Continuous Authentication of Smartphone Users by Fusing Typing, Swiping, and Phone Movement Patterns

Rajesh Kumar
Syracuse University, USA
rkuma102@syr.edu

Vir V. Phoha
Syracuse University, USA
vvphoha@syr.edu

Abdul Serwadda
Texas Tech University, USA
abdul.serwadda@ttu.edu

Abstract

We studied the fusion of three biometric authentication modalities, namely, swiping gestures, typing patterns and the phone movement patterns observed during typing or swiping. A web browser was customized to collect the data generated from the aforementioned modalities over four to seven days in an unconstrained environment. Several features were extracted by using sliding window mechanism for each modality and analyzed by using information gain, correlation, and symmetric uncertainty. Finally, five features from windows of continuous swipes, thirty features from windows of continuously typed letters, and nine features from corresponding phone movement patterns while swiping/typing were used to build the authentication system. We evaluated the performance of each modality and their fusion over a dataset of 28 users. The feature-level fusion of swiping and the corresponding phone movement patterns achieved an authentication accuracy of 93.33%, whereas, the score-level fusion of typing behaviors and the corresponding phone movement patterns achieved an authentication accuracy of 89.31%.

1. Introduction

A significant amount of research has recently explored how the sensors built in smartphones could aid in user authentication. Some of the most studied sensors include the touch sensor [1–4], the motion and orientation sensors [5, 6], the microphone [7] and the camera [8]. While the majority of research on these sensors has found them promising in user authentication, the adoption of these sensor-driven authentication systems in practical applications continues to be a distant dream. Some of the potential reasons for this challenge include: (1) High error rates– a number of studies on such authentication systems continue to report high error rates (see [2, 5]) which are far from the thresholds specified by NIST for authentication systems, (2) Realism of the experiment settings– many of the studies on this type of authentication use constrained experimental settings, which makes it difficult to extrapolate how the system would perform in the wild e.g., see experiments where users are constrained to perform a particular task [1–4, 9], (3) Intermittent availability of data– in most studies, researchers focus on a single sensor and evaluate how data from this sensor discriminates between users. The challenge with these scenarios is that, in practice, a single sensor would provide useful data during only those spells when the user undertakes activities which trigger this sensor. During spells when the sensor is not used, an authentication system designed based on the sensor in question would be redundant, which in turn implies it would not help defend against active adversaries at that point in time, and (4) Spoof attacks– it has been shown that several of the systems based on the individual sensors can be spoofed by adversaries who have access to population statistics [10–13] or those who have access to user-specific data [10, 11, 13, 14].

The sum-total of these challenges presents a major impediment to the potential realization of these methods in real systems. In this paper, we design and evaluate an authentication system which takes steps towards addressing these challenges. Specifically, we present a fusion-based authentication mechanism for smartphone users that combines typing, swiping, and phone movement patterns while typing/swiping as respectively recorded by the touch and motion sensors. The combination of these sensors does not only result in low error rates but also naturally presents a defense to the spoof attacks designed to defeat the individual sensors. The low error rates result from the multiple sensors providing a large amount of information about the users behavior, while the spoof resistance emanates from the difficulty an attacker would face to forge samples from all sensors at once.

To cap it all, we evaluate our system using data which was collected while users freely interacted with the device during routine web browsing, allowing us to provide insights on sensor-driven authentication from a perspective that is drastically different from the vast majority of past research. To ensure that we collect a sufficient number of touch and keystrokes, we provided seven browsing exercises to the participants spread over a period of at least four days. The core of our authentication system is the feature and score-level fusion framework which dynamically combines information expressing the user's typing, swiping and phone movement behavior.

The contributions of our work are summarized below:

- Using data collected from 28 users over a period of four to seven days under a completely unconstrained environment, we designed a multimodal fusion-based continuous authentication system. To the best of our knowledge, no past work has fused this full set of sensors and performed evaluations in a realistic setting which closely mirrors real-world conditions.

- We extracted a large number of features from all three modalities and rigorously evaluated their informativeness using several feature quality measures. Our belief is that this corpus of ranked features will be a handy resource for researchers conducting further research in this area.

