SEGAUTH: A Segment-based Approach to Behavioral Biometric Authentication

Yanyan Li*, Mengjun Xie*, Jiang Bian†
*University of Arkansas at Little Rock, Email: {yxli5, mxxie}@ualr.edu
†University of Florida, Email: bianjiang@ufl.edu

***Abstract*—Many studies have been conducted to apply behavioral biometric authentication on/with mobile devices and they have shown promising results. However, the concern about the verification accuracy of behavioral biometrics is still common given the dynamic nature of behavioral biometrics. In this paper, we address the accuracy concern from a new perspective—behavior segments, that is, segments of a gesture instead of the whole gesture as the basic building block for behavioral biometric authentication. With this unique perspective, we propose a new behavioral biometric authentication method called SEGAUTH, which can be applied to various gesture or motion based authentication scenarios. SEGAUTH can achieve high accuracy by focusing on each user's distinctive gesture segments that frequently appear across his or her gestures. In SEGAUTH, a time series derived from a gesture/motion is first partitioned into segments and then transformed into a set of string tokens in which the tokens representing distinctive, repetitive segments are associated with higher genuine probabilities than those tokens that are common across users. An overall genuine score calculated from all the tokens derived from a gesture is used to determine the user's authenticity. We have assessed the effectiveness of SEGAUTH using 4 different datasets. Our experimental results demonstrate that SEGAUTH can achieve higher accuracy consistently than existing popular methods on the evaluation datasets.**

I. INTRODUCTION

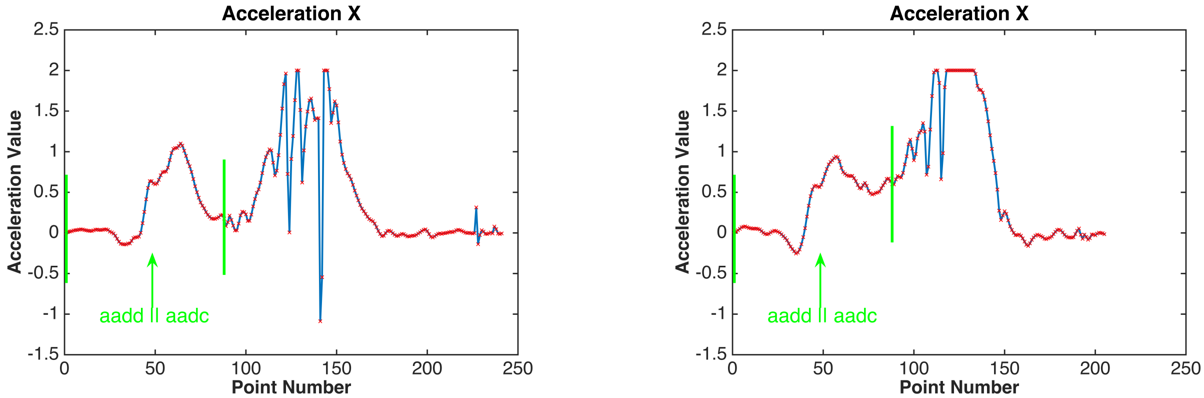
Biometrics becomes increasingly popular in user authentication, especially on mobile devices. For example, many smartphones of latest models have been equipped with a fingerprint scanner so that users can use their fingerprint instead of a password or personal identification number (PIN) to unlock the screen. For biometric authentication a user can be verified based on either his or her distinct physiological traits (e.g., fingerprint, iris) or behavioral patterns (e.g., gait, hand gesture). Compared to physiological biometrics, behavioral biometrics, which is embedded in humans' behavior, is more difficult to steal or mimic and therefore causes less privacy concerns. Among growing interests in applying behavioral biometrics to security and privacy protection, gesture or motion based behavioral biometric authentication has attracted a great deal of research efforts (e.g., [1], [4], [18], [21], [24], [22], [19], [32], [30], [3]).

Although many research studies have shown that gesture-based behavioral biometrics is very promising for mobile authentication, a common concern about the accuracy of behavioral biometric authentication still exists. Unlike physiological biometrics, behavioral biometrics is inherently dynamic, which poses a big challenge to verification methods. Using gesture-based behavioral biometric authentication as an example, due to the dynamic nature in performing gestures, a testing sample of the same gesture by the same person can vary nontrivially in part from that person's prior training samples. Therefore, the testing sample can be rejected by the authentication system if the system assumes a user can repeat the gesture in exactly the same manner. On the other hand, a usable authentication system should also be resistant to a variety of attacks especially mimicry attacks while accepting legitimate users.

Figures 1 (for the user with ID 13) and 2 (for the user with ID 25) exemplify such a case. Both figures depict the x-axis acceleration time series obtained from the user's two samples of the same drawing-a-circle gesture performed at different times [30]. On one hand, the two time series from the same user appear to be different more or less. For example, the curve in Figure 1 (a) exhibits strong oscillation in the middle while the curve in Figure 1 (b) does not; The curve in Figure 2 (a) is apparently longer than the curve in Figure 2 (b) besides other differences. On the other hand, a certain level of similarity manifests between the time series in Figure 2 (a) and the one in Figure 1 (b). In fact, if dynamic time warping (DTW) is applied to measure the similarity (or difference) of two time series, the DTW distance between the two time series in Figures 1 (b) and 2 (a) (denoted as D_S) is actually smaller (i.e., more similar) than the DTW distance between the two time series in Figures 1 (a) and 1 (b) (denoted as D_L). As a toy example, suppose the time series in Figure 1 (b) is adopted as the template for user 13 and DTW is employed as the verification method. The time series in Figure 1 (a) would be falsely rejected if the similarity threshold is smaller than D_L and/or the time series in Figure 2 (a) would be falsely accepted if the similarity threshold is greater than D_S. It appears unavoidable to have either a false rejection, a false acceptance, or both.

