones. Due to the limited computational resources at the mobile devices, the designed algorithms should be lightweight in order to perform signature verification in real time.

B. Adversary Model

An adversary may attempt to forge a legitimate user's signature patterns to complete certain online transactions to gain benefits. For example, adversaries can obtain some knowledge of the user's signature, and sign an unauthorized online financial transaction or change the terms of an online contract through signature fraud. We consider two possible adversarial behaviors as described below.

- **Random Attack:** The adversary only has the knowledge of the spelling of the legitimate user's signature without the knowledge of geometric layout of the signature.
- **Observe and Imitate Attack:** In this case, the adversary is more advanced by not only having the knowledge of the spelling of the legitimate user's signature but also obtaining chances to observe how the legitimate user signs his signature, i.e., he possesses the geometric layout of the legitimate user's signature when forging it.

C. System Overview

By leveraging a rich set of information provided by multi-touch screen, our system identifies the user's signing behavior and exploits the geometric layout of the signature and the user's behavior and physiological characteristics to achieve accurate and robust online signature verification. It identifies the critical segments in each individual user's signatures and extracts unique features to describe the user's intrinsic signing behavior. The features are selected such that they can describe the invariance and stability of a particular user's signature, which is hard for an adversary to imitate. In particular, we design two groups of features: the first group consists of the coordinates (including x and y dimensions) of a signature describing the geometric layout of the signature, and the second group consists of the velocity (on both x and y dimensions), touch pressure, distance pattern and correlation between two fingers, which capture the behavior and physiological characteristics of the user during the signature signing process. By comparing the extracted features of a signature to the pre-constructed user signature profile, our system makes decision on whether to accept or reject the input signature based on the similarity comparison.

As shown in Figure 1, our system consists of four major components: *Signature Normalization and Interpolation*, *Feature Extraction*, *Critical Segment Extraction* and *Signature Comparison*. Given a signature captured on the touch screen, the system first applies normalization and interpolation to deal with the signature geometric distortions caused by different writing sizes, orientations and locations on the touch screen under various signing conditions. In the *Feature Extraction* component, the system then extracts features from the normalized signatures to capture the geometric layout of the signature and the user's behavior and physiological characteristics. Critical segments extraction is used to capture the user's intrinsic signing behaviors and identify the feature segments that remain stable within the user's signatures. The extracted critical segments are usually invariant in the presence of variations in the user's signatures, and hard for attackers to imitate. After that, the signature verification is performed by calculating a similarity score between the extracted features from the input signature and the pre-constructed user profile which is constructed when the user enrolls in the verification system. Based on the similarity score between the testing signature and the user profile, our system makes decision on whether to accept or reject the user.

IV. Critical Segment Identification

A signature can be decomposed into several stroke segments. However, only a few of these decomposed segments are invariant across a set of signatures a user signs. Such segments reflect the user's intrinsic signing behavior, and we refer to them as *critical segments*. Existing work [7], [15] on signature verification did not consider capturing critical segments for signature verification. In our approach, we extract critical segments that reflect the user's intrinsic signing behavior to increase the accuracy of signature verification and combat adversarial signature forging activities.

To identify the critical segments in a user's signature, we develop an algorithm by examining and comparing the user's genuine signatures. Our algorithm takes a pair of signatures (which can be described by the features discussed in Section V) from the user as inputs and compare them using the dynamic time warping (DTW) [16] technique. Given a feature sequence used to represent the signature (e.g., x and y coordinates of the signature, the signing pressure of the signature), the resulting coupling sequence from DTW denotes an optimal alignment between two feature sequences. From [17], we know that the direct matching samples (DMSs) in the coupling sequence represent the segments without significant distortion between the two input signatures. Thus, the DMSs extracted from the coupling sequence can be utilized to derive a weight vector which denotes the similarity between two signatures. To capture the invariance of the signatures, we repeat the above comparison between each pair of the user's genuine signatures

in a signature pool, and then average over all the weight vectors to extract the critical segments as these segments have high similarity among a group of genuine signatures.

Critical Segment Identification Algorithm. To simplify the description of the critical segment extraction algorithm, we assume a signature is already normalized and interpolated to a length L and five features (i.e., f_1 to f_5 including the coordinates, the velocities and touch pressure) are used for describing the signature. The details of signature normalization and feature extraction are presented in Section VI and Section V respectively. We then assume $\{f_u^q, 1 \leq q \leq D\}$ to be a set of the u-th features ($1 \leq u \leq 5$) extracted from D genuine signatures. Thus, the weight vector of the u-th feature from the pair of q_1-th and q_2-th signatures can be represented as: $w_u^{q_1,q_2}$ with $1 \leq q_1, q_2 \leq D$ and $q_1 \neq q_2$. We then use the coupling sequence generated by the DTW algorithm to estimate the weight value. For each feature sequence (i.e. u-th feature), the dynamic warping procedure generates a coupling sequence CS^{q_1,q_2} with a length of K as: $CS^{q_1,q_2} = \{(f_u^{q_1}(i,j_k), f_u^{q_2}(i,j'_k)), 1 \leq k \leq K\}$, where $K \leq 2L$ and $1 \leq j_k, j'_k \leq L$. The i denotes that the feature is extracted from the i-th finger in case multiple figures are used for signing signatures. A direct matching sample (DMS) in the coupling sequence is defined as a feature sample in the q_1-th signature which has an one-to-one coupling with a sample in the q_2-th feature sequence. In other words, the matched sample $f_u^{q_1}(i,j_k)$ is a DMS if and only if both $f_u^{q_1}(i,j_k)$ and $f_u^{q_2}(i,j'_k)$ only appear once in the coupling sequence. The DMSs represent the signature region without significant distortion between two signature features. We thus define a weight sample $w_u^{q_1,q_2}(i,j)$ as:

