Jekyll and Hyde: On The Double-Faced Nature of Smart-Phone Sensor Noise Injection

Richard Matovu[1], Abdul Serwadda[2], David Irakiza[3], Isaac Griswold-Steiner[4]

Abstract: To combat privacy attacks that exploit the motion and orientation sensors embedded in mobile devices, a number of recent works have proposed noise injection schemes that degrade the quality of sensor data. Much as these schemes have been shown to thwart the attacks, the impact of noise injection on continuous authentication schemes proposed for mobile and wearable devices has never been studied. In this paper, we empirically tackle this question based on two widely studied continuous authentication applications (i.e., gait and handwriting authentication). Through a series of machine learning and statistical techniques, we show that the thresholds of noise needed to overcome the attacks would significantly degrade the performance of the continuous authentication applications. The paper argues against noise injection as a defense against attacks that exploit motion and orientation sensor data on mobile and wearable devices.

Keywords: continuous authentication, gait authentication, handwriting authentication, wearables and mobile phones.

1 Introduction

To authenticate the user after the initial login action, there are a myriad continuous authentication mechanisms recently proposed for mobile and wearable devices. Examples of these mechanisms include those centered on gait [Pr14], touch [Fr13], phone grasp [Si16] and handwriting patterns [GSMS17], to mention but a few. The core driving force behind these authentication mechanisms are the sensors inbuilt in the devices. For example, gait, touch and phone grasp based authentication relies on patterns captured by the accelerometer and gyroscope sensors built in the devices, while touch patterns are recorded thanks to the touch sensor.

Continuous access to these sensor data streams — as required by continuous authentication applications — could however provide an avenue for privacy violations. For example, several studies have shown that data from the accelerometer and gyroscope could be leveraged to infer a user's typed inputs (e.g., see example of this attack based on a smart-phone [Mi12] and on a smart-watch [WLRC15]). In another attack that applies to these two sensors as well as other sensors on mobile and wearable devices, it has been shown that sensor output could be leveraged to fingerprint sensor-equiped devices and maliciously track these devices whenever they access a given website (e.g., see [DBC16, Da16]). In the rest of the

[1] Texas Tech University, Lubbock, TX, USA, richard.matovu@ttu.edu
[2] Texas Tech University, Lubbock, TX, USA, abdul.serwadda@ttu.edu
[3] Louisiana Tech University, Ruston, LA, USA, dir003@latech.edu
[4] Texas Tech University, Lubbock, TX, USA, isaac.griswold-steiner@ttu.edu

paper, we refer to the former attack as Attack #1 and the latter as Attack #2 (Figure 1 gives a high level overview of these attacks).

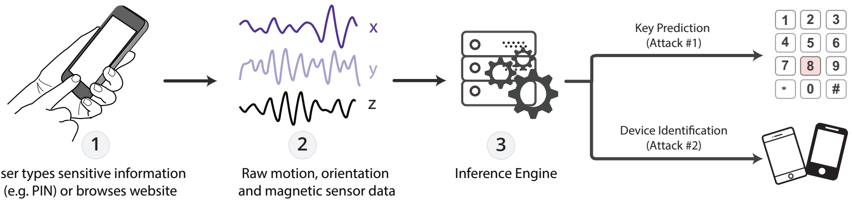


Fig. 1: High level overview of Attack #1 and Attack #2. Attack #1 uses data collected from sensors in a smart-phone or wearable device such as a smart-watch to predict the user's inputs. Attack #2 on the other hand uses the same data to fingerprint the identity of the device (i.e., to identify the device).

To ensure that a sensor-oriented application (such as continuous authentication application) can never feasibly exploit sensor data for these kinds of attacks, recent research has called for the injection of noise into the sensor data stream in order to degrade its quality before the sensor data can be accessed by the applications (e.g., see [DBC16, Da16, SMS16]). While these noise injection schemes have been shown to successfully overrun both Attacks #1 and #2, there has never been research on how continuous biometric authentication applications might perform given a sensor input stream as input that has been degraded through noise injection. This paper explores this question. In particular, we take the case of two widely studied continuous authentication applications (i.e., sensor-oriented gait and handwriting authentication) and empirically explore their behavior under one of the noise injection schemes proposed in recent research.

Our contributions: The main findings and contributions of our work are summarized below:

1. *Impact of Noise Injection on Authentication Error Rates*: At increasing thresholds of noise injection, we studied the behavior of gait and handwriting-based continuous authentication. We found that the injection of moderate amounts of noise causes the two applications to see statistically significant reductions in authentication accuracy relative to when no noise was injected. When we compared the authentication applications to each other under a wide range of noise injection thresholds, we found statistically significant differences in the impacts of noise on them. This observation indicates that the impacts of noise injection are highly application-dependent, which in turn implies that noise injection will have to be studied for many categories of apps before making reliable conclusions about its impacts.
2. *Exploring User-level Dynamics of Noise Injection based on the Biometric Menagerie*: Beyond the global authentication accuracy of the biometric system, we delved into the user-level dynamics of noise injection to understand how noise injection affects different categories of users. We found that the decrement in global mean F-score seen due to noise injection mostly manifested as an increment in one of the two classes of poor performing users: the ”lambs”. This observation is particularly interesting as it suggests that solutions targeting categories of users (e.g., as stipulated

by the biometric menagerie) might enable noise injection to co-exist with sensor-oriented continuous authentication in mobile and wearable gadgets.

