On the Effectiveness of Sensor-enhanced Keystroke Dynamics Against Statistical Attacks

Valeriu - Daniel Stanciu
VU University Amsterdam
The Netherlands
valeriu.stanciu@cti.pub.ro

Riccardo Spolaor
University of Padua
Italy
rspolaor@math.unipd.it

Mauro Conti
University of Padua
Italy
conti@math.unipd.it

Cristiano Giuffrida
VU University Amsterdam
The Netherlands
giuffrida@cs.vu.nl

ABSTRACT

In recent years, simple password-based authentication systems have increasingly proven ineffective for many classes of real-world devices. As a result, many researchers have concentrated their efforts on the design of new biometric authentication systems. This trend has been further accelerated by the advent of mobile devices, which offer numerous sensors and capabilities to implement a variety of mobile biometric authentication systems. Along with the advances in biometric authentication, however, attacks have also become much more sophisticated and many biometric techniques have ultimately proven inadequate in face of advanced attackers in practice.

In this paper, we investigate the effectiveness of sensor-enhanced keystroke dynamics, a recent mobile biometric authentication mechanism that combines a particularly rich set of features. In our analysis, we consider different types of attacks, with a focus on advanced attacks that draw from general population statistics. Such attacks have already been proven effective in drastically reducing the accuracy of many state-of-the-art biometric authentication systems. We implemented a statistical attack against sensor-enhanced keystroke dynamics and evaluated its impact on detection accuracy. On one hand, our results show that sensor-enhanced keystroke dynamics are generally robust against statistical attacks with a marginal equal-error rate impact (<0.14%). On the other hand, our results show that, surprisingly, keystroke timing features non-trivially weaken the security guarantees provided by sensor features alone. Our findings suggest that *sensor dynamics* may be a stronger biometric authentication mechanism against recently proposed practical attacks.

CODASPY'16, March 09-11, 2016, New Orleans, LA, USA
© 2016 ACM. ISBN 978-1-4503-3935-3/16/03…$15.00
DOI: http://dx.doi.org/10.1145/2857705.2857748

1. INTRODUCTION

Password-based systems have been the most common form of authentication for many years. Albeit simple, password-based authentication is prone to several attacks such as guessing attacks, dictionary attacks, and shoulder surfing attacks. With the advent of mobile computing, such attacks have dramatically increased their effectiveness.

In a mobile context, password-based authentication is even more problematic due to the peculiar nature of the devices. Most of such problems relate to the users being unaware of privacy leakage or exhibiting incautious behavior. Indeed, users ofter choose passwords that are easy to type and thus easy to guess. They also tend to leave visible finger marks on the touchscreen (i.e., smudges), which can lead to password leaks [1]. Moreover, an unaware user could be victim of shoulder surfing attacks due to the exposure of mobile devices [30, 24]. In addition, new increasingly sophisticated attacks are rampant. For example, researchers have demonstrated the proneness of mobile password authentication to automated attacks based on videos captured by low-end cameras and motion analysis through repeated reflections [33]. As an another example, more advanced attacks that rely only on hands dynamics [28] can achieve even 50% penetration rate at the first attempt. Given the ever-growing amount of highly-sensitive information (e.g., credit card transactions, confidential emails, and personal photos) stored on mobile devices, there is an increasing demand for stronger mobile authentication techniques.

To address the emerging threats to password-based mobile authentication, many researchers have recently devised a number of biometric authentication systems [14] for mobile devices. Biometric authentication identifies users by relying on their behavioral or physiological characteristics, such as fingerprints, body geometry, voice, gestures or signature. Since modern mobile devices offer an increasingly complex and diverse set of user-driven features, they provide an ideal platform for capturing and analyzing biometric traits. Prior research efforts have explored a broad range of mobile authentication techniques, including authenticating users based on their walking patterns [17], phone call gestures [5], touch gestures [23, 6, 7, 18, 27] or keystroke dynamics [15, 4, 11, 31]. However, most existing biometric authentication mechanisms either yield insufficient accuracy for real-world deployment or have recently proven ineffective against advanced statistical attacks [26, 29, 25]. The recent sensor-enhanced keystroke dynamics [9], in turn, yields promising accuracy results against zero-effort attack conditions, but its real effectiveness in face of advanced statistical attacks remains unexplored.

Contributions

In this paper, we assess the effectiveness of sensor-enhanced keystroke dynamics against statistical attacks drawing input data from general population statistics. Sensor-enhanced keystroke dynamics relies on both mobile devices built-in sensors (e.g., accelerometer, gyroscope) and keystrokes timing (e.g., hold time) information for user authentication purposes. This rich set of features can be potentially generalized to different biometric authentication systems, thereby motivating our focus in this paper. In order to conduct our investigation, we implemented a state-of-the-art statistical attack that forges mobile user inputs based on real population statistics. Next, we ran the attack on a pool of 20 users using our forged input samples. We have assessed the effectiveness of our attack and its impact on the end-to-end accuracy of sensor-enhanced keystroke dynamics, also evaluating its effects on the individual features (i.e., sensor-based and timing-based). Our experimental results show that sensor-enhanced keystroke dynamics is resilient to both zero-effort and advanced statistical attacks in practice. Finally, our results suggest that timing features are generally detrimental to detection accuracy in statistical attack scenarios, but they still achieve a measurable accuracy improvement in face of zero-effort attacks.