- An algorithm is presented for implementing a multimodal framework. Although the framework focuses only on the typing, swiping, and phone movement patterns, it can be easily extended to accommodate any number of modalities. Additionally, we implemented a multi-template classification framework (MTCF), especially for classifying swipe gestures and the corresponding phone movement patterns. Our experimental finding shows that it is better to use MTCF compared to a traditional single-template based classification framework (STCF).

The rest of the paper is organized as follows: Section 2 presents related work; Section 3 describes the data collection, preprocessing, and feature analysis; Section 4 discusses authentication/classification framework, training/testing of classifiers, feature/score-level fusion; Section 5 talks about the performance of different modalities; finally, Section 6 concludes our work.

2. Related Work

The touch- and typing pattern based continuous authentication systems have been widely studied recently [1–4, 15–17]. However, these authentication systems, like classical biometrics, are also susceptible to mimicry-attacks e.g. see [10, 11]. Sitova et al. [9] studied the phone movement patterns as hand-movements, orientation and grasp (HMOG) under two specific conditions: walking and sitting. They showed that the phone movement patterns while typing achieved equal error rates (EERs) of 19.67% and 13.62% respectively under the sitting and walking conditions. The fusion of typing patterns with HMOG achieved EERs of 7.16% and 10.05% respectively for walking and sitting conditions.

Similarly, we propose to fuse the phone movement patterns (before, while, and after swiping/typing) with the swiping or typing behaviors. However, our work differs from Sitovas on the following aspects: (i) our data collection was completely unconstrained, (ii) we apply feature-level fusion of modalities in addition to the score-level fusion, (iii) we present a comparative analysis of single- and multi-template frameworks for different kinds of swipe gestures, (iv) and we test the system under continuous authentication paradigm and report mean error rates.

The fusion not only improves the classification accuracy but also provides more complex feature space. It may be possible for adversaries to train a robot [10] or a human imitator [11] to imitate swipe/type or phone movement patterns separately. However, we believe that it will be extremely difficult to imitate both (swiping/typing patterns and the corresponding phone movement patterns) simultaneously, especially when the features extracted from both of the modalities are less correlated.

3. Data Collection and Feature Analysis

3.1. Data Collection

Following IRB approval, we invited university students and staff members to participate in our data collection experiment. The participants were pre-informed that i) their swiping/typing behaviors shall be collected while they used our customized web browser, ii) their phone movement patterns would be collected all the time. More than 85% of the participants were university students, while the rest were university staff or faculty.

Figure 2 summarizes the system architecture used for the data collection. Smartphones running Android, version 4.0 were used to collect the data with no hardware and/or software modifications. The data collection app consisted of two core components: a customized web browser called Lightening [18] to collect typing and swiping patterns, and a service that runs in the background to collect phone movements continuously through an accelerometer sensor. On its first startup, the web browser was programmed to start the service.

Participants browsed a series of exercises through the web browser by swiping back and forth and typing responses to them. A total of seven exercises were provided, one for each day, and each containing twenty objective and ten subjective questions. Participants were not given specific instructions on how and when to attempt the exercises, however, they were required to type at least 1000

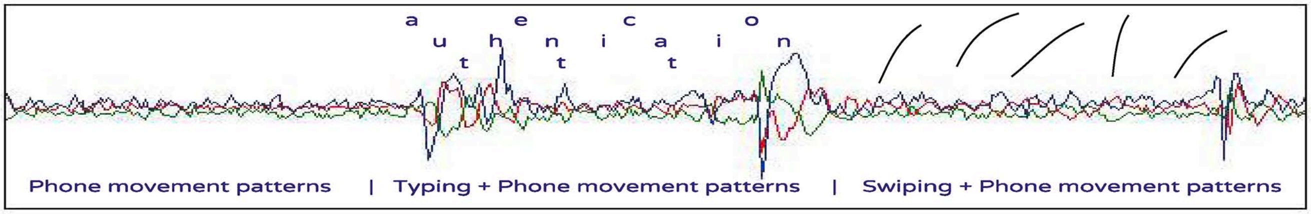

Figure 1. Availability of a smartphone user's activities on the phone over the time.