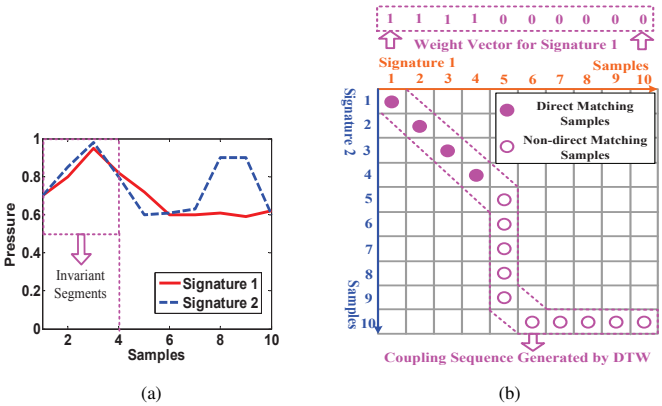
$$w_u^{q_1,q_2}(i,j) = \begin{cases} 1 & \text{if } f_u^{q_1}(i,j) \text{ is DMS in } CS^{q_1,q_2} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

To generalize the findings from each pair of signatures, our algorithm can examine a group of signatures from the same user and average the weight vectors over every pair of the signatures. Thus, the final weight vector of a particular feature for the user can be represented as:

$$\bar{w}_u(i,j) = \frac{\sum_{q_1=1}^{D} \sum_{\substack{q_2=1 \\ q_2 \neq q_1}}^{D} w_u^{q_1,q_2}(i,j)}{D \times (D-1)}, 1 \leq j \leq L \quad (2)$$

Each average weight value $\bar{w}_u(i,j)$ which ranges from 0 to 1 indicates the stability of the j-th sample of the u-th feature of that user's signatures. Intuitively, a larger value denotes better stability. Our algorithm treats the samples with higher average weights more significantly during the signature comparison procedure as they can represent the user's intrinsic signing behaviors. The segments with larger average weight values are identified as critical segments of the user's signature.

Example. Figure 2 shows an example on how a weight vector is extracted from a user's signatures by applying our critical segment identification algorithm. In this example, we collect two original signatures (i.e., signature 1 and 2) from the same user, and the touch pressure feature (i.e., f_5) is used for illustration. To simplify the description of the algorithm, we only use a portion of the pressure features (i.e., 10 samples). Figure 2 (a) plots the pressure features extracted from two signatures. From this figure, we observe that the first 4 samples are very similar, while the rest samples between two signatures differ significantly. We then generate the coupling sequence between two features based on DTW. In Figure 2 (b), the direct matching samples (DMSs) in the coupling sequence are shown as solid dots and the non-direct matching samples are shown as hollow dots. The weight vector is thus extracted according to the coupling sequence using Equation 1 and it is shown on the top of the figure. We find that the first four samples are assigned with a higher weight (i.e., 1) and other samples are assigned with a lower weight (i.e., 0). These four samples thus can be identified as one of the critical segments. This example shows that our critical segment identification algorithm can extract the invariant segments embedded in the signature for capturing the user's intrinsic signing behavior.



(a) (b)

Fig. 2. Illustration of critical segment identification: (a) Two genuine pressure feature collected from a user; (b) Weight vector extraction by utilizing direct matching samples (DMSs).