Roadmap

The remainder of the paper is organized as follows. In Section 2, we survey related work, provide background information on sensor-enhanced keystroke dynamics, and present modern attacks against biometric authentication systems. In Section 3, we present the threat model, our attack methodology, and implementation. In Section 4, we present experimental results, and discuss our findings. In Section 5, finally, we draw conclusions and discuss directions for future work.

2. RELATED WORK

In the literature, there have been numerous attempts to attack biometric authentication systems. In the following, we first provide background information to adequately introduce the main concepts and then detail prior efforts related to our work.

2.1 Equal Error Rate

Measuring the accuracy of a biometric identification system is a well-studied problem [32]. Briefly, the performance of a system is characterized by two metrics:

- False Match Rate (FMR), which expresses the number of impostors accepted as legitimate users;
- False Nonmatch Rate (FNMR), which expresses the number of legitimate users rejected as impostors.

The parameters of an authentication system can be tuned according to its specific purpose, that is, facilitating authentication for the average user, but accepting more impostors (i.e., with a high FMR), or offering better security but constraining the user to a more strict authentication behavior (i.e., with a high FNMR).

In the literature, e.g., [7], [19], [12], the Equal Error Rate (EER) is a common accuracy metric. The EER reflects the accuracy of the authentication system when the False Match Rate and the False Nonmatch Rate are equal. It is expressed as a single value between 0 and 1, with lower values representing better accuracy. Following the common approach in the literature, we rely on the EER to measure accuracy and compare our results against prior work.

2.2 Sensor-enhanced Keystroke Dynamics

Keystroke dynamics have been widely used as biometrics for characterizing users' behavior. From hardware to software keyboards, from fixed- to free-text input, from traditional to mobile devices, keystroke dynamics has successfully exploited intrinsic typing characteristics for authenticating interactive users. In detail, keystroke dynamics relies on timing *key-press* and *key-release* events, associating them in different ways and deriving features such as the key hold time, inter-key-press time, etc.

With the advent of mobile devices, a fairly large number of sensors became available for biometric authentication purposes, such as accelerometer, gyroscope, temperature, air pressure, and many others. In this paper, we specifically focus on movement sensors, i.e., accelerometer and gyroscope, which have proven effective for authentication purposes in the context of *sensor-enhanced keystroke dynamics* [9].

Sensor-enhanced keystroke dynamics [9] augments keystroke timing features borrowed from traditional keystroke dynamics with sensor-based features derived from real-time data sampled from movement sensors. Building on this rich feature set, prior sensor-enhanced keystroke dynamics solutions implemented different detection algorithms, including the mean and k-nearest neighbors (kNN, with k=1) algorithms based on Euclidean (unweighted, weighted, normed, normed weighted) and Manhattan (unweighted, weighted, scaled, scaled weighted) distances. Such detection algorithms have been evaluated against zero-effort attacks, ultimately reporting a 0.08% EER [9].

2.3 Zero-effort Attacks

Zero-effort attacks [16, 14] are the common setting to evaluate the accuracy of biometric authentication systems and their resilience to attacks. In short, a zero-effort attack refers to an impostor generating inputs to bypass an authentication system without any knowledge of other users and their behavioral characteristics. For our evaluation, this translates to evaluating the accuracy of our authentication system for every user against any other user who participated in the experiment. This evaluation strategy has been widely used in prior work in the field [7, 12, 18].

As mentioned earlier, a zero-effort attack setting was also previously used to evaluate the accuracy of sensor-enhanced keystroke dynamics [9]. Nevertheless, accuracy measurements solely based on zero-effort attacks are not alone sufficient to thoroughly assess the robustness of biometric authentication systems against modern attacks [22]. In this paper, we thereby consider more sophisticated (statistical) attacks to thoroughly evaluate the accuracy of sensor-enhanced keystroke dynamics in practice.

2.4 Other Attacks

Many prior efforts have shown that zero-effort attacks achieve significantly lower EER than more sophisticated non-zero-effort attacks in practice [22, 2, 8, 13, 20, 21, 3, 26, 29, 25]. In what follows, we discuss the most relevant efforts for our work and draw appropriate comparison.

Rahman et al. [21] propose a snoop-forge-replay attack against a keystroke-based continuous verification system. Their technique is based on snooping keystroke samples from the user using a keylogger [10] and reuse the samples to evade detection. Their attack achieves extremely high EERs, between 43.0% and 96.5%, with average EER increases between 69.3% and 2,919.3%. Such high EER increases naturally stem from the attack setting, which relies on real keystrokes "stolen" from the victim to create the forged inputs. We assume that, in such an attack setting, the system has already been irremediably compromised and we construct our attack only based on external population statistics (i.e., without any prior knowledge about the victim).