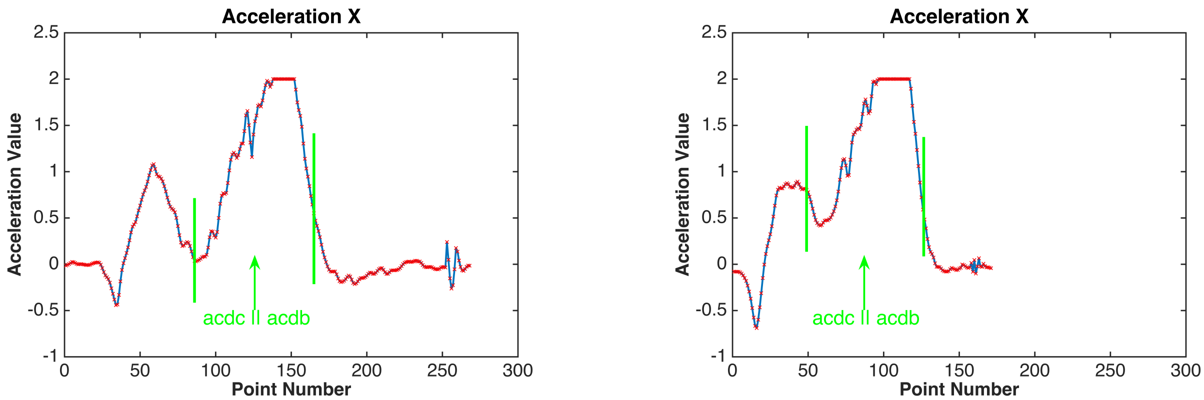
If we change our perspective by treating the entire time series as a sequence of segments, each of which is a smaller time series, we can find that the two time series from the same user usually have highly similar segments, e.g., the segments between the two green vertical lines in (a) and (b) of Figure 1 (and 2). By applying a symbolic representation technique, i.e., Symbolic Aggregate Approximation (SAX) [12], we can transform a time series into a string to facilitate pattern discovery and comparison. The segments between the vertical lines in Figure 1 are converted to the same string $aadd \mid\mid aadc$ (formed by two shorter strings each being transformed from a segment; $\mid\mid$ is the symbol for string concatenation). Similarly, the segments between the vertical lines in Figure 2 are mapped to the same

978-1-5090-3065-1/16/$31.00 ©2016 IEEE



(a) (b)

Fig. 1: User ID 13's x-axis acceleration time series for Circle gesture from 2 different trials



(a) (b)

Fig. 2: User ID 25's x-axis acceleration time series for Circle gesture from 2 different trials

string $acdc \,||\, acdb$.

Intuitively, for the same gesture, the trials by the same user in general are more likely to have segments that are alike than the trials by different users. By focusing on segments, more fine-grained and richer information can be considered in the verification method, which we believe is helpful in achieving high accuracy. And verification decisions (either binary or probabilistic) are easier to apprehend. Our key idea for improving authentication accuracy is to develop verification methods based on segments of behavioral biometric time series, which is very different from previous studies that only take the whole time series into consideration.

In this paper, we present a new behavioral biometric authentication method called SEGAUTH, which can be applied to various gesture or motion based authentication scenarios. SEGAUTH aims to achieve high accuracy by focusing on each user's distinctive behavior segments that appear frequently across gestures to reduce the impact of dynamic nature in performing a gesture on verification. In SEGAUTH, a time series acquired from a gesture is first partitioned into segments via sliding window and then transformed into a set of string tokens in which the tokens representing distinctive, repetitive segments are associated with higher genuine probabilities than those tokens that are common across users. Collectively, an overall genuine score can be calculated from all the tokens derived from a gesture and used to determine the user verification outcome. We have assessed the effectiveness of SEGAUTH using 4 different datasets and measured the accuracy of SEGAUTH using metrics accuracy (ACC) and equal error rate (EER). Our experimental results demonstrate that SEGAUTH can achieve higher accuracy consistently than existing popular methods on the evaluation datasets.

The rest of this paper is organized as follows: We present the design of SEGAUTH in Section II and detail its evaluation in Section III. We then discuss the factors affecting SEGAUTH in Section IV. We briefly describe the related work in Section V and conclude this paper in Section VI.

II. SYSTEM DESIGN

SEGAUTH is designed as a general approach to behavioral biometric authentication especially gesture based authentication. SEGAUTH leverages the sensing data generated by a user's motion intended for authentication to verify the user's identity. Similar to other user authentication schemes, SEGAUTH first builds a model for an enrolled user in the training phase and later applies the model in the verification phase. Figure 3 depicts the general process and basic modules of SEGAUTH. The illustration of Segmentation & Tokenization (S&T) is adapted from [12]. The data acquisition module collects sensory data through smart devices during authentication and outputs a

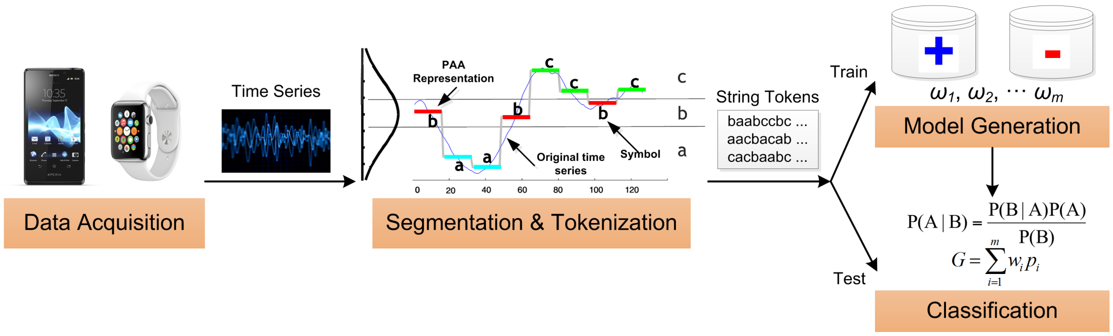


Fig. 3: Overview of SEGAUTH

set of features $\{f_1, f_2, ..., f_m\}$ for each sample (either a training one or a testing one), e.g., x, y coordinates of a finger tip on a touch screen and their derived velocities. Each feature of a sample is represented as a time series. In S&T phase, each time series is segmented and transformed into its symbolic representation, that is, a set of string tokens, which is detailed in Section II-B. The set of n tokens for feature f_i in a sample is denoted as $\{X_1^i, X_2^i, ..., X_n^i\}$.