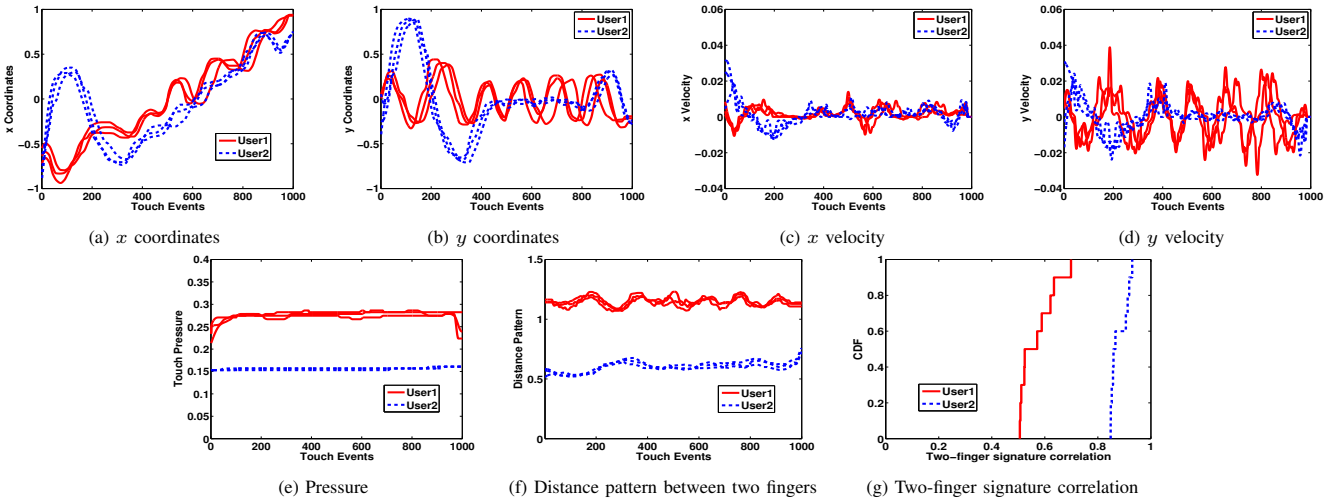
V. Feature Extraction and Signature Verification

In this section, we first present the important features we used in our signature verification system. We then describe how to construct user signature profiles using these features. Next, we discuss the metrics we used for signature verification.

A. Feature Extraction

The multi-touch screen provides rich set of information to describe a user's signature. In particular, when the user signs a signature with one or multiple fingers on the multi-touch screen, a sequence of touch events are generated and captured by the screen. Every touch event is characterized by the following information: a finger ID uniquely assigned to each finger, x and y coordinates of the touch point, pressure, and the time stamp of the event. We thus can use such information as the basis to describe the characteristic of signature that consists of a series of touch events.

Based on the information of each touch event, our signature verification system extracts several features to describe both the geometric layout of the signature as well as the user's signing behavior. Moreover, if the user signs the signatures using multiple fingers, we further examine two physiological features to enhance the performance of signature verification system. We assume the signature consists of L touch events



(a) x coordinates (b) y coordinates (c) x velocity (d) y velocity

(e) Pressure (f) Distance pattern between two fingers (g) Two-finger signature correlation

Fig. 3. An illustration of extracted features of user 1 and 2

after normalization and interpolation, and is performed by one or more fingers (the number of fingers is denoted as M). We will describe details of conducting signature normalization and interpolation in Section VI. The signature then can be represented as: $\{x''(i,j), y''(i,j), p''(i,j), 1 \leq i \leq M, 1 \leq j \leq L\}$. We next detail seven features we used in our verification system.

1) x coordinates and y coordinates: The x and y coordinates of the touch event sequence during signing process describe the geometric shape of the signature. The x and y coordinates of the extracted critical segments further represent the invariant geometric layout of the user's signatures. We thus use the x and y coordinates as the first (i.e., f_1) and second (i.e., f_2) feature to characterize the geometric shape of a signature.

$$\left\{ \begin{pmatrix} f_1(i,j) \\ f_2(i,j) \end{pmatrix} = \begin{pmatrix} x''(i,j) \\ y''(i,j) \end{pmatrix}, 1 \leq i \leq M, 1 \leq j \leq L \right\} \quad (3)$$

2) x velocity, y velocity and touch pressure: As the geometric shape of a signature can be easily observed and imitated by an adversary, we further propose three features to enhance the security of our system. These three features, the x and y velocities and the touch pressure, are used to capture the user's signing behavior during the signing process. These features are very hard, if not impossible, for adversity to observe and imitate. Moreover, each user usually has his/her own signing behavior during his/her signing process. Thus, these three features are good discriminators for accurate and attack-resilient signature verification. Specifically, the features of x velocity (i.e., f_3), y velocity (i.e., f_4) are represented as follows:

$$\left\{ \begin{pmatrix} f_3(i,j) \\ f_4(i,j) \end{pmatrix} = \begin{pmatrix} \frac{x''(i,j) - x''(i,j-1)}{\Delta t} \\ \frac{y''(i,j) - y''(i,j-1)}{\Delta t} \end{pmatrix}, 1 \leq i \leq M, 1 \leq j \leq L \right\} \quad (4)$$

The Δt denotes the time interval between two consecutive touch events. The touch pressure feature (i.e., f_5) is denoted as: $\{f_5(i,j) = p''(i,j), 1 \leq i \leq M, 1 \leq j \leq L\}$. The pressure value ranges from 0 to 1 with 1 and 0 representing the maximum pressure and no pressure at all respectively.

3) Distance pattern between fingers and two-finger signature correlation: The multi-touch screens on mobile devices allow users to sign their signatures using more than one finger and each finger will generate a sequence of touch events. In this work, we only consider two fingers (i.e., $M = 2$) as users barely use three or more fingers on touch screen simultaneously. Given more than one fingers are used, we further propose two additional features to capture the users' physiological uniqueness between two fingers. These features can be utilized to further enhance the performance of our signature verification system.