Other researchers [26] have demonstrated practical statistical attacks against touch-based authentication systems. Such attacks rely on a "Lego" robot to physically issue forged inputs based on statistical analysis of the touch patterns of a population. The authors in [26] have reported EER increases between 339% and 1,004%, with mean EER values between 30% and 55%. Unlike such attacks, we consider an ideal (and thus pessimistic) setting with no physical constraints for the attacker, providing worst-case results for the accuracy of sensor-enhanced keystroke dynamics.

Other researchers have proposed statistical attacks to evaluate the accuracy of keystroke dynamics [29]. Such attacks, somewhat closer to our work, also rely on statistically drawn inputs issued by a physical robot. The attacks presented in [29], however, are more focused on raising concerns about synthetic forgery attacks rather than detailing the statistical attack design and impact in practice. Serwadda et al. [25] provide a step forward, with a detailed accuracy analysis of keystroke dynamics against statistical attacks based on timings drawn from general population statistics. The attack presented in [25] groups keystroke dynamics features in probability trees and relies on a binning procedure to extract statistically relevant features from a given population. Our attack draws from the methodology presented in [25], but focuses on sensor-enhanced keystroke dynamics and scales to a much higher number of features and inputs.

3. ATTACK METHODOLOGY

In this section, we present the methodology we followed to perform the attack. We first introduce the threat model (Section 3.1), then give details about feature selection (Section 3.2) and detection (Section 3.3) algorithms, and finally discuss how we implemented the attack (Section 3.4).

3.1 Threat Model

We consider an attacker seeking to bypass a sensor-enhanced keystroke dynamics authentication system based on password-based keystroke dynamics combined with accelerometer and gyroscope sensor information. If the attacker enters the correct password with the expected typing behavior, then authentication is successful. Otherwise, authentication fails, even if the typed password is correct.

We assume the password to be known to the attacker. This is a common assumption in biometric authentication, since the main purpose of the accuracy analysis is to assess the extra security added by biometric characteristics.

We also assume an ideal attacker able to perfectly forge a particular typing pattern. In other words, we assume the attacker to be a "perfect machine" able to learn and reproduce human typing patterns. In particular, in order to obtain worst-case accuracy results, we assume the attacker is not limited by any physical constraints. This model matches the assumptions made in prior sensor-enhanced keystroke dynamics efforts [9].

Finally, we assume the attacker can draw from general population statistics (except the victim) to generate impostor samples and try to bypass the authentication system. This is a common assumption in statistical attack scenarios [25], which normally impose no restrictions on the population data available to the attacker.

3.2 Feature Selection

Sensor-enhanced keystroke dynamics relies on features derived from keystroke timings and sensor data continuously sampled while typing. In particular, the features we select from the sensor distributions sampled over time are based on standard statistical metrics, that is root mean square, minimum and maximum value, number of local maxima and minima, mean delta, sum of positive values, sum of negative values, mean value, mean value during keystroke events, and standard deviation. We rely on these statistical metrics to identify features for both gyroscope and accelerometer samples. For keystroke dynamics, we rely on two main features: *KHT* (*Key Hold Time*) and *KIT* (*Key Interval Time*, or inter-key-press time), for each key and pair of keys typed, respectively. Each sample is represented as a vector of F feature values, where F is the total number of features according to the distance metrics considered.

3.3 Detection

To authenticate users and detect impostors, we rely on standard threshold-based binary classification. Similar to prior efforts [9], we experimented with two classes of detection algorithms, k-nearest neighbors (kNN, with k = 1), and the mean algorithm (comparing the test samples against the mean training sample rather than against all the training samples). We omitted the SVM and Naive Bayes algorithms from our analysis, after unsuccessful early attempts. Finally, we based our detection algorithms on a number of distance metrics commonly used in the literature [9]: Euclidean, Euclidean normed, Manhattan, Manhattan scaled, and their weighted versions (weights representing the importance of a given feature in the feature vectors).

3.4 Attack Design and Implementation

This section presents the design and implementation of our statistical attack and details the challenges involved. Figure 1 illustrates a general overview of our attack strategy.

Our statistical attack strategy consists of four main steps:

- Statistically analyze the samples of a given population to identify common characteristics, that is the most frequent combinations of feature values.
- Generate forged input samples based on the previously discovered characteristics.
- Feed the forged samples to the authentication system.
- Analyze the impact of the attack.

To analyze the impact of the attack, we rely on leave-one-out cross-validation and EER calculations under statistical attacks, as follows:

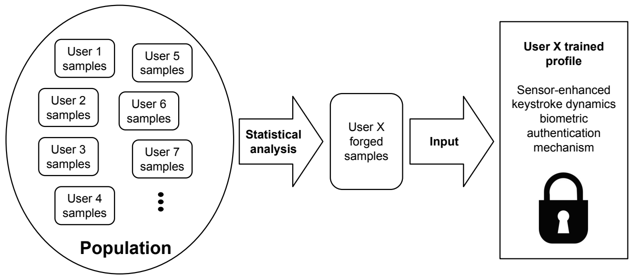


Figure 1: Statistical attack overview.