In model generation, string tokens from training samples are stored in their respective databases. The tokens generated by genuine samples are stored in the positive database while the tokens generated by forgery samples are stored in the negative database. With the two databases, the probability of a token (i.e., a feature segment) being genuine (or forgery) can be derived from given training samples. Considering that features often differ in their capability of differentiating a genuine user from imposters, each feature f_i is assigned a weight w_i reflecting its distinguishability, which is derived based on the verification equal error rates (EER) using single features.

The classification of SEGAUTH is straightforward. Let the weights for features $f_1, ..., f_m$ be $w_1, ..., w_m$. Given a testing sample, SEGAUTH first applies the same S&T procedure to generate tokens, then applies the Bayes' theorem to compute the probability of the sample being genuine (p_i) based on each feature (f_i), and finally computes the sample's genuine score G using the formula $G = \sum_{i=1}^{m} w_i \cdot p_i$. The testing sample will be accepted as authenticated if G is equal to or greater than the predefined threshold.

A. Data Acquisition

The data acquisition module is responsible for collecting sensory data from motion sensors (e.g., touch sensor, accelerometer, etc) in smart mobile devices during gesture performance and exporting the time series of the features derived from sensory data. Nowadays motion sensors are ubiquitous on mobile devices and collecting motion data is well supported by major mobile platforms. Given the diversity of mobile devices, different types of sensory data may be collected. For example, finger coordinates, pressure, and size can usually be gathered on touch screen smartphones and tablets but they are not available on wristbands.

The feature set of an authentication gesture usually contains two parts: raw features directly from sensor readings (e.g., acceleration in $x/y/z$-axis) and derived features computed from raw features (e.g., the first derivative of acceleration in $x/y/z$-axis). Given the diversity of gestures for user verification and availability of sensors, the set of features for behavioral biometric authentication can vary in different authentication systems. For example, TouchIn [24] employs x, y coordinates, velocities, and accelerations, touch pressure, curvature, direction, and hand geometry while MotionAuth [30] uses 3-axis accelerations and angular accelerations, magnitude, angles, and their first and second derivatives. Selection of features for optimal authentication performance is dependent on many factors including devices, sensors, and behavioral biometrics and therefore is out of the scope of this paper.

B. Segmentation & Tokenization

SEGAUTH applies symbolic aggregate approximation (SAX) method [12] to segment each feature's time series and convert those segments into strings. SAX is effective in transforming a time series into symbolic representations, which reduces data dimensionality and can significantly speed up data processing such as comparison. Therefore, SAX is suitable for discovering repetitive patterns from time series. It has been proven that the distance measure in SAX symbolic representation space is to lower bound the distance between the time series in the original space. Refer to [12] for the detail of SAX and its applications.

For a time series T of length n, SAX first transforms the data into the Piecewise Aggregate Approximation (PAA) [7] representation and then symbolizes the PAA representation into a discrete string of length β (typically $\beta \ll n$) with alphabet size α. Before transformation, the time series is normalized to make its mean and standard deviation as 0 and 1 respectively. The SAX discretization produces symbols with equiprobability by first determining the breakpoints that produce α equal-sized areas under a $N(0, 1)$ Gaussian curve and then mapping PAA coefficients to SAX symbols based on those breakpoints. The concatenation of all the symbols of PAA representation forms a SAX string. In addition, trivial matches, which are the adjacent identical strings, are removed from the generated strings through numerosity reduction.

After string generation, the original time series is converted to a string array (Assuming the array size is N, the number of the generated strings). Next, a set of 2-grams (or bigrams) is generated from those strings and these 2-grams become the tokens. In general, 2-gram tokens can better capture a user's behavioral characteristic and differentiate users than unigrams (original individual strings). The S&T procedure is described in Algorithm 1 and illustrated in Figure 3.

Algorithm 1 Procedure of Segmentation & Tokenization

Input: time series of a sample's features $(f_1, f_2, ..., f_m)$
$\triangleright$ Let $\mathcal{T}_i$ be the time series of f_i, $\mathcal{T}_i = \{t_1^i, ..., t_n^i\}$
Parameters: SAX alphabet size α, string length β,
Sliding window size γ
Output: Set of tokens Tok for the sample

for all $\mathcal{T}_i \in \{\mathcal{T}_1, ..., \mathcal{T}_m\}$ **do**
  $\mathcal{S}_i \leftarrow \{S_j^i : S_j^i = \{t_j^i, ..., t_{j+\gamma-1}^i\}, 1 \leq j \leq |\mathcal{T}_i| - \gamma + 1\}$ $\triangleright$ Apply sliding window subsequence extraction to get set of segments $\mathcal{S}_i$, $|\mathcal{T}_i| = n$
  for all $S_j^i \in \{S_1^i, ..., S_{n-\gamma+1}^i\}$ **do**
    $Str_j^i \leftarrow SAX(S_j^i, \alpha, \beta)$ $\triangleright$ Apply SAX to convert S_j^i into a string
  end for
  Apply numerosity reduction to remove trivial matches $\triangleright$ $\mathsf{Str}_i \leftarrow \{Str_1^i, ..., Str_k^i, k \leq n - \gamma + 1\}$
  $\mathsf{Tok}_i \leftarrow \{Str_j^i || Str_{j+1}^i : 1 \leq j \leq k-1\}$
  $\bigcup \{Str_j^i || Str_{j+2}^i : 1 \leq j \leq k-2\}$
  $\bigcup \{Str_j^i || Str_{j+3}^i : 1 \leq j \leq k-3\}$
end for
$\mathsf{Tok} \leftarrow \{\mathsf{Tok}_i : i = 1, ..., m\}$
return Tok

There are several parameters, e.g., sliding window size (γ), PAA size or string length (β), and SAX alphabet size (α) in the procedure. The window size determines the length of a time series segment being converted into a string; the PAA size determines the number of letters in a string; and the alphabet size is the total number of the letters of the alphabet used in conversion. Normally, each letter corresponds to a range of numerical values with equal probability in order to ensure the fairness of alphabet conversion. The setting of those parameters is discussed in Section IV-C.