Specifically, we observe that the distance pattern between two fingers is often unique for each person due to the uniqueness of a person's hand/finger size. We thus propose the sixth feature as distance pattern between two fingers (i.e., f_6) and it is defined as:

$$\{f_6(j) = \sqrt{(x''(1,j) - x''(2,j))^2 + (y''(1,j) - y''(2,j))^2}, 1 \leq j \leq L\} \quad (5)$$

Moreover, we find that a user usually has a consistent correlation between the movements of two fingers during signing process, and different people may have different degrees of correlation between two fingers. We thus use this information as a complementary feature in our signature verification system. Specifically, the seventh feature, correlation between two fingers (i.e., f_7) is defined as follow:

$$f_7 = corr \left\{ \begin{pmatrix} \{x''(1,j)\}_{j=1}^{L} \\ \{y''(1,j)\}_{j=1}^{L} \end{pmatrix}, \begin{pmatrix} \{x''(2,j)\}_{j=1}^{L} \\ \{y''(2,j)\}_{j=1}^{L} \end{pmatrix} \right\} \quad (6)$$

where $corr$ denotes the Pearson correlation coefficient (PCC) [18], which measures the degree of the linear relationship between two given matrix.

4) Feasibility Study: We provide a feasibility study on how these features are similar for the same user and differ for different users by illustrating 20 signatures for each of two users with 10 single-finger based signatures and 10 multiple-finger based signatures. Figure 3 plots the seven features (i.e., f_1 to f_7) of these two users. Specifically, Figure 3 (a) to (f) present the feature patterns of f_1 to f_6 from three randomly

selected signatures of each user, respectively. Whereas Figure 3 (g) shows the cumulative distribution function (CDF) of the two finger signature correlation (i.e., f_7) values for each user's 10 multiple-finger based signatures. We observe that the feature patterns/values for the same user are very similar, while the features patterns/values for different users differ significantly. These observations show the feasibility of using these seven proposed features for signature verification.

B. User Signature Profile Construction and Signature Comparison

In this subsection, we describe how to construct user signature profile when a user enrolls in the system and how to compare the input signature with the pre-constructed user signature profile for verification.

1) Creation of User Signature Profile: The user signature profile creation phase starts when a user enrolls in the system. The user will be prompted to input D genuine signatures. The system first extracts features from these D genuine signatures by using the steps of Signature Normalization (discussed in Section VI) and Feature Extraction. The extracted features together with their associated weight vectors derived from Critical Segment Identification are used as the user's signature profile. Based on the number of fingers used in signing signatures, two different user signature profile creation procedures are performed.

Single-finger Based Signature. If a single finger is used to sign a signature, only the first five features (i.e., f_1 to f_5) are used. We first compute the average of each feature over D signature samples and use $\{\bar{f}_u, 1 \leq u \leq 5\}$ to denote the average feature values for the x and y coordinates, x and y velocities and pressure. We then extract the weight vectors for each feature as discussed in Section IV, and they are represented as $\{\bar{w}_u, 1 \leq u \leq 5\}$. The $\{\bar{f}_u, 1 \leq u \leq 5\}$ and $\{\bar{w}_u, 1 \leq u \leq 5\}$ are then used as the user signature profile.

Multiple-finger Based Signature. If multiple fingers are used to sign a signature, we add two additional features, namely the distance pattern between fingers (i.e., f_6) and the two-finger signature correlation (i.e., f_7). Similarly, we compute the average of each feature over D signatures and use $\bar{f}_6$ and $\bar{f}_7$ to denote them respectively. Note that there is no weight vectors for f_6 and f_7. This is because the distance pattern between fingers (i.e., f_6) is a physiological feature which remains relatively invariant among different signatures from the same user. Further, the two-finger signature correlation (i.e., f_7) is a single value instead of a feature vector. Finally, the $\bar{f}_6$ and $\bar{f}_7$ along with $\{\bar{f}_u, 1 \leq u \leq 5\}$ and $\{\bar{w}_u, 1 \leq u \leq 5\}$ are used for the user signature profile.

2) Distance Score Computation: We propose to use the weighted Manhattan distance to compare the input signature to a user's signature profile. Different computation procedures are also carried based on whether single-finger or multiple-finger are used to sign the signature.

Single-finger Based Signature. For single finger-based signatures, we define the distance score as the difference between the u-th feature obtained from the input signature $\{f_u, 1 \leq u \leq 5\}$ and the u-th average reference feature $\{\bar{f}_u, 1 \leq u \leq 5\}$ stored in the user signature profile. Specifically, the distance score is calculated by using the weighted Manhattan distance with the weights $\bar{w}_u$:

$$d_u = \sum_{i=1}^{M} \sum_{j=1}^{L} \left| f_u(i,j) - \bar{f}_u(i,j) \right| \bar{w}_u(i,j) \tag{7}$$

In practice, the distance values of different features may have different ranges. For example, the distance between velocity features usually has a range of $[0, 0.04]$ while the distance between coordinate features belongs to $[0, 1]$. We thus perform distance normalization on each d_u as: $d'_u = \frac{d_u - mean_u}{std_u}$, where $mean_u$ and std_u denote the mean and standard deviation of distance values of the u-th feature. Finally, the overall distance score is computed by averaging all the normalized distance values over 5 features: $\bar{d} = \frac{\sum_{u=1}^{5} d'_u}{5}$.