Table 1. List of features extracted from a window of phone movement patterns corresponding to each window of swipe gestures. The X, Y, Z are the accelerometer readings in X, Y, and Z dimensions. M is the resultant acceleration and is defined as $M = \sqrt{X^2 + Y^2 + Z^2}$

Mean M	Mean X	Mean Y	Std Y	MedianFreq Y	MedianFreq X	AbsSum X	AbsSum Y	AbsSum Z

letters (= 10 questions × 100 letters for each question) and generate on an average 25 swipes to complete one exercise. Additionally, we encouraged participants to freely browse pages of their choice in addition to attempting the daily exercises. This helped ensure that the browsing activity seen in our study was reflective of users free behavior while they interact with a web browser in their daily life. So, our dataset not only consisted of those gestures that were generated while users answered the questions but also consisted of those gestures that were generated while users freely browsed pages of their choice. Since the way in which users browse the web (i.e., by swiping and occasionally clicking a link or button) is very similar to how they interact with a significant proportion of apps and, our experimental setting also addresses the challenges that are posed by app browsing.

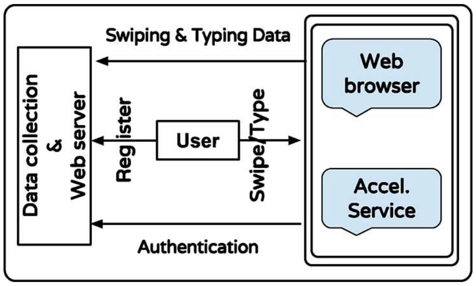

Figure 2. Underlying architecture for data collection.

To collect typing pattern, Javascript was to record time stamps associated with key presses and release events for each character. The swiping gestures that were recorded by the browser contained action type, orientation, x, y, pressure, area, and time, and at every touch point along the touch stroke (or swipe). The fields respectively represent the type of action (i.e. finger up, down, or moving), the orientation of the phone (landscape or portrait), the x-coordinate of a point touched, the y-coordinate of a point touched, the pressure exerted on the phone by the finger, the area occluded between the finger and the phone screen and the time at which a point is touched.

The typing and swiping data was sent to the server whereas the data generated through phone movement patterns i.e. (linear acceleration in x, y and z directions, rotation vector, and timestamp) was saved on the phone to prevent data transfer costs. The delay for collecting accelerometer data was set to SENSOR_DELAY_NORMAL that generated an average of five samples per second mostly and consumed the least amount of battery.

3.2. Preprocessing and Feature Analysis

3.2.1 Availability of activities and preprocessing

Figure 1 shows a subset of activities individuals perform typically while they interact with the smartphone. Observe that the phone movement patterns are available throughout the interaction time (e.g., while a person types, swipes, etc.). Typing and swiping, on the other hand, do not happen all the time and occur during non-overlapping time intervals. The availability of phone movement patterns at all times provides a perfect scenario for fusion with swiping data or typing data whenever any of the latter modalities is available.

We extracted the segments of phone movement patterns (i.e. accelerometer readings) corresponding to individual swipe gestures. We hypothesized that the phone movement patterns before, after and while typing/swiping is useful and may be unique to every user. Hence, for each swipe, we extracted an additional three seconds of phone movement patterns that was generated before and after each swipe in addition to the data along with the swipe. Similarly, we extracted segments of phone movement patterns corresponding to the windows of typed characters.

3.2.2 Swiping and phone movement patterns

To make our authentication system continuous in nature, we applied a sliding window-based mechanism for extracting

Table 2. List of features extracted from sliding windows of swipes. Windows of four consecutive swipes were used. We plotted x and y coordinates of touch points of all the swipes over a time-scale and observed some unique pattern. Hence, we treated these points as signals and extracted some features from signal processing domain such as the energy of the signal x and y. The energies of these signals were selected among the best features for classification.

Sum of pairwise distance among touch points	Mean Pressure	Mean Area	Energy of the signal formed by X coordinates of touch points	Energy of the signal formed by Y coordinates of touch points

Table 3. The symmetric uncertainty of 30 character pairs selected from the 40 most frequent character pairs. We used these pairs as features for classification.

ce: .0617	ha: .0541	us: .034	co: .033	te: .023	or: .021	se: .0194	an: .0153	ar: .0152	le: .0150
me: .013	ri: .010	of: .0073	es: .0071	is: .0068	al: .0065	ou: .0053	in: .0051	ca: .0042	on: .0039
nd: .0039	be: .0038	to: .0035	at: .002	en: .0018	er: .0017	ne: .0016	ng: .0016	ed: .0015	th: .0015

features from each of the modalities. For swiping and phone movements patterns while swiping, we used a window of four consecutive swipes with a sliding window of two swipes (see Algorithm 2).