C. Model Generation

In model generation, the tokens from a user's genuine training samples are stored into the positive database for that user along with their frequency of occurrence in all genuine training samples. And the tokens from that user's forgery training samples are stored into the negative database in a similar manner. The probability of a token being genuine (i.e., from the authentic user's motion) can be calculated based on the information from the two databases. For instance, if a token occurs p times in genuine samples and q times in forgery samples, then its probability of being genuine is recorded as $p/(p+q)$. Therefore, if a string token occurs far more frequently in genuine samples than in forgery samples, then that token is more likely to be associated with the genuine user. Building such a model for each user lays the foundation for the next stage—classification. Table I exemplifies user information stored in the databases.

TABLE I: Sample user information stored in databases

String Tokens	GC*	FC#	Probability
AABC BCBA	24	2	92.3%
BBAA CABB	20	6	76.9%
BACA CBAC	12	12	50%
CAAA BCCC	4	15	21.1%

* GC refers to genuine count, # FC refers to forgery count

Verifying a testing sample as genuine or not relies on the sample's genuine score $G = \sum_{i=1}^{m} w_i \cdot p_i$, where m is the number of features, p_i is the probability of the sample being genuine determined solely by feature f_i, and w_i is f_i's associated weight. Thus, deriving the weight of each employed feature for each user is also performed in the model generation. When multiple features are employed in user authentication, it is common that those features differ in their capability of differentiating genuine users from imposters. SEGAUTH adopts a per-user feature weighting method by Snelick *et al.* [23], in which a metric d is designed to calculate each feature's weight w for each user (u). The metric d for feature f_i is defined as

$$d_i = \frac{\mu_i(gen) - \mu_i(imp)}{\sqrt{(\sigma_i(gen))^2 + (\sigma_i(imp))^2}}, i = 1, ..., m,$$

where gen and imp are the training sets for genuine samples and forgery samples respectively, μ_i and σ_i represent the mean and standard deviation of the probabilities of a sample being genuine based on feature f_i from u's genuine or forgery training samples respectively. The computation of probability based on single feature is detailed in the next section. d_i measures the difference of the probabilities between genuine samples and imposter samples on feature f_i. d_i becomes relatively large if the classification based on f_i can tell genuine samples from imposter samples more accurately. The weight w_i is then calculated using $w_i = d_i / \sum_{i=1}^{m} d_i, i = 1, ..., m$. Apparently, $\sum_{i=1}^{m} w_i = 1$. Basically, if a feature f_i is more capable of distinguishing genuine samples from forgeries then its weight w_i is larger.

D. Classification

SEGAUTH verifies a testing sample based on the sample's genuine score G. The genuine score is the weighted sum of the probabilities of the testing sample being classified genuine on each feature. SEGAUTH employs the naïve Bayes classification method in individual feature based classifications. Within the framework of Bayesian classification, verifying whether a testing sample Y with feature f_i is genuine or not is achieved by computing the probability of Y being genuine with the given set of n tokens (denoted as $X = \{x_1, ..., x_n\}$) derived from f_i's time series, i.e., $P(Y = g|X)$ or simply $P(g|X)$ where g (and $\neg g$) represents the genuine class (and forgery class). If the probability is equal to or greater than the predefined threshold, then the testing sample Y is classified as genuine.

Assume $y_0 = \neg g$ and $y_1 = g$. According to the Bayes' theorem,

$$P(g|X) = \frac{P(X|g) \cdot P(g)}{P(X)} = \frac{P(X|g) \cdot P(g)}{P(X|g) \cdot P(g) + P(X|\neg g) \cdot P(\neg g)}.$$

With the conditional independence assumption, we have $P(X_1, ..., X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$, where $X_1, ..., X_n$ are conditionally independent variables. Thus, we have

$$P(g|X) = \frac{P(g) \cdot \prod_i P(x_i|g)}{P(g) \cdot \prod_i P(x_i|g) + P(\neg g) \cdot \prod_i P(x_i|\neg g)}.$$

A default genuine probability σ will be assigned to the token if a token from a testing sample cannot be found in the databases. In our evaluation σ is set to 0.5 as we find that the variation in terms of overall accuracy with different σ (in range [0.2, 0.8]) is quite minor on the evaluation datasets.

III. Evaluation

In this section we present the evaluation results of SEGAUTH on the multiple datasets that have been used in previous studies. The datasets, sample selection methods, evaluation metrics, and evaluation results are detailed in the following.

A. Datasets

Considering availability, applicability, and representativeness, we selected 4 different datasets for evaluating SEGAUTH. These datasets cover 3 types of behavioral biometric data, i.e., 3-D motion gestures, 2-D unlock patterns, and 2-D tablet signatures. The detail of the datasets is listed in Table II.

TABLE II: Datasets used in Evaluation

Dataset	Category	User #	G # | F # *
uWave [13]	3-D motion gestures	8	70 | NA
MotionAuth [30]	3-D motion gestures	26	40 | NA
Pattern [4]	2-D unlock patterns	34	21 | 99
SUSIG [9]	2-D tablet signatures	94	20 | 10

*G and F refer to genuine samples and forgery samples for each user

The uWave and MotionAuth datasets are employed for 3-D gestures evaluation. The uWave dataset consists of 8 hand gestures, each of which has 70 samples by each user. The samples were collected using Wii remote controller with only accelerometer readings [13]. The MotionAuth dataset consists of 4 arm gestures (circle, down, up, and rotation) from 26 users with 40 genuine samples per user per gesture [30]. Those samples were collected using a smart watch with accelerometer and gyroscope sensors. Although the uWave dataset was originally used for gesture recognition instead of authentication, we include it to compare the accuracy that different verification techniques can achieve.

For 2-D unlock patterns, we select the Android Unlock Pattern dataset [4] as it is accessible and contains a relatively large set of unlock patten samples collected from 34 users each having 21 genuine samples and 99 skilled forgery samples (each user contributing 3 forgery samples to every other user). However, we encountered a few problems in data processing following the description in [4]: 1) The dataset only contains data for 34 users instead of 38 listed in the paper; 2) Not every user has 21 valid samples. For example, 2 users only have 19 samples. Therefore, we have to remove another 9 users (from the 34 users) who either are invalid based on the description in [4] and metadata information in the dataset or have samples less than the required number. In the end, we have 25 users in the Pattern dataset and each user is associated with 20 genuine samples and 72 skilled forgery samples.