If $\bar{d}$ is less than a predefined threshold, the input signature is accepted by the system. Otherwise, the system rejects the signature.

Multiple-finger Based Signature. For multiple finger-based signatures, two additional features, the distance pattern between fingers (i.e., f_6) and the two-finger signature correlation (i.e., f_7) are included in computing distance scores using Manhattan distance as: $d_6 = \sum_{j=1}^{L} \left| f_6(j) - \bar{f}_6(j) \right|$ and $d_7 = \left| f_7 - \bar{f}_7 \right|$.

Similarly, we perform distance normalization on d_6 and d_7 to derive d'_6 and d'_7, respectively. The final distance score is then computed by averaging all the normalized distance scores over all 7 features: $\bar{d} = \sum_{u=1}^{7} d'_u/7$. Finally, if $\bar{d}$ is less than a predefined threshold, the input signature will be accepted by the system and vice versa.

VI. Signature Normalization and Interpolation

In this section, we present the signature normalization and interpolation algorithms. We assume a signature is performed with one or more fingers (i.e., the number of fingers is denoted as M), the raw x and y coordinates, and the pressure of a signature are represented as: $\{(x(i,j), y(i,j), p(i,j)), 1 \leq i \leq M, 1 \leq j \leq L_i\}$, where L_i denotes the number of touch events of the signature.

A. Signature Normalization

Users may write their signatures with various sizes, orientations, and regions on the touch screen under various signing conditions. We thus need to normalize users' signatures such that they have the same size, orientation and origin in the signature coordinate system to facilitate signature comparison. Existing work however did not study this problem systematically as the users use a stylus instead of fingers to sign their signatures in a small bounded rectangular region [5], [6], [15]. In our work, we perform systematical study on signature normalization via the sequential steps of translation, scaling and rotation.

1) Translation: Translation aims to move the geometric center of the signature to the origin of the signature coordination system such that all signatures can be at the same position for comparison. To achieve this, we define the geometric center of a signature as (x_c, y_c) where x_c is computed as follow:

$x_c = \sum_{i=1}^{M}\sum_{j=1}^{L_i} x(i,j) \Big/ \sum_{i=1}^{M} L_i$. Similar computation is done for y_c as well. The signature can be then translate to the origin of the coordinate system by subtracting x_c and y_c along vertical and horizontal directions, respectively.

2) Scaling: Scaling is to enlarge or shrink the signature by a scaling factor in all directions such that signatures can be of the same size. However, identifying the scaling factor is challenging as signatures are usually written with different orientations by users. For this reason, we propose to use a circle, which is a geometrical shape that is invariant to the writing orientations, as a reference to derive the scaling factor.

Specifically, we generate C concentric circles with different radius $\{R_c, 1 \leq c \leq C\}$ and place the center of the circles as the origin of the coordinate system. The value of R_c is chosen based on pre-defined ratios $\{P_c, 1 \leq c \leq C\}$ which is defined as the number of touch points within a circle over the total number of touch points. The scaling factor based on circle R_c thus can be calculated as $\alpha_c = P_c/R_c$, with the coordinates of the signature scale to a range of $[-1, 1]$. We then average the scaling factors derived from all circles and use the averaged value (i.e., α) as the scaling factor of the signature. Finally, we perform the scaling by multiplying each sample of a signature with the signature scale factor α. In this work, we empirically choose $C = 2$ circles with $P_1 = 0.5$ and $P_2 = 0.8$, respectively.

3) Rotation: Rotation aims to rotate signatures such that they are all parallel to the x-axis to facilitate the signature comparison. To identify the rotation angle θ, we examine the average distance between the positions of touch events and two reference points on the x-axis as a function of the rotation angle. The rationale behind this is that the average distance should be minimum when θ is equal to the writing angular of the signature (i.e., the signature becomes parallel to the x-axis).

In particular, we empirically select two points $(-0.5, 0)$ and $(0.5, 0)$ on x-axis as reference points. We then rotate the signature to locate the angle θ with which the average Euclidean distance between the signature's touch events and two reference points becomes minimum. We then rotate the whole signature counter-clockwise by the angle θ. After this, the signatures are normalized to the same origin, size and orientation in the signature coordinate system.

$$\begin{pmatrix} x'(i,j) \\ y'(i,j) \end{pmatrix} = \left\{ \alpha \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x(i,j) - x_c \\ x(i,j) - y_c \end{pmatrix} \right\}$$

where $1 \leq i \leq M, 0 \leq j \leq L_i$.