The list of features extracted from the window of swiping gestures and corresponding phone movement patterns are presented respectively in Table 2 and 1. These features were selected by the correlation-based feature subset selection method [19] from a total of 36 features extracted from the phone movements data and seven features from the swipe gestures data. We observed that the feature reduction not only reduced the computational complexity but also improved classification performance significantly.

3.2.3 Typing and phone movement patterns

Similar to swipes, we used a sliding window 80 typed characters with a sliding window of 40 characters. From each window of typed letters, we extracted key hold times (KHTs) for each character and key interval times (KITs) for all possible pairs. The KHT is the latency between the press and release of a given key, while the KIT is the latency between the release of a key and the press of the next key.

While both KHTs are KITs are widely studied in keystroke dynamics for desktop, we observed KHTs performed very poorly during preliminary experiments may be due to poor clock resolution [20] so we did not study them further. Given a large number of possible digraphs 729($= 27 \times 27$) considering 28 letters of the alphabet and a shift key on the smartphone keypad, we evaluated the discriminative power of 40 most frequent digraphs so as to focus on a smaller number of highly informative digraphs.

We used the symmetric uncertainty to evaluate the informativeness of all 40 most frequent digraphs. The symmetric uncertainty (SU) of a diagraph is computed as, $SU = 2 \times I(F, U)/(H(F) + H(U))$. The term I(F,U) is the mutual information between the feature and the class labels while the $H(F)$ and $H(U)$ are respectively the entropy of the feature and the entropy of the class labels. We selected top 30 digraphs for classification (see Table 3). For phone movement patterns, we used accelerometer readings along with the windows of 80 consecutively typed characters with an overlap of 40 characters. Similar to phone movements while swiping, we extracted the same set of features (see Table 1) from the data collected from the phone movement patterns while typing.

Algorithm 1: Multimodal authentication framework

```
Input: $\Psi$={phone movements, swiping, typing}
//Availability of modalities
Input: $a_{conf}$, $c_{thr}$, and $a_{flag}$
//$a_{conf}$: Authenticity confidence,
//$c_{thr}$: Threshold for $a_{conf}$,
//$a_{flag}$: User active/inactive flag
while ($a_{flag}$) do
    if (($a_{conf} \leq c_{thr}$) $\wedge$ ($\Psi(1) \vee \Psi(2) \vee \Psi(3)$)) then
        while ($\Psi(1) \vee \Psi(2) \vee \Psi(3)$) do
            //Checking the availability
            of modalities in order of
            their performance
            if ($\Psi(1) \wedge \Psi(2)$) then
                $updateAuthConf(\Psi(1), \Psi(2))$
            else if ($\Psi(1) \wedge \Psi(3)$) then
                $updateAuthConf(\Psi(1), \Psi(3))$
            else if ($\Psi(2)$) then
                $updateAuthConf(\Psi(2))$
            else if ($\Psi(3)$) then
                $updateAuthConf(\Psi(3))$
            else
                $updateAuthConf(\Psi(1))$
            end
        end
    else
        $updateAuthConf(fingerprint \vee face \vee$
        $PIN \vee password \vee ...)$
    end
end
```

4. Design of Experiments

4.1. Multimodal Framework

One of the major issues in building a complete continuous authentication for smartphones is the availability of unique behavioral patterns across the interaction timeline. Smartphone users generally interact with the phone by swiping, typing, zooming, speaking, clicking, etc. in a random order. Hence, the security provided by an individual modality based authentication system is not comprehensive. For example, swiping- (or touch-) based authentication alone is not sufficient to cover the entire interaction window (see Figure 1). This issue defeats the philosophy of the continuous authentication which requires the continuous monitoring of the access to the device. Adversaries can exploit the windows of interaction where swiping does not take place to get into the system. Therefore, to develop a complete continuous authentication system for smartphone users, we need a multi-modal framework that uses more than one modality (as per their availability) and possibly fuses them in order to cover the entire interaction window of users with the phone.