Moreover, there is another subtle issue with the Pattern dataset. A number of phones of different models were used to collect data. Therefore, some users' forgery samples are not applicable in testing another user who used a different phone. Based on the phone information, the 25 users are divided into 3 groups: 16 users in group one using a phone of the same model, 4 users in group two using a different phone of the same model, and the rest 5 users all using a different phone. We only use the data from the 20 users in groups one and two when calculating the accuracy. Therefore, a user in group one (or group two) has 45 (or 9) valid skilled forgeries.

For 2-D tablet signatures, the SUSIG dataset [9] is chosen since it is popular in signature verification research. The SUSIG dataset contains signatures from 94 users with 20 genuine samples and 10 skilled forgery samples for each user.

B. Sample Selection

The classification of SEGAUTH requires both genuine and forgery samples. On all four datasets, we choose the same number of forgeries as that of genuine samples for classifier training. If selection of samples is random, we repeat the same experiment 5 times to reduce the impact of random selection and use the average of the results as the result of that experiment.

Regarding selection of forgery samples, the uWave and MotionAuth datasets have no skilled forgeries. Since all the participants in their studies performed the same set of gestures, a workaround is to make up a forgery using another user's sample of the same gesture, which was used in [30]. Therefore, for each gesture, we pick genuine samples of other users as the forgeries for the test user in the training phase of classification. In terms of selection of genuine training samples, some studies (e.g., [20], [30]) apply leave-one-out cross-validation method which uses $n-1$ out of n genuine samples for training, while other studies (e.g. [18], [4]) select the first k samples based on the order in which the data was acquired as the training set. Both methods have been used in our evaluation and each method is applied upon a particular dataset in order to compare the results to those reported in the literature.

C. Evaluation Metrics

The metrics used in our evaluation include accuracy (ACC) and equal error rate (EER). In this paper a true positive/negative (TP/TN) refers to a genuine/forgery sample being correctly accepted/rejected. And a false positive/negative (FP/FN) refers to a forgery/genuine sample being incorrectly accepted/rejected.

ACC measures how well a binary classification test correctly identifies a condition and is defined as the proportion of true results (TPs and TNs) among the total number of cases examined [15].

EER is a rate when the false negative rate (FNR) equals the false positive rate (FPR) and is often used to measure the

TABLE III: Features selected on the 4 datasets

System	Raw Features	Derived Features
uWave [13]	3-D acceleration (A)	magnitude (M), 3-D angle (θ), 1st derivatives of A, θ and M
MotionAuth [30]	3-D acceleration (A) & angular acceleration (G)	magnitude (M), 3-D angle (θ), 1st derivatives of A, G, θ and M
Pattern [4]	x, y coordinates	velocities in x, y, magnitude (M), and 1st derivative of M
SUSIG [9]	x, y coordinates	velocities in x, y, magnitude (M), and 1st derivative of M

quality of an authentication system. FPR = (#FP)/(#FP + #TN) and FNR = (#FN)/(#FN + #TP).

D. Evaluation Results

We apply SegAuth to each of the 4 datasets following the evaluation procedure used by previous studies on the same dataset and compare the results attained by SegAuth to the ones reported in the literature.

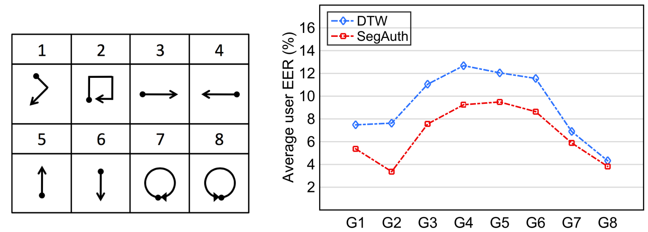
Given the diversity of the datasets, we select different features for different datasets. Our feature selection is guided by the principle of using common features that can be easily acquired or derived from ubiquitous motion sensors, e.g., accelerations, coordinates on a touch screen, velocity, etc. The features we use for each dataset are listed in Table III. Our feature selection can be further optimized for SegAuth to achieve better results but we want to examine the capability of SegAuth with a set of common features. Feature selection is further discussed in Section IV-B.

1) Results of uWave Dataset: The uWave dataset consists of 8 guestures, denoted as G1 to G8 and depicted in Figure 4 (a). The original study on the uWave dataset [13] does not provide authentication results on the 8 gestures. We apply both DTW (used in the original study for gesture recognition) and SegAuth to authentication using the same set of data and features. We use the leave-one-out method for training in both DTW and SegAuth.

For each gesture, we first obtain the EER values of all 8 participants when using DTW (and SegAuth) for authentication and then calculate the average EER value. Figure 4 (b) depicts the average EERs for the 8 gestures achieved by DTW and SegAuth. We can clearly see that in general SegAuth can achieve higher verification accuracy (i.e., smaller EERs) than DTW, especially when the gestures are relatively simple (e.g., G3-G6).

According to [13], all 8 gestures in the uWave dataset are hand movements. As hand gestures can be swift and subtle, a person's intrinsic motion characteristics in simple gestures such as horizontal hand movements may be better captured by SegAuth that focuses on motion segments than by the methods eyeing the entire motion.

2) Results of MotionAuth Dataset: We apply the same leave-one-out training method as the original study [30] does and compare the performance of SegAuth against DTW and Histogram methods used in [30] in terms of average user EER. Figure 5 depicts the per-gesture result achieved by SegAuth



(a) 8 gestures (b) Comparison of EER values

Fig. 4: uWave dataset and its evaluation results

along with those archived by the Histogram and DTW methods reported in [30]. Evidently, SegAuth outperforms the other two on all four gestures. Aligned with the observation in [30], SegAuth achieves higher accuracy with more complex gestures. More interestingly, SegAuth is able to maintain high accuracy for the relatively simple rotation gesture while DTW and Histogram have a much bigger increase for average EER.