B. Signature Interpolation

The user usually writes signatures at different speeds which produce different numbers of touch events of signature. To deal with variable number of touch events, our system further performs signature interpolation. This step allows us to perform robust signature verification by directly measuring the similarity between two identical length of touch events.

To perform signature interpolation, we extend the sequence of signature touch events (i.e., x and y coordinates and pressure) to a reference length L by using cubic spline interpolation [19]. Further, we choose a large L (e.g., $L = 1000$ samples) so that it is large enough to include the length of touch events from any user's signature under normal signing process. The sequence of the x coordinates after interpolation are represented as (the same as y coordinates): $\{x''(i,1), ..., x''(i,L)\} = \text{interp}\left(\{x'(i,1), ..., x'(i,L_i)\}\right), 1 \leq i \leq M$. Similarly, the sequence of signature pressure is represented as: $\{p''(i,1), ..., p''(i,L)\} = \text{interp}\left(\{p(i,1), ..., p(i,L_i)\}\right), 1 \leq i \leq M$.

VII. Performance Evaluation

In this section, we evaluate the performance of our user signature verification system with more than twenty users on Google Nexus smartphones over a time period of six months.

A. Experimental Methodology

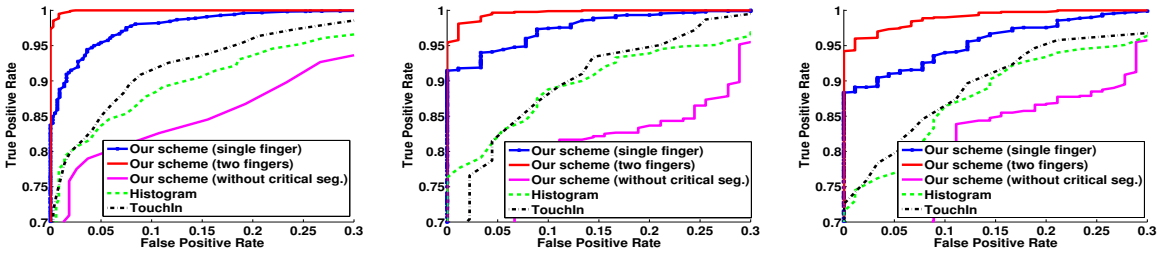
We use two Google Nexus smartphones equipped with multiple touch screens to collect signatures from 25 volunteer users over six months. Each Google Nexus smartphone is equipped with a 1.5 GHz quad-core Krait processor and runs Android operation system. We then develop an Android application to collect signatures from users. Specifically, the touch events triggered by each finger are written into a log file on the smartphone when the users sign their signatures on touch screen. During the experiments, we set the sampling rate as 50 Hz and the duration between two consecutive touch points is 20 ms. We conduct experiments in representative phone placements including holding the smartphone in a user's hand or placing the phone on a table. Our experiments are also conducted in different locations such as offices and apartments.

1) Evaluation Scenarios: We conduct experiments with 25 volunteers (ranging from 17 to 34 years old) to evaluate the effectiveness of our signature verification system. A size of 25 volunteers is a typical size for verification studies [2]. We evaluate our system under three scenarios including one normal user signature verification scenario and two representative attack scenarios.

Signature Verification: Each user conducts normal signature verification without having the knowledge of others' signatures. We collect a set of genuine signatures from each user to construct his/her signature profile, and use the rest of signatures to test the signature verification. In this scenario, users are told to sign their signatures using their own signatures and signing styles during signature signing process.

Random Attack: The attacker only has the knowledge of the spelling of a victim's signature without the knowledge of the geometric layout of the signature. More specifically, we choose 8 users as victims. We then select another 15 users as attackers whose handwriting fonts and speeds are similar to the selected victims to launch the random attack. In this scenario, attackers sign signatures with the same spelling of the victim's signature during the signing process.

Observe and Imitate Attack: The attacker not only has the knowledge of the spelling of the victim's signature but also observes how a legitimate user signs his signature. Similarly, we choose 8 users as victims and select another 15 users as attackers. In this scenario, attackers possess the geometric layout of the victim's signatures and then perform imitation attacks.



(a) Signature verification (b) Random attack (c) Observe and imitate attack

Fig. 4. Comparison with the existing schemes under signature verification scenario and two representative attack scenarios.

2) Metrics: We use the following metrics to evaluate the effectiveness of our signature verification system.

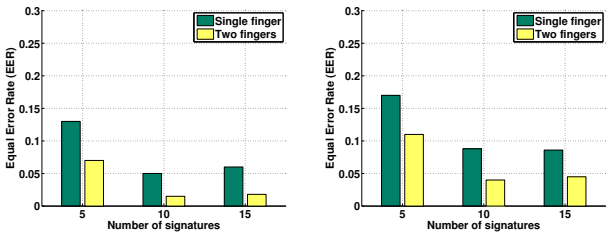
Receiver Operating Curve (ROC): We use ROC to evaluate how the true positive rate and false positive rate of our system changes when the threshold is varied. Specifically, we let tp, tn, fp and fn denote the total number of true positives, true negatives, false positives and false negatives. The true positive rate (TPR) and false positive rate (FPR) are defined as $TPR = tp/(tp + fn)$ and $FPR = fp/(fp + tn)$, respectively.