Algorithm 1 presents steps to implement a tri-modal framework. We believe that it is not mandatory to keep the continuous authentication module active all the time as it is quite resource consuming. Therefore, we proposed to use a measure of the authenticity of the user that is called authenticity confidence and represented by a_{conf}. The latest value of a_{conf} and a predefined threshold c_{thr} are used to decide whether to enable the continuous authentication module or not. The value of a_{conf} decreases based on time spent on the phone by the user. It increases when the system receives legitimate biometric patterns that successfully verifies the identity of the user. If none of the modalities are available and a_{conf} goes below to the c_{thr}, users are prompted to enter a PIN, a password, face, fingerprint etc. to verify their identity.

In order to achieve the best possible classification accuracy, we have organized the if-else block in such a way that the system first searches for the best available combination of modalities. For example, the fusion (combination) of swiping and phone movements while swiping achieves the best accuracy at the time of validation compared to the fusion of typing behaviors and phone movements while typing. In this scenario, we first check for the availability of swiping and phone movements while swiping; if both are available, we use the fusion of these two to update the authentication confidence a_{conf} by invoking updateAuthConf(). The updateAuthConf() function is an overloaded function that takes one or more modalities and update a_{conf} based on the classification score obtained by the supplied modalities.

4.2. Choice of Classifiers

We used k-NN (k=11) with Euclidean distance [21], and random forest [22] with one thousand trees. We studied these two classifiers only because there exist several studies that compare classifiers for touch-based authentication and these two have been tested and proven to be the good ones in this area [1,2].

4.3. Training and Testing of Classifiers

We divided the whole dataset into two equal parts. Since the data was collected over multiple (four to seven) days, the divided data naturally created an inter-session scenario. We trained both classifiers for every user separately. To train classifiers for a user, we first created genuine and imposter feature vectors by using the sliding window mechanism (see line 1-6 Algorithm 2).

The genuine feature vectors were created from the corresponding users training data whereas impostor feature vectors were created by using the data from the rest of the users. We selected five random vectors from each of the rest of the users to create a total of 108 (=27×4) imposter feature vectors for each user.

Algorithm 2: The feature level fusion framework.

```
   Input: $n, d, s_{index}, a_{conf}, u_{thr}$, and $\delta$
   //n:# of swipes in a window,
   //d:# of swipes to slide,
   //$s_{index}$:Index of first swipe,
   //$a_{conf}$ : authenticity confidence,
   //$u_{thr}$ : user-specific threshold,
   //$\delta$:  conf update factor,
   Input: $\omega_s = \{s_1, s_2, s_3, ...\}, \omega_p = \{p_1, p_2, p_3, ...\}$
   //$\omega_s$ :Stream of swipe gestures,
   //$\omega_p$ :Stream of phone movement
   Input: $c_{template}$ //Classification template
1  $s_{index} \leftarrow -d$ //initializing $s_{index}$
2  while $(\omega_s \wedge \omega_p)$ do
3      $s_{window} \leftarrow \omega_s(s_{index} + d : s_{index} + d + n)$
4      $p_{window} \leftarrow \omega_p(s_{index} + d : s_{index} + d + n)$
5      $s_{index} \leftarrow s_{index} + d$
6      $s_{fv} \leftarrow getFeatures(s_{window})$
7      $p_{fv} \leftarrow getFeatures(p_{window})$
8      $c_{fv} \leftarrow fuseFeatures(s_{fv}, p_{fv})$
9      $c_{score} \leftarrow getMatchScore(c_{fv}, c_{template})$
10     if $c_{score} \geq u_{thr}$ then
11         $a_{conf} \leftarrow a_{conf} + \delta$
12     end
13 end
```

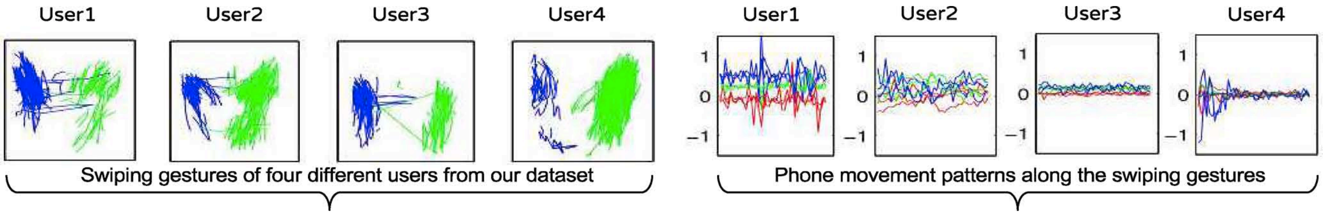

Figure 3. The appearance of swipe gestures on two opposite sides of the screen formed the basis to explore the multi-template classification framework. An obvious and unique patterns can be observed in phone movement patterns, e.g., User1 seems pressing the phone too hard while swiping.