A deeper look into the SegAuth results reveals that by using SegAuth 1) all participants on Circle gesture and nearly 90% of participants on Down and Up gestures achieve an EER lower than 5%, and 2) 3 out of 4 gestures have no participants' EERs higher than 10%. Those results are quite different from the results in [30] and clearly demonstrate the effectiveness of SegAuth in gesture based authentication.

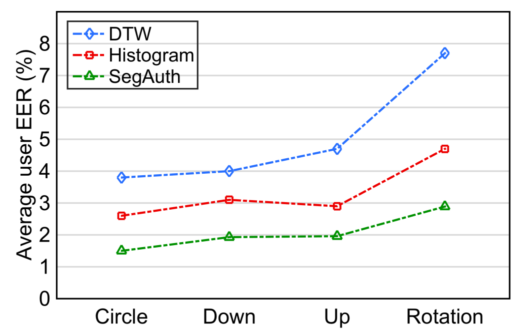


Fig. 5: Comparison of DTW, Histogram, and SegAuth on the MotionAuth dataset

3) Results of Pattern Dataset: For the Pattern dataset, we apply the first k training method used in [4] to compare our results with those reported in the original study [4] and evaluate the performance of SegAuth in terms of FNR, FPR, and ACC. The results, obtained with the genuine score threshold of 0.85, are listed in Table IV. Through experiments we find that the threshold of 0.85 renders a relatively balanced result in terms of FNR and FPR. The differences on #TP+#FN and #TN+#FP between SegAuth and DTW in Table IV are attributed to the issues in the dataset as explained in III-A.

The results shown in Table IV clearly indicate that SegAuth achieves a much lower FPR than DTW, which effectively enhances the unlock pattern security as SegAuth blocks more skilled attack attempts. Note that the results of SegAuth are obtained only using x and y coordinates. The results

TABLE IV: Comparison between SEGAUTH and DTW on Pattern dataset (Format: SEGAUTH value [DTW value reported in [4]])

#TP	#FN	#TN	#FP	ACC
246 [398]	54 [92]	588 [852]	68 [231]	87% [77%]
FNR: 18% [19%]		FPR: 10% [21%]		

could be further improved by using other measurements such as touch size, pressure and speed that appear useful in user authentication [4]. This also indicates that SEGAUTH is more capable of differentiating different users with similar behaviors.

4) Results of SUSIG Dataset: Regarding the SUSIG dataset, we use the first k training method to compare our result with those reported in the literature. Specifically, we first train the classifier using a user's first 5 genuine samples and 5 randomly selected forgery samples, and then test the user's remaining genuine samples and forgery samples to derive his/her FNR, FPR, and EER values. We repeat the experiment 5 times and use the averages of those values as the final result.

Table V lists the EER values achieved by different studies and the technique they used, include DTW, Fourier descriptors, and Histogram. Without special optimization in feature selection, SEGAUTH still outperforms the other techniques with a much higher accuracy (with 0.98% EER). As signatures are more personal and complex than other types of motion gesture, more distinctive behavior segments can be generated from a user's signature and therefore help improve verification accuracy significantly.

TABLE V: Comparison of EERs on the SUSIG dataset

System	Technique	EER
[9]	DTW	3.30
[8]	DTW	3.06
[31]	Fourier Descriptors	6.20
[19]	Histogram	6.08
[19]	Histogram (with weight)	4.37
SEGAUTH	Proposed Method	**0.98**

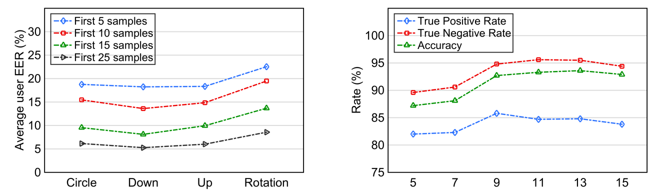
IV. DISCUSSION

In this section, we discuss several factors that can affect the SEGAUTH's accuracy and effectiveness including size of training set, feature selection, and setting of SAX parameters.

A. Size of Training Set

The size of training set affects both usability and accuracy of an authentication system. A larger training set usually helps generate more accurate classification models and therefore improves verification accuracy. However, a larger training set also means longer time for model creation and often asks for more training samples from a user, which potentially reduces users' interest and degrades the system's usability.

We study the impact of training sample size on verification accuracy using MotionAuth and Pattern datasets since both of



(a) MotionAuth dataset (b) Pattern dataset

Fig. 6: Impact of training sample size on accuracy

user number and samples per user in those two datasets are relatively large. We apply the first k sampling method to both datasets. We vary the number of genuine (and forgery) training samples from 5 to 10, 15, 25 for MotionAuth dataset, and from 5 to 7, 9, 11, 13, 15 for Pattern dataset and display the change in verification accuracy in Figure 6. Note that the x-axis and y-axis in Figure 6 (a) differ from those in Figure 6 (b). From 6 (a), it is obvious that increase of sample size correlates with decrease of average user EER attained by SEGAUTH for all 4 gestures on the MotionAuth dataset. For the Pattern dataset, the improvement on accuracy, TPR and TNR by increasing the training sample size can also be observed but it becomes minor and even slightly negative with increase of sample size.

B. Feature Selection

We use the MotionAuth dataset as an example to demonstrate the variation of different features in their capability of distinguishing genuine users from imposters. We perform single feature based verification for each feature and depict the average EERs in Figure 7. The first 10 features (f1-f10) are linear accelerations, angular accelerations, and angles in three dimensions, and magnitude. The second 10 features and last 10 features are the first and second derivatives of the first 10 features, respectively. Figure 7 evidently manifests that features differ in their distinguishability. Overall, the raw and derived features have better distinguishability than their first derivatives, which are better than the second derivatives in accuracy.