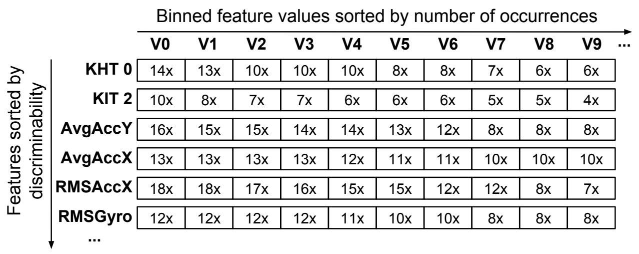
- For each user in the population, we iteratively "leave one sample out" and create a *training set* containing the remaining samples.
- Unlike zero-effort attacks, we create a *testing set* for each user with inputs statistically generated using the common characteristics from other users' samples.
- For each iteration, we calculate the corresponding EER and acceptance threshold for the given user (based on the reported False Match and False Nonmatch rates) and evaluate the effectiveness of our attack.

To identify common characteristics in the population and generate forged input samples based on statistical analysis, we have devised the procedure depicted in Algorithm 1. The algorithm receives the following parameters in input: an array of real-world input samples ($realInputs$), an array of weights ($weights$) for each feature (calculated across the population by an SVM classifier), the number of bins ($numberOfBins$) each $[0; 1]$ feature value distribution should be partitioned into, and the number of most relevant bins (h) to consider to forge input samples.

```
Input: realInputs[ ]; //Real-world input samples
Input: weights[ ]; //Feature weights
Input: numberOfBins; //Number of bins for each feature
Input: h; //Number of bins for forged input generation
Output: forgedInputs[ ]; //Forged input samples

/* Generate bins sorted by feature weights and occurrences */
binnedInputs[ ] := binInputs(realInputs);
sortBinsByOccurrence(binnedInputs);
sortFeaturesByWeight(binnedInputs, weights);

/* Initialize bin indexes to 0 */
binIndexes[ ] := initBinIndexes();

foreach realInputs do
    forgedInput := generateInput(binnedInputs, binIndexes, h);
    addForgedInput(forgedInput, forgedInputs);
end

return forgedInputs;
```

Algorithm 1: Procedure to create forged input samples.

Given the input parameters, the algorithm runs a binning procedure that iterates over all the samples and feature values received in input and, for each feature value visited, increments the corresponding bin. The bins uniformly partition the original $[0; 1]$ feature value intervals in $numberOfBins$ smaller (and identical) value intervals. After running this procedure, the algorithm obtains, for each feature, an array of bins sorted by number of occurrences in descending order (see Figure 2). In other words, for each feature, the first N bins determine the N most common feature values across the population. In Algorithm 1, all the sorted per-feature bins are ultimately assembled and sorted by feature weights in descending order in the $binnedInputs$ array. Next, the algorithm initializes the $binIndexes$ array containing, for each feature, the bin to use to generate the next guess. Initially, all the indexes are set to 0.


Binned feature values sorted by number of occurrences
Features sorted by discriminability

	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	...
KHT 0	14x	13x	10x	10x	10x	8x	8x	7x	6x	6x	
KIT 2	10x	8x	7x	7x	6x	6x	6x	5x	5x	4x	
AvgAccY	16x	15x	15x	14x	14x	13x	12x	8x	8x	8x	
AvgAccX	13x	13x	13x	13x	12x	11x	11x	10x	10x	10x	
RMSAccX	18x	18x	17x	16x	15x	15x	12x	12x	8x	7x	
RMSGyro	12x	12x	12x	12x	11x	10x	10x	8x	8x	8x	
...											

Figure 2: Bin generation example.

The next step is to identify the relevant bins for each feature and generate forged input samples. To identify the relevant per-feature bins, we simply select the first h bins with most occurrences. To generate the forged input samples, we traverse all the features available and generate combinations of features values associated only to the first h per-feature bins. A naive strategy, which instead considers all the possible per-feature bins without discrimination, would face a prohibitively large guess space and have trouble scaling. In Algorithm 1, our generation strategy is implemented by the $generateInput$ subprocedure, which runs $realInputs$ times to generate as many forged input samples as real-world samples given in input.

The order in which the combinations are generated is dictated by the discriminability offered by the our feature weights (calculated by an SVM classifier). The features with higher discriminability are considered first, thanks to the sorting strategy adopted when generating the $binnedInputs$ array. As an example, suppose we have only three features, KHT0 (key hold time for keystroke 0), KIT2 (time between keystroke 2 and keystroke 3), and AvgAccY (average acceleration on Y axis), as represented in Figure 2. Also suppose the value of h is set to 2, AvgAccY has the highest discriminability, and KHT0 has the lowest one. In this example, the first five combinations generated by the algorithm would be {V0, V0, V0}, {V0, V0, V1}, {V0, V1, V0}, {V0, V1, V1}, and {V1, V0, V0}.