4.4. Multi-template Classification Framework

We plotted swiping gestures of every user and observed that a majority of users had swiped in two different parts of the screen i.e. left and right (see Figures 3 and 4). The left swipes were generated from the left hand and right from the right hand of the user as it is impractical for an individual to swipe on two opposite sides of the smartphone screen using only one hand. Since a significant percentage of the total swipes appeared on the left so discarding them was not an option (see Figure 4).

Most of the researchers have used location-based features (e.g. coordinates at the start, mid and end of swipes) that may not be very useful in the scenario where swipe gestures appear across the screen [1–4]. We believe that the location of swipes also depends on the kind of application the user is interacting with. We address this problem by applying two different techniques: first by defining location independent features (velocity, length, area, and pressure of the swipe (see Table 2)); second by creating two separate templates, namely, left template and right template. The left template was created using all swipes that appeared on the left side of the screen, whereas, the right template was created using all swipes that appeared on the right part of the screen.

To identify the type of swipes we rely on the coordinates of the touch-points. If 80% of the touch points of a swipe gesture lie on the left part of the screen (with the screen divided into two equal parts vertically), we classify it as a left-swipe otherwise as a right-swipe. The *80% criteria* was able to separate the left and right swipes for all of the users.

For testing, we first identified the type (left or right) of the incoming swipe, extracted features, and then fed the feature vector to the classifier for finding the matching score with the corresponding template. Further, we evaluated the performance of each modality under both, single and multi-template classification frameworks; we present their performance in Table 4. It can be observed that the multi-template framework outperforms the single-template approach. Therefore, we suggest using a multi-template classification framework whenever it is possible to accurately find out what template an incoming pattern belongs to.

4.5. Fusion-based Classification Framework

We studied the fusion of modalities at feature- and score-level fusion, that are discussed below:

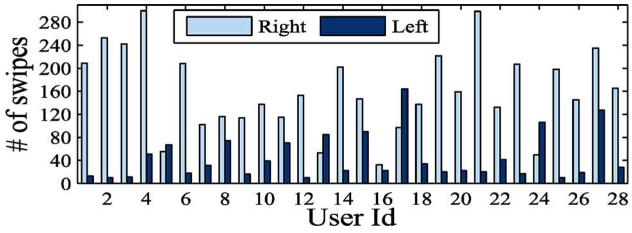

Figure 4. User-wise distribution of number of left and right swipes.

4.5.1 Feature level fusion (FLF)

Algorithm 2 presents the implementation steps for the feature-level fusion of swipe gestures and phone movement patterns. To fuse these modalities, we first checked for their availability. Then, we extracted the next window of swipes from the stream of incoming swipes, and the corresponding window of accelerometer readings to extract five and nine features respectively. Later, we concatenated these features to obtain a fused-features vector that contained a total of 14 features. The fused feature vector was then supplied to the matcher i.e. to the getMatchScore() function. This function returned the probability of the fused feature vector to be of a genuine type. The results of the feature-level fusion is presented in Table 4 and discussed in Section 5.

4.5.2 Score level fusion (SLF)

Similar to the FLF, we first extracted feature vectors from each of the to be fused modalities. These feature vectors were supplied to the corresponding classifiers to get the match scores. To compute the final match score, we take a weighted average of these scores by using the formula: $s_f = (w_s \times s_s) + (w_p \times s_p)$, where, s_s and s_p be the classification scores obtained after classifying swipe gestures and the corresponding phone movements respectively and $w_s + w_p = 1$. To find the range of

Table 4. Classification performance of authentication systems based on swipes, phone movement patterns while swiping and their fusion at feature and score-level under both, single and multi-template classification framework. The abbreviations used in this table are, PMs: phone movement patterns; STCF: single-template framework; MTCF: multi-template classification framework.