Equal Error Rate (EER): It is defined as the rate at which the FPR is equal to one minus TPR (i.e., the location on a ROC curve where $FPR = 1 - TPR$). The EER shows the trade-off between two error types and lower EER represents better performance.

B. Performance Comparison with Existing Schemes

In the first set of experiments, we evaluate the effectiveness of our proposed signature verification system in Figure 4 by comparing it with two existing signature verification schemes: a histogram based signature verification system [7] and TouchIn [2]. In Figure 4, we use "*Our scheme (single finger)*" and "*Our scheme (two fingers)*" to denote the results from our system using one and two fingers, respectively. The histogram based signature verification system [7] captures the distribution of attributes such as the first or second order derivative of coordinates derived from the signature for verification, and we utilize the legend "*Histogram*" to denote it in the results. The TouchIn authentication system [2] first asks a user to draw curves on the touchscreen, and then verify the user based on the properties such as the curvature and acceleration of the drawing curves. To compare the performance of our system to Touchin, we apply Touchin to handle signatures instead of curves as originally proposed. And the results are denoted as "*TouchIn*". Additionally, we evaluate our system performance when only using extracted features without considering the key technique, i.e., critical segments, for signature verification. This scenario is similar to the existing stylus-based online signature verification schemes [5], [6], which treat the user's signature as a whole. We use the legend "*Our scheme (without critical seg.)*" to denote the results from such scenarios.

Figure 4 (a) to (c) depict the ROC of signature verification under three scenarios of our proposed system and these two existing methods. We observe that our system can achieve a significant higher true positive rate with lower false positive rate than that of existing methods. In particular, our system achieves nearly 100% true positive rate for two-finger and over 95% true positive rate for one-finger when the false positive rate is 5%. However, the true positive rate of the existing methods is less than 85% when the false positive rate is 5%. Further, when comparing these results with those we obtained without using critical segments, we observe that including critical segments yields significantly better results. This is because our system captures the intrinsic signing behavior of the user by identifying the critical segments and further utilizes a rich set of information provided by multi-touch screen. Moreover, we find that two-finger based signatures can give us higher accuracy due to these two physiological features are used in the two-finger case.



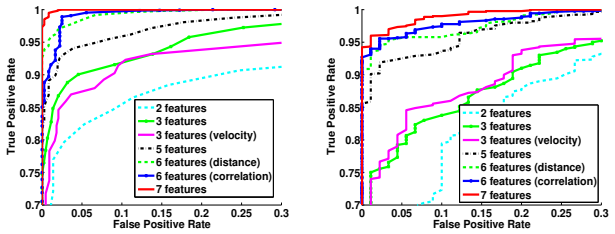
(a) Signature verification (b) Observe and imitate attack

Fig. 5. Robustness study under different number of signatures for user signature profile construction.

From Figure 4 (b) and (c), we further observe that our system is much more robust than existing methods under both *random attack* and *observe and imitate attack*. Specifically, our system achieves more than 95% true positive rate with less than 10% false positive rate, while existing methods only have about 85% true positive rate with 10% false positive rate. This is because our system captures the user's signing behavior during the signing process which is hard for a adversary to observe and imitate. These results indicate that our proposed system has significant better performance and is also more robust under attacks than existing schemes [2], [7].

C. Impact of User Signature Profile Construction

We study the robustness of our system when different number of signatures (i.e., D) are used in the user signature profile construction. Specifically, we evaluate the performance of our signature verification system when the size D equals to 5, 10 and 20, respectively. Figure 5 (a) and (b) show the Equal Error Rate (EER) under the scenarios of regular *signature verification* and *observe and imitate attack* under different numbers of signature used for signature profile construction. We observe that the EER decreases as the number of training signatures increases due to using a larger D can capture the user's



(a) Signature verification (b) Observe and imitate attack

Fig. 6. Robustness study under different number of applied features.

Legend	Feature
2 features	x, y coordinates
3 features	x, y coordinates + pressure
3 features(velocity)	x, y velocity + pressure
5 features	x, y coordinates + x, y velocity + pressure
6 features(distance)	x, y coordinates + x, y velocity + pressure + distance pattern between fingers
6 features(correlation)	x, y coordinates + x, y velocity + pressure + two-finger signature correlation
7 features	all features

TABLE I
IMPACT OF FEATURE NUMBER STUDY: FEATURES USED IN EACH SCENARIO

invariant signature pattern better. In particular, our system has around 5% EER for single-finger and less than 2% EER for two-finger when D is equal to or larger than 10. Moreover, even if under the *observe and imitate attack*, our system has less than 5% EER with two-finger when D is 10 or 20. The results indicate that a number of 10 signatures is sufficient for our system to achieve a low EER under both *signature verification* and *observe and imitate attack* scenarios.