Algorithm 2 presents the $generateInput$ subprocedure, which receives our $binnedInputs$, $binnedIndexes$, and h parameters in input, and each time returns a forged input sample ($forgedInput$) in output. First, the subprocedure iterates through the features represented in the $binnedInputs$ array. For each feature, the subprocedure relies on its bin index to locate the corresponding relevant feature value in the $binnedInputs$ array and add it to the forged input sample. Next, the subprocedure updates the $binIndexes$ array to select a new combination of feature values at the next subprocedure call. This is done by using a global counter ($guessCounter$), which, at each invocation of the subprocedure, is incremented and converted into base h to encode the next combination in the $binIndexes$ array. This allows our search strategy to potentially explore all the possible $h^{\#features}$ combinations, while always prioritizing the most discriminative features thanks to the sorting strategy adopted when generating the $binnedInputs$ array. Finally, the subprocedure returns the generated forged input sample including all the necessary feature values.

```
Input: binnedInputs[ ]; //Binned input samples
Input: binIndexes[ ]; //Current indexes
Input: h; //Maximum number of bins to use for each feature
Output: forgedInput; //Forged input sample

/* Initialize an empty forged input sample */
forgedInput[ ] := initForgedInput();

foreach binnedInputs as binnedInput do
    /* Retrieve current feature index */
    i = getFeatureIndex(binnedInput);

    /* Get corresponding relevant feature value */
    v = binIndexes[i];

    /* Add current feature value to forget input sample */
    addFeatureValue(v, forgedInput);

end

/* Update bin indexes */
guessCounter++; //Global variable
currentGuess = guessCounter;

foreach binIndexes as binIndex do
    binIndex = currentGuess % h;
    currentGuess = currentGuess / h;
end

return forgedInput;
```

Algorithm 2: Forged input generation subprocedure.

4. EVALUATION

We evaluated our statistical attack against all the detectors considered, that is, combinations of detection algorithms and distance metrics. Moreover, we experimented with all the combinations of features, namely keystrokes-only (i.e., keystroke dynamics), sensors-only (i.e., sensor dynamics), and keystrokes and sensors (i.e., sensor-enhanced keystroke dynamics).

To reproduce the experimental setting from prior work [9], we based our analysis on a dataset generated by 20 different test subjects (students in our department). The test subjects were asked to type 20 different samples of 2 passwords negotiated with our test subjects in advance, i.e., "internet" and "satellite" (easy to type in a mobile setting, yet sufficiently long to provide adequate protection). We gathered keystroke and sensor samples in a controlled environment, on a Samsung Nexus S with a soft keyboard in landscape mode. We sampled sensor value distributions at a frequency of 17 Hz. To obtain targeted results, we focused our attack and accuracy analysis on whole words rather than n-graphs.

For each detector, we ran the experiments while varying the number of bins between 25 and 300, with a step size of 25 (using less than 25 bins would have severely lowered accuracy in statistics, while using more than 300 bins would have generated overly specific results, given that we had about 350 samples to statistically group). To highlight the cases in which the attack was most damaging, we first report only the highest EERs across all the bin configurations. We also experimented with different values of h, but we ultimately resorted to $h = 3$ for all the configurations. Given the large feature space, increasing h (beyond 3) did not significantly affect the results. Table 1 summarizes our results.

We first mounted our attack against our keystroke dynamics configuration. As shown in Figure 4a, our experiment reported EER increases between 210.09% (using the "mean, euclidean normed weighted" combination) and 452.92% (using the "kNN (k=1), euclidean normed" combination)—i.e., mean EER values between 31.47% and 50.56% for the input combination with the highest impact. In contrast, Serwadda et al. [25] reported EER increases of between 28.6% and 84.4% for statistical attacks against keystroke dynamics.

Table 1: EER increases and ranges under our attack.

	Min. EER increase	**Max. EER increase**	**Min. EER**	**Max. EER**
Keystroke Dynamics	210.09%	452.92%	31.47%	50.56%
Sensor Dynamics	28.29%	1368.14%	0.22%	3.89%
Sensor-Enhanced Keystroke Dynamics	148.1%	3143.36%	0.22%	14.91%

We believe the discrepancy with previous results [25] to be caused by the following reasons:

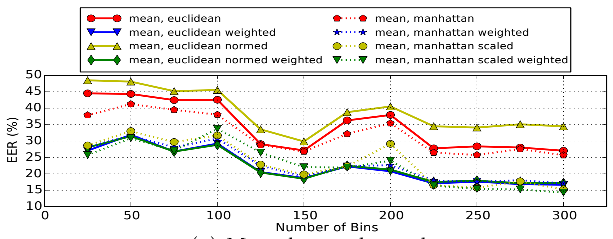
1. We relied only on the most probable combinations generated by our algorithms, while Serwadda et al. [25] also used forged inputs randomly sampled throughout the entire guess space.
2. For each user, we generated a much larger number of forged input samples (around 350, rather than 50 [25]).
3. We considered a different environment, focusing on mobile rather than traditional [25] devices.

Despite the discrepancies, our results confirm and even provide stronger evidence that keystroke dynamics is vulnerable to statistical attacks. Next, we mounted our attack against our sensor-based configurations. Our results show that both configurations, that is sensors dynamics and sensor-enhanced keystroke dynamics, provide strong resistance against statistical attacks. While we did observe EER increases due to statistical attacks in our experiments, such increases were small, resulting in fairly limited impact in practice as detailed below.