SEGAUTH uses the first 20 features in this study while all 30 features were used in the MotionAuth study [30]. We further calculate the average EERs of the 4 gestures using only the first 10 features and compare the average EERs achieved by SEGAUTH with 10, 20, 30 features for each gesture in Figure 8. We can easily notice that the accuracy with the 20 features is very close to that with all 30 features but is evidently smaller than the one using the first 10 features for every gesture. Since less features mean less computation and runtime, selecting 20 features for classification appears to be a better tradeoff between accuracy and efficiency than the other two alternatives. We note that feature selection is important for achieving high accuracy but optimal feature selection is out of the scope of this paper.

C. SAX Parameters

SEGAUTH leverages the SAX method to effectively reduce the dimensions of original time series and convert the continuous time series into discrete strings. Here we discuss the setting of sliding window size, PAA size, and SAX alphabet size.

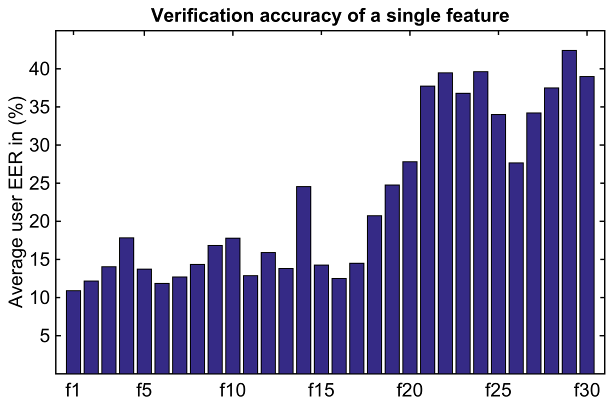


Fig. 7: Average user EER obtained from MotionAuth dataset using a single feature for classification

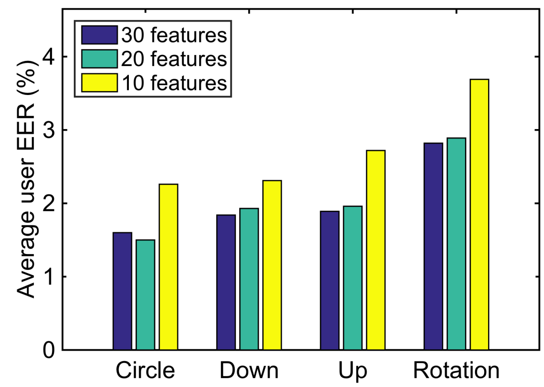


Fig. 8: Comparison of accuracy with different number of features on the MotionAuth dataset

Using sliding window is necessary to represent the original time series comprehensively and therefore is applied in SEGAUTH. However, it is difficult to set the size of sliding window. There is no standard value for reference. In addition, since the window size determines the length of a segment of the original time series to be converted, it should not be set too large or too small. Suppose the windows size is γ and the length of the time series is n, then the number of converted strings will be $n - \gamma + 1$. Given n, a larger γ produces fewer SAX strings and vice versa. Conceptually, the larger the window size, the less likely a user's subtle motion characteristic will be captured. On the other hand, the smaller the window size, the more likely noise will be introduced in SAX string generation. Based on our empirical study, the window size being half the length of the raw time series (i.e., $\gamma = n/2$) often deliver reasonably good results. Therefore, we set the window size approximately to $n/2$ in practice.

The PAA size β defines the length of a SAX string, i.e., the number of symbols in a SAX string. With PAA size β, a segment of one window size γ is equally divided into β sub-fragments and the length of each sub-fragment is γ/β. This parameter is often correlated with the window size. In our experiments, we set the PAA size to 5 or 6 and found the settings work well on different datasets.

The SAX alphabet size α determines the maximal number of unique symbols that may be used in a SAX string. Different from the PAA size β that decides the segmentation horizontally (on x-axis), the alphabet size α determines the segmentation of a time series vertically (on y-axis). SAX partitions the time series on the y-axis into α fragments with equal probability. Usually the first fragment is assigned symbol A, and then symbol B, ... and so on. Based on our empirical study, 4, 5 or 6 appear to be a good choice for the alphabet size in that a large value of α can make fragments too narrow to remove the noise introduced in data collection.

V. RELATED WORK

A biometric authentication system verifies a person based on either his or her physiological traits (e.g., fingerprint, face, iris, etc) or behavioral characteristics (e.g., finger or hand movements) [28], [33]. Biometric authentication is more user friendly in nature than those approaches relying on something you know (e.g., passwords) and something you have (e.g., security tokens). Physiological biometrics is subjected to various attacks [16], [25] and their accuracy can be largely affected by environmental factors such as illumination and background noise [2]. In contrast, behavioral biometrics appears more robust to theft or mimicry attacks.

Behavioral biometric authentication has been an active research area for many years. Early behavioral biometrics studies mainly focus on keystroke dynamics and mouse movements. As mobile devices are becoming ubiquitous, recent years have witnessed growing research interest in applying behavioral biometrics to mobile authentication. Many behavioral biometrics studies have been conducted based on phone touch operations [5], [29], [32], PIN and unlock pattern operations [4], [11], online signatures [19], [10], multitouch gestures [18], [21], [22] and three-dimensional gestures [13], [30], [28].

Unlock pattern mobile authentication has been widely deployed on smartphones and tablets. In [26], the security of unlock patterns was found to be comparable to 3-digit PINs. In [4], the authors analyzed multiple unlock patterns on Android unlock grid and achieved an accuracy of 77% using DTW.

Similar to unlock pattern, online signature authentication is also well studied. In [31], Yanikoglu and Kholmatov proposed a Fourier Descriptors authentication scheme to obtain an EER of 6.20% on the SUSIG dataset [9]. In [19] the EER for the same dataset is reduced to 4.37% with the proposed two dimensional histogram method. The highest accuracy on the SUSIG dataset reported in [31] is the EER of 3.30% achieved by using the DTW method. Ren *et al.* proposed a critical segment based online signature verification system to secure mobile transactions on multi-touch mobile devices and tested their approach extensively with 25 users over six months [17].

Our work differs from most existing approaches for behavioral biometric authentication. SEGAUTH utilizes symbolic aggregate approximation (SAX) method [12], which is able to transform time series data into symbolic representations to reduce the dimensionality/numerosity of the original time series. Compared to other dimensionality reduction methods, SAX retains many features of original time series such as time and shape. The string token generated based on the converted strings can effectively capture users' intrinsic behavior characteristics. Moreover, SEGAUTH employs a simple but effective classification method, Naive Bayes classification, which has been widely used in a variety of classification tasks such as email spam filtering and text classification [14], [27], [6].