D. Impact of Features

We further evaluate our system by utilizing different number of features for signature verification. Specifically, we evaluate the system performance by varying the number of features used in the system from 2 to 7. The denotations of the legends are shown in the Table I. Figure 6 (a) and (b) plot the ROC under the scenarios of regular *signature verification* and *observe and imitate attack* with varying number of features. We observe that the accuracy increases as the number of features increases. This is because with more features, our system can capture the user's intrinsic behaviors within the signature better. In addition, we found that five or more features result in acceptable performance with over 90% true positive rate and less than 5% false positive rate in all these three scenarios. Furthermore, similar true positive rate and false positive rate are achieved even if the attacker performs imitation attacks. This demonstrates that our system is robust against malicious attacks. Finally, we observe that our system is also robust to different phone holding positions such as having a user holding the smartphone in his hand or placing it on a table. Due to space limitation, we omit these results and include them in our extended report.

VIII. Conclusion

In this paper, we present a critical segment based online signature verification system to secure mobile transactions on mobile devices. The proposed system identifies the critical segments, which remain invariant within a user's signature, to capture the user's intrinsic signing behavior. By leveraging the rich set of information enabled by touch screens, our system extracts useful features to describe both the geometric layout of the signature as well as a user's behavioral and physiological characteristics during the signing process. Moreover, our signature normalization and interpolation methods enable robust signature verification in the presence of signature geometric distortions caused by different writing sizes, orientations and locations on touch screens. Extensive experiments with 25 users over six months time period show that our proposed system can achieve better performance than existing methods and is robust to signature forging attacks including random attack and observe and imitate attack.

Acknowledgement: This work is supported in part by National Science Foundation Grants CNS 0954020, SES 1450091, CNS 1016296, CNS 1217379 and ARO Grants ARO W911NF-13-1-0288.

References

[1] J. Hopkins and J. Turner, *Go Mobile: Location-Based Marketing, Apps, Mobile Optimized Ad Campaigns, 2D Codes and Other Mobile Strategies to Grow Your Business*. John Wiley & Sons, Inc., 2012.
[2] J. Sun, R. Zhang, J. Zhang, and Y. Zhang, "Touchin: Sightless two-factor authentication on multi-touch mobile devices," in *Proceedings of CNS*, 2014.
[3] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones," in *Proceedings of NDSS*, 2013.
[4] M. Shahzad, A. X. Liu, and A. Samuel, "Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it," in *Proceedings of ACM MobiCom*, 2013.
[5] C. Gruber, T. Gruber, S. Krinninger, and B. Sick, "Online signature verification with support vector machines based on lcss kernel functions," *IEEE Transactions on Systems, Man, and Cybernetics*, 2010.
[6] D. Impedovo and G. Pirlo, "Automatic signature verification: The state of the art," *IEEE Transactions on Systems, Man, and Cybernetics*, 2008.
[7] N. Sae-Bae and N. Memon, "Online signature verification on mobile devices," *IEEE Transactions on Information Forensics and Security*, 2014.
[8] N. L. Clarke and S. Furnell, "Authentication of users on mobile telephones - a survey of attitudes and practices," *Computers and Security*, 2005.
[9] T. Clancy, N. Kiyavash, and D. Lin., "Secure smartcard-based fingerprint authentication," in *Proceedings of the ACM SIGMM workshop on Biometrics methods and applications*, 2003.
[10] G. Bailador, C. Sanchez-Avila, J. Guerra-Casanova, and A. de Santos Sierra, "Analysis of pattern recognition techniques for in-air signature biometrics," *Pattern Recognition*, 2011.
[11] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "User evaluation of lightweight user authentication with a single tri-axis accelerometer," in *Proceedings of MobileHCI*, 2009.
[12] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, "Biometric-rich gestures: A novel approach to authentication on multi-touch devices," in *Proceedings of ACM SIGCHI*, 2012.
[13] M. Beton, V. Marie, and C. Rosenberger, "Biometric secret path for mobile user authentication: A preliminary study," in *Proceedings of WCIT*, 2003.
[14] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "Smartphone based user verification leveraging gait recognition for mobile healthcare systems," in *Proceedings of IEEE SECON*, 2013.
[15] D. Muramatsu and T. Matsumoto, "Effectiveness of pen pressure, azimuth, and altitude features for online signature verification," *Advances in Biometrics*, 2007.
[16] M. Meinard, *Information Retrieval for Music and Motion*. Springer-Verlag Berlin Heidelberg, 2007.
[17] G.Dimauro, S.Impedovo, and G.Pirlo, "A new methodology for the measurement of local stability in dynamical signatures," in *Proceedings of IWFHR*, 1994.
[18] G. Casella, R. Berger, and R. Berger, *Statistical inference*. Duxbury Press Belmont, Calif, 1990.
[19] R. V. Dukkipati, *Numerical Methods*. New Age International Pvt Ltd Publishers, 2010.