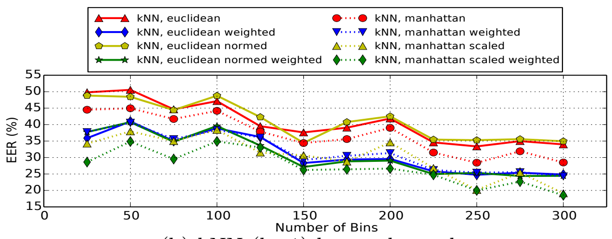
In the sensor-only configuration (see Figure 4b), we reported EER increases between 28.29% and 1368.14% and EER values between 0.22% and 3.89% ($numberOfBins = 200$, $h = 3$). Our best performer, the "kNN (k=1), euclidean weighted" combination, reported EERs of 0.22% under our statistical attack and 0.09% under zero-effort attacks, confirming the sensor-only configuration to be robust against statistical attacks and appropriate for adoption.

In the sensor-enhanced keystroke dynamics configuration (see Figure 4c), the smallest EER under our statistical attack was the same as in the previous sensors-only configuration (i.e., 0.22% for "kNN (k=1), euclidean weighted"). However, other detectors reported much higher EER increases than in the sensors-only configuration. As an example, for "kNN (k=1), euclidean", the EER increase reaches 14.92% from the original 0.46% under zero-effort attacks. This suggests that adding keystroke timing features to even robust sensor features can significantly degrade the performance of some (but not all) detectors under statistical attacks.

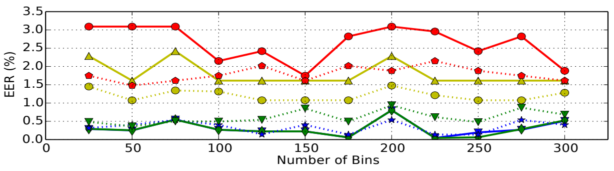
To validate our results, we tried several more combinations of values for h and $numberOfBins$. Significantly increasing h results in the most important features being consistently assigned the same feature values for all the forged input samples, potentially biasing the results. When gradu-



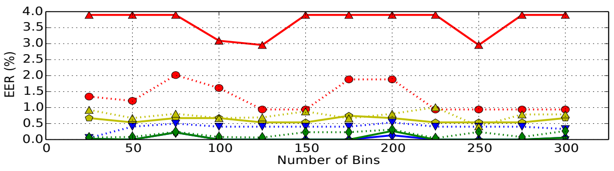
(a) Mean keystrokes only.



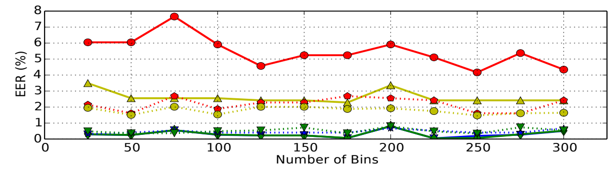
(b) kNN (k=1) keystrokes only.



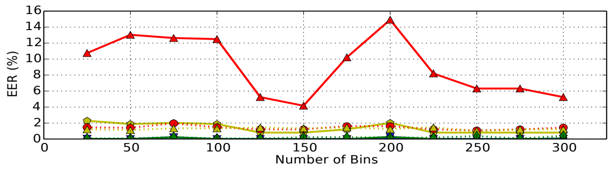
(c) Mean sensors only.



(d) kNN (k=1) sensors only.

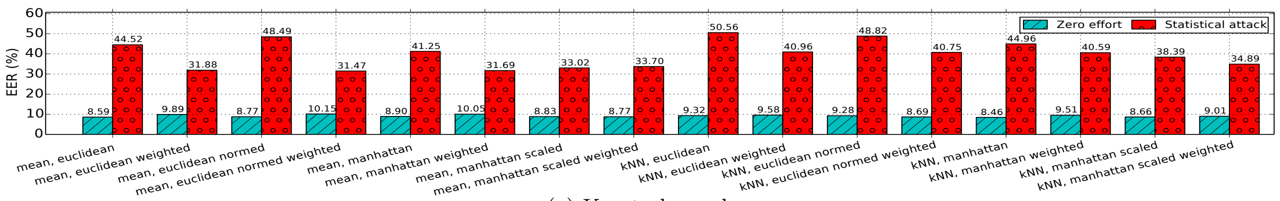


(e) Mean keystrokes and sensors.

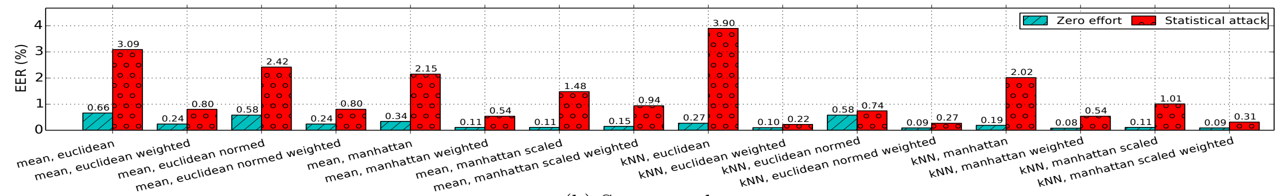


(f) kNN (k=1) keystrokes and sensors.