Modality Used for Classification	**Average performance**					
	kNN-Euclidean			**Random Forest**		
	FAR(%)	**FRR(%)**	**Acc(%)**	**FAR(%)**	**FRR(%)**	**Acc(%)**
Swiping gesture w/ STCF	11.08	12.52	88.15	12.71	6.75	90.51
Swiping gesture w/ MTCF	**7.92**	**15.21**	**87.47**	**10.09**	**5.59**	**92.89**
PMs while swiping w/ STCF	14.71	23.63	80.55	14.06	14.54	86.02
PMs while swiping w/ MTCF	**11.53**	**23.42**	**81.13**	**13.66**	**14.69**	**86.12**
FLF of swipes & PMs while swiping w/ STCF	8.42	9.44	91.03	12.19	6.25	91.02
FLF of swipes & PMs while swiping w/ MTCF	**6.84**	**10.20**	**91.15**	**11.44**	**4.23**	**93.33**
SLF of swipes & PMs while swiping w/ STCF	**11.08**	**11.90**	**88.45**	**7.20**	**7.01**	**92.85**
SLF of swipes & PMs while swiping w/ MTCF	9.41	18.70	84.10	6.13	7.78	92.80

weights that gives the best performance, we repeated the fusion experiments starting from $w_s = 1.0$ and $w_p = 0.0$ with a respectively decreasing and increasing step of 0.02, and continued until $w_s = 0.0$ and $w_p = 1.0$. The range of weights that achieved the best accuracy during the SLF of swiping and corresponding phone movement patterns were $w_s = [0.86, 0.96]$ and $w_p = [0.14, 0.04]$ for both single template classification framework (STCF) and multi-template classification framework (MTCF). Similarly, the SLF of typing behaviors and corresponding phone movement patterns was also carried out and achieved the best accuracy for $w_t = [0.88, 0.94]$ and $w_p = [0.12, 0.06]$, where, w_t and w_p are the weights assigned to the scores obtained from typing corresponding phone movement patterns.

Table 5. Classification performance of authentication systems based on typing behaviors, phone movement patterns while typing and their fusion at score-level. The abbreviations used in this table are, PMs: phone movement patterns, S Verifier: Similarity verifier, and RandFor: random forest.

Modality & Classifiers used for classification	**Classifier performance**		
	FAR	**FRR**	**Acc**
Typing behavior (S Verifier)	11.31	13.65	88.45
PMs while typing (RandFor)	16.43	18.71	81.53
Typing and PMs (SLF)	10.33	12.57	89.31

5. Performance Evaluation

We present the classification performances by using three metrics, namely, average false accept rates (FAR), average false reject rates (FRR) and average accuracy (Acc). These metrics are computed from a series of continuous scores returned by the classifiers, one for each feature vector. In order to give an authentication decision that is to decide whether a feature vector belonged to the genuine class, we used user-specific thresholds. The user-specific thresholds are created at the time of enrollment process.

We computed equal error thresholds by using the genuine and imposter scores obtained during the validation. For validation, we used half of the training data to train and the other half to validate. Following sections talk about the performance of the specific modalities and their possible fusion.

5.1. Swiping and phone movement patterns

We used two classifiers, namely, k-NN and random forest for verification purpose. In order to compare MTCF and STCF, we ran experiments under both the setup and computed the mean FARs, FRRs, and accuracies. Table 4 presents the results for swiping gestures, phone movement patterns while swiping, and their fusion at the feature- and score-level. We can observe that MTCF always performs better than STCF. The accuracies obtained by each modality and their fusion are significant, especially considering the fact that our data was collected in a completely unconstrained environment. Also, the FLF outperforms the SLF in most of the cases. Therefore, we suggest using the MTCF and FLF frameworks with the random forest classifier to achieve the best performance.

5.2. Typing and phone movement patterns

We evaluate the performance of typing behavior and the corresponding phone movement patterns by using similarity verifiers [13] and random forest. We studied the fusion of these two modalities at score-level only. Table 5 presents mean FARs, FRRs, and accuracies for both of the modalities and their fusion at score-level. We can observe that the score-level fusion is able to improve performance.

6. Conclusion and Future Work

We investigated three modalities, namely, swiping gestures, typing behavior, phone movement patterns while typing/swiping, and their possible fusion at the feature- and score-level for authenticating smartphone users continuously. Our experimental findings suggest that the fusion of available modalities improves the overall authentication accuracy. Specifically, the fusion of swiping gestures and corresponding phone movement patterns at the feature-level achieves the best classification accuracy. Similarly, the fusion of typing behavior and corresponding phone movement patterns at the score-level outperforms the authentication systems based only on typing or corresponding phone movement patterns. Also, for building an authentication system that included swipe gestures, a multi-template framework is recommended.