VI. CONCLUSION

In this paper, we have presented a segment-based approach to behavioral biometric authentication namely SEGAUTH. SEGAUTH is unique in that it emphasizes on verifying a user based on his or her distinctive and repetitive segments of a gesture instead of the whole gesture used by previous methods. SEGAUTH leverages the symbolic aggregate approximation and Bayes' theorem to effectively and efficiently segment a time series, transform a segment into a string token, derive the genuine score of a gesture, and verify the user. We have evaluated SEGAUTH using 4 different datasets collected from 3 different types of behaviorial biometric authentication. SEGAUTH consistently achieves higher verification accuracy than the results reported by previous studies with all the 4 datasets. Our experimental results clearly demonstrate that SEGAUTH is a general and effective approach to behavioral biometric authentication.

ACKNOWLEDGMENT

This work was supported in part by the NIH grant UL1TR001427. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

REFERENCES

[1] 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*, 44(10):2468–2478, 2011.

[2] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Advances in Signal Processing*, 2004:430–451, 2004.

[3] Y. Chen, J. Sun, R. Zhang, and Y. Zhang. Your song your way: Rhythm-based two-factor authentication for multi-touch mobile devices. In *Proc. INFOCOM '15*, pages 2686–2694, 2015.

[4] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you!: Implicit authentication based on touch screen patterns. In *Proc. CHI '12*, pages 987–996, 2012.

[5] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, Jan. 2013.

[6] D. M. Freeman. Using naive bayes to detect spammy names in social networks. In *Proc. 2013 ACM workshop on artificial intelligence and security*, pages 3–12, 2013.

[7] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, 3(3):263–286, 2001.

[8] M. I. Khalil, M. Moustafa, and H. M. Abbas. Enhanced dtw based on-line signature verification. In *Proc. ICIP 2009*, pages 2713–2716, 2009.

[9] A. Kholmatov and B. Yanikoglu. Susig: an on-line signature database, associated protocols and benchmark results. *Pattern Analysis and Applications*, 12(3):227–236, 2009.

[10] Y. Li, M. Xie, and J. Bian. Usign–a security enhanced electronic consent model. In *Proc. IEEE EMBC'14*, pages 4487–4490, 2014.

[11] Y. Li, J. Yang, M. Xie, D. Carlson, H. G. Jang, and J. Bian. Comparison of pin- and pattern-based behavioral biometric authentication on mobile devices. In *Proc. MILCOM*, pages 1317–1322, 2015.

[12] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: A novel symbolic representation of time series. *Data Min. Knowl. Discov.*, 15(2):107–144, 2007.

[13] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

[14] A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48, 1998.

[15] C. E. Metz. Basic principles of roc analysis. *Seminars in nuclear medicine*, 8(4):283–298, 1978.

[16] N. Ratha, J. Connell, and R. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.

[17] Y. Ren, C. Wang, Y. Chen, M. C. Chuah, and J. Yang. Critical segment based real-time e-signature for securing mobile transactions. In *Proc. IEEE CNS 2015*, pages 7–15, 2015.

[18] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: A novel approach to authentication on multi-touch devices. In *Proc. ACM CHI '12*, pages 977–986, 2012.

[19] N. Sae-Bae and N. Memon. Online signature verification on mobile devices. *IEEE Transactions on Information Forensics and Security*, 9(6):933–947, June 2014.

[20] N. Sae-Bae, N. Memon, and K. Isbister. Investigating multi-touch gestures as a novel biometric modality. In *Proc. BTAS 2012*, pages 156–161, 2012.

[21] 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 *Proc. MobiCom '13*, pages 39–50, 2013.

[22] M. Sherman, G. Clark, Y. Yang, S. Sugrim, A. Modig, J. Lindqvist, A. Oulasvirta, and T. Roos. User-generated free-form gestures for authentication: Security and memorability. In *Proc. ACM MobiSys '14*, pages 176–189, 2014.

[23] R. Snelick, U. Uludag, A. Mink, M. Indovina, and A. Jain. Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):450–455, 2005.

[24] J. Sun, R. Zhang, J. Zhang, and Y. Zhang. Touchin: Sightless two-factor authentication on multi-touch mobile devices. In *Proc. IEEE CNS 2014*, pages 436–444, 2014.

[25] Q. Tang, J. Bringer, H. Chabanne, and D. Pointcheval. A formal study of the privacy concerns in biometric-based remote authentication schemes. In *Proc. ISPEC '08*, pages 56–70, 2008.

[26] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz. Quantifying the security of graphical passwords: The case of android unlock patterns. In *Proc. CCS '13*, pages 161–172, 2013.

[27] Q. Wang, G. M. Garrity, J. M. Tiedje, and J. R. Cole. Naive bayesian classifier for rapid assignment of rrna sequences into the new bacterial taxonomy. *Applied and environmental microbiology*, 73(16), 2007.

[28] C. Xu, P. H. Pathak, and P. Mohapatra. Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch. In *Proc. ACM HotMobile 2015*, pages 9–14, 2015.

[29] H. Xu, Y. Zhou, and M. R. Lyu. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *Proc. SOUPS 2014*, pages 187–198, 2014.

[30] J. Yang, Y. Li, and M. Xie. Motionauth: Motion-based authentication for wrist worn smart devices. In *Proc. the 1st Workshop on Sensing Systems and Applications Using Wrist Worn Smart Devices*, pages 550–555, 2015.

[31] B. Yanikoglu and A. Kholmatov. Online signature verification using fourier descriptors. *Journal on Advances in Signal Processing*, 2009:12, 2009.

[32] N. Zheng, K. Bai, H. Huang, and H. Wang. You are how you touch: User verification on smartphones via tapping behaviors. In *Proc. ICNP '14*, pages 221–232, 2014.

[33] N. Zheng, A. Paloski, and H. Wang. An efficient user verification system via mouse movements. In *Proc. ACM CCS*, pages 139–150, 2011.