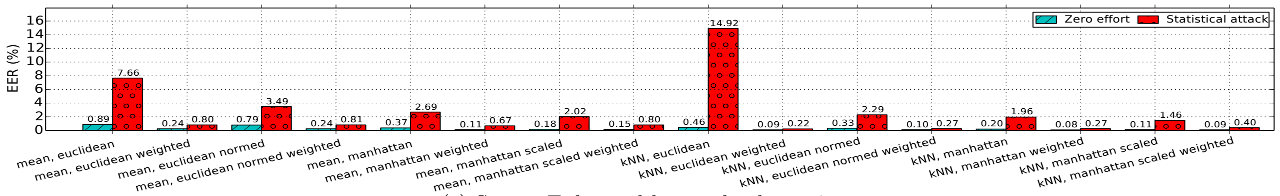
Figure 3: EER rate vs. number of bins used in our statistical attack.



(a) Keystrokes only.



(b) Sensors only.



(c) Sensor-Enhanced keystroke dynamics.

Figure 4: Comparison between EERs obtained by zero-effort attacks and our statistical attack.

ally increasing h (up to 300), we observed EER variations of 5-10% (compared to our previous results) for our keystroke dynamics configuration. However, we observed no noticeable EER variations for our sensors-only or sensor-enhanced keystroke dynamics configurations.

When gradually increasing the $numberOfBins$ (between 25 and 300, with increments of 25), in turn, we observed that the optimal number of bins depends on the number of features and on the detector used. In particular, in our keystroke dynamics configuration, we obtained optimal results for a small number of bins (less than 100). In our sensor-based configurations, in contrast, we obtained optimal results for a much larger number of bins (higher than 200). We believe this behavior to stem from the different number and discrimination power of the features used. For example, keystroke dynamics has more limited features, so the calculated thresholds are generally wider. As a result, having wider bins (smaller number of bins) can provide better spread for the attack. With 25 bins (0.04 interval size per bin), traversing the features for $h = 3$ covers 12% of the most probable values. On the other hand, having 70-80 different unique features, with more discriminative power, provides tighter thresholds. Hence, it is more complicated for a forged input sample based on wide bins to fit into the thresholds (i.e., succesfully authenticate), translating to the need for more than 200 tighter bins.

Another finding that emerges from Figure 3 is that, on average, the attack is more effective against mean-based rather than kNN-based detectors. This means that, on average, kNN-based detectors are more resistant against statistical attacks. In addition, in most of the cases, weighted detectors offer better accuracy than unweighted detectors in face of statistical attacks. Weighted detectors also tend to provide better results for a large number of bins. We believe this is due to weighted detectors being able to achieve tighter thresholds and better raise the bar for the attacker, similar to the behavior observed earlier for sensor-based features.

5. CONCLUSION

In this paper, we analyzed the behavior of state-of-the-art biometric authentication systems based on sensor-enhanced keystroke dynamics under statistical attacks. Our goal was to establish whether the combination of keystroke timings and mobile sensor-based features offers a sufficiently robust authentication mechanism against sophisticated attacks.

For our purposes, we designed and implemented a statistical attack against sensor-enhanced keystroke dynamics. Our approach is to forge statistically relevant inputs by drawing from the characteristics of a given population and attempt to evade detection. We attacked sensor-enhanced keystroke dynamics for all the three combinations of features used in prior work [9]. Our results confirm that basic keystroke-dynamics authentication is very prone to statistical attacks, with the best classifier available yielding an EER of 28.83% and an EER increase of 184% compared to the zero-effort attack. When sensors are considered, in turn, we obtained much more promising results. The best classifier reported an EER of 0.22% for both sensors-only and sensor-enhanced keystroke dynamics, with EER increases of 123.28% and 148.09% (respectively). The effective percentage increase in these cases is 0.12-0.13%. Our results show that the effectiveness of statistical attacks against these two mechanisms is low, demonstrating their robustness in practice. Moreover, our results suggest that sensor dynamics alone is a stronger mobile biometric authentication mechanism against statistical attacks, since it proved robust for all the classifiers we considered (not only the weighted ones) while full sensor-enhanced keystroke dynamics performed poorly for unweighted classifiers.

To conclude, we have shown that, by using sensor-based biometric features, it is possible to build highly accurate mobile authentication systems, robust against both human and modern statistical attacks.

6. ACKNOWLEDGMENTS

Mauro Conti is supported by a Marie Curie Fellowship (PCIG11-GA-2012-321980). This work is also supported by the EU TagItSmart! project (H2020-ICT30-2015-688061), the EU-India REACH project (ICI+/2014/342-896), the Italian PRIN TENACE project (20103P34XC), the Univ. Padua PRAT-2013 project on malware detection, and by the ReCover project funded by NWO.

7. REFERENCES

[1] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. 2010.

[2] B. Biggio, Z. Akhtar, G. Fumera, G. L. Marcialis, and F. Roli. Security evaluation of biometric authentication systems under real spoofing attacks. *IET biometrics*, 1(1):11–24, 2012.