In the future, we plan to investigate the following: the possibility of fusion of typing behavior and corresponding phone movement patterns at the feature-level; phone movement patterns while not typing/swiping; define metrics that can be used to evaluate the overall performance of a multimodal fusion-based authentication system; and the behavioral patterns other than swiping and typing.

7. Acknowledgment

We thank Sujit Poudel, Diksha Shukla, and Zibo Wang for their invaluable contribution during data collection. This work was supported in part by DARPA Active Authentication contract: FA 8750-13-2-0274 and by National Science Foundation Award SaTC Number: 1527795.

References

[1] 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-TIFS*, vol. 8, pp. 136–148, Jan 2013.

[2] A. Serwadda, V. V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," in *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pp. 1–8, Sept 2013.

[3] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones.," in *NDSS*, The Internet Society, 2013.

[4] H. Xu, Y. Zhou, and M. R. Lyu, "Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones," in *SOUPS 2014*, pp. 187–198, USENIX Association, July 2014.

[5] A. Primo, V. V. Phoha, R. Kumar, and A. Serwadda, "Context-aware active authentication using smartphone accelerometer measurements," in *CVPRW, 2014*, pp. 98–105.

[6] R. Kumar, V. V. Phoha, and R. Raina, "Authenticating users through their arm movement patterns," *CoRR*, vol. abs/1603.02211, 2016.

[7] H. Lu, A. Brush, B. Priyantha, A. Karlson, and J. Liu, "Speakersense: Energy efficient unobtrusive speaker identification on mobile phones," in *The Ninth International Conference on Pervasive Computing (Pervasive 2011)*, June 2011.

[8] S. Chen, A. Pande, and P. Mohapatra, "Sensor-assisted facial recognition: An enhanced biometric authentication system for smartphones," MobiSys '14, (New York, NY, USA), pp. 109–122, ACM, 2014.

[9] Z. Sitov, J. Łedenka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, "Hmog: New behavioral biometric features for continuous authentication of smartphone users," *IEEE-TIFS*, vol. 11, pp. 877–892, May 2016.

[10] A. Serwadda, V. V. Phoha, Z. Wang, R. Kumar, and D. Shukla, "Toward robotic robbery on the touch screen," *ACM TISSEC*, vol. 18, pp. 14:1–14:25, May 2016.

[11] C. M. Tey, P. Gupta, and D. Gao, "I can be you: Questioning the use of keystroke dynamics as biometrics.," in *NDSS*, The Internet Society, 2013.

[12] A. Serwadda and V. V. Phoha, "Examining a large keystroke biometrics dataset for statistical-attack openings," *ACM-TISSEC*, vol. 16, pp. 8:1–8:30, Sept. 2013.

[13] K. A. Rahman, K. S. Balagani, and V. V. Phoha, "Snoop-forge-replay attacks on continuous verification with keystrokes," *IEEE-TIFS-2013*, pp. 528–541.

[14] R. Kumar, V. V. Phoha, and A. Jain, "Treadmill attack on gait-based authentication systems," in *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*, pp. 1–7, Sept 2015.

[15] S. Zahid, M. Shahzad, S. Khayam, and M. Farooq, "Keystroke-based user identification on smart phones," in *RAID* (E. Kirda, S. Jha, and D. Balzarotti, eds.), vol. 5758 of *Lecture Notes in Computer Science*, pp. 224–243, Springer, 2009.

[16] A. Buchoux and N. L. Clarke, "Deployment of keystroke analysis on a smartphone," in *Australian Information Security Management Conference*, p. 48, 2008.

[17] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: User verification on smartphones via tapping behaviors," in *IEEE-ICNP 2014*, pp. 221–232, Oct 2014.

[18] A. Restaino, "Lightning browser." `https://github.com/anthonycr/Lightning-Browser`, 2014. [Online; Last accessed 04 March, 2014].

[19] M. A. Hall, *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.

[20] K. Killourhy and R. Maxion, "The effect of clock resolution on keystroke dynamics," RAID '08, (Berlin, Heidelberg), pp. 331–350, Springer-Verlag, 2008.

[21] D. Aha and D. Kibler, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[22] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.