[3] I. Chingovska, A. Rabello dos Anjos, and S. Marcel. Biometrics evaluation under spoofing attacks. *IEEE TIFS*, 9(12):2264–2276, 2014.

[4] N. L. Clarke and S. Furnell. Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security*, 6(1):1–14, 2007.

[5] M. Conti, I. Zachia-Zlatea, and B. Crispo. Mind how you answer me!: transparently authenticating the user of a smartphone when answering or placing a call. In *Proceedings of ACM ASIACCS*, 2011.

[6] 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 *Proceedings of ACM CHI*, 2012.

[7] M. Frank, R. Biedert, E.-D. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE TIFS*, 8(1):136, 2013.

[8] D. Gafurov, E. Snekkenes, and P. Bours. Spoof attacks on gait authentication system. *IEEE TIFS*, 2(3):491–502, 2007.

[9] C. Giuffrida, K. Majdanik, M. Conti, and H. Bos. I sensed it was you: authenticating mobile users with sensor-enhanced keystroke dynamics. In *Proceedings of DIMVA*, 2014.

[10] C. Giuffrida, S. Ortolani, and B. Crispo. Memoirs of a browser: A cross-browser detection model for privacy-breaching extensions. In *Proceedings of ACM ASIACCS*, 2012.

[11] X. Huang, G. Lund, and A. Sapeluk. Development of a typing behaviour recognition mechanism on android. In *Proceedings of IEEE TrustCom*, 2012.

[12] S.-s. Hwang, S. Cho, and S. Park. Keystroke dynamics-based authentication for mobile devices. *Computers & Security*, 28(1):85–93, 2009.

[13] A. K. Jain and K. Nandakumar. Biometric authentication: System security and user privacy. *IEEE Computer*, 45(11):87–92, 2012.

[14] A. K. Jain, A. Ross, and S. Pankanti. Biometrics: a tool for information security. *IEEE TIFS*, 1(2):125–143, 2006.

[15] E. Maiorana, P. Campisi, N. González-Carballo, and A. Neri. Keystroke dynamics authentication for mobile phones. In *Proceedings of ACM SAC*, 2011.

[16] A. J. Mansfield and J. L. Wayman. *Best practices in testing and reporting performance of biometric devices*. Centre for Mathematics and Scientific Computing, NPL, 2002.

[17] J. Mäntyjärvi, M. Lindholm, E. Vildjiounaite, S.-M. Mäkelä, and H. Ailisto. Identifying users of portable devices from gait pattern with accelerometers. In *Proceedings of IEEE ICASSP*, 2005.

[18] Y. Meng, D. S. Wong, R. Schlegel, et al. Touch gestures based biometric authentication scheme for touchscreen mobile phones. In *Information Security and Cryptology*, pages 331–350, 2013.

[19] F. Okumura, A. Kubota, Y. Hatori, K. Matsuo, M. Hashimoto, and A. Koike. A study on biometric authentication based on arm sweep action with acceleration sensor. In *Proceedings of IEEE ISPACS*, 2006.

[20] K. Rahman, K. S. Balagani, V. V. Phoha, et al. Making impostor pass rates meaningless: A case of snoop-forge-replay attack on continuous cyber-behavioral verification with keystrokes. In *Proceedings of IEEE CVPRW*, 2011.

[21] K. A. Rahman, K. S. Balagani, and V. V. Phoha. Snoop-forge-replay attacks on continuous verification with keystrokes. *IEEE TIFS*, 8(3):528–541, 2013.

[22] A. Rattani and N. Poh. Biometric system design under zero and non-zero effort attacks. In *Proceedings of IEEE ICB*, 2013.

[23] 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 CHI*, 2012.

[24] F. Schaub, R. Deyhle, and M. Weber. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proceedings of ACM MUM*, 2012.

[25] A. Serwadda and V. V. Phoha. Examining a large keystroke biometrics dataset for statistical-attack openings. *ACM TISSEC*, 16(2):8, 2013.

[26] A. Serwadda and V. V. Phoha. When kids' toys breach mobile phone security. In *Proceedings of ACM CCS*, 2013.

[27] 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.

[28] D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha. Beware, your hands reveal your secrets! In *Proceedings of ACM CCS*, 2014.

[29] D. Stefan, X. Shu, and D. D. Yao. Robustness of keystroke-dynamics based biometrics against synthetic forgeries. *Computers & Security*, 31(1):109–121, 2012.

[30] F. Tari, A. Ozok, and S. H. Holden. A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In *Proceedings of ACM SOUPS*, 2006.

[31] C.-J. Tasia, T.-Y. Chang, P.-C. Cheng, and J.-H. Lin. Two novel biometric features in keystroke dynamics authentication systems for touch screen devices. *Security and Comm. Networks*, 7(4):750–758, 2014.

[32] J. L. Wayman. Error rate equations for the general biometric system. *Robotics & Automation Magazine*, 6(1):35–48, 1999.

[33] Y. Xu, J. Heinly, A. M. White, F. Monrose, and J.-M. Frahm. Seeing double: Reconstructing obscured typed input from repeated compromising reflections. In *Proceedings of ACM CCS*, 2013.