Paper structure: The rest of the paper is organized as follows. We present related work in Section 2 followed by our experimental design in Section 3. We discuss our results in Section 4 and finally present our conclusions in Section 5.

2 RELATED WORK

There is now a sizable body of work proposing the injection of noise into the sensor stream in order to make it impossible for sensor-oriented applications to violate the privacy of the user. However, none of these works has explored how biometric continuous authentication applications might be affected by the noise. In [Ow12], it was shown that accelerometer measurements could be used by a sensor-oriented application to decode a 6-character password in an average of about 4.5 guesses (i.e., Attack #1). As a mitigation strategy against the attack, they suggested the use of vibrational noise to perturb the accelerometer sensor data stream. Arguing that vibrational noise is not sufficient to obfuscate the keystrokes, Shrestha *et al.* [SMS16] proposed a defensive application (called *Slogger*) that defended the attacks through more aggressive noise injection. The application injects programmatic noise at random intervals between touchstroke events (sometimes replacing some or all events with noise values) when a user is inputting sensitive information. *Slogger* was shown to make it difficult for an attacker to distinguish between noisy and actual sensor values.

With regard to Attack #2, the works in [DBC16, Da16] explored the impact of the attacks, and showcased several noise injection schemes that were able to thwart the attacks. To support the argument that noise injection would not have a negative impact on benign sensor-oriented applications, they studied the impact of noise on a step counting app in [DBC16] and the impact of noise on a gaming app in [Da16]. In both cases they concluded that these two apps were not significantly affected by noise. As we show in our work however, different apps can be affected very differently by noise injection. This is likely due, at least in part, to the differences in underlying operating mechanisms of the apps.

For example, many step counting apps operate by keeping track of the number of cycles in the sensor time series. A biometric authentication application on the other hand would have to do more than just count cycles — e.g., for gait authentication, the application, depending on the features used, might have to additionally distinguish between the nature of these cycles in order to separate between two users. For this reason, if the step counting application is not affected by noise, this does not necessarily imply that the gait authentication application is also not affected by noise. As the body of work fronting noise injection continues to grow, there has never been a study exploring the impact of noise on behavioral biometric authentication. It is this focus on *behavioral biometric authentication* that separates our work from all past research on this problem.

3 EXPERIMENTAL DESIGN

The general flow behind our methodology is as follows: (1) We implement the authentication applications (i.e., gait and handwriting authentication), and (2) We implement the noise injection mechanism and concurrently study its impact on both the authentication applications. Below, we briefly describe the implementation of our applications and their associated data collection experiments before presenting the performance evaluation in Section 4. All the applications used during our research were covered under the same Institutional Review Board approval.

3.1 Implementing the Authentication Applications

Gait authentication: We implemented a sensor-oriented application which collects gyroscope and accelerometer sensor data. After installing this app on Samsung Galaxy S6 phone, we had participants use the app for gait authentication experiment which involved users walking along a corridor (in order for us to monitor their gait). The experiment involved 21 users who participated over two sessions that were at least 1 day apart. These users placed the phone in their front-left pocket while they undertook the experiment.

Handwriting authentication: For this application, we implemented an Android smartwatch application that captures accelerometer and gyroscope sensor data on an LG Urbane smart-watch. The application captures a user's hand movement pattern during writing. Again, data was collected from 21 subjects who wore the watch as they wrote text from a randomly assigned document. Each subject participated in 2 different writing experiments that were conducted on two different days.

Data processing and Machine Learning frameworks for the 2 authentication apps: Having collected data for the two authentication applications, the core application logic was implemented offline as we applied Machine Learning algorithms to the collected data. The accelerometer and gyroscope data was collected in the form of $\{t_s, x, y, z\}$ where t_s is the timestamp of that particular sensor value and x, y, z are the sensor values along the three axes x, y, and z respectively. At each timestamp, we then computed the magnitude $m = \sqrt{(x^2 + y^2 + z^2)}$.

To smooth the time series for each of the x, y, z and m components, we computed a simple moving average based on a window of 3 consecutive sensor readings. The time series obtained from this step was then broken into sliding windows with a 50% overlap between consecutive windows. Gait authentication used 10-second windows while handwriting authentication used 15-second windows. This difference in window size was a result of tuning the applications to a setting which gave the best performance. From each window we extracted several features motivated by feature-sets reported in well-performing systems in previous research. In particular, we used the features reported in [Pr14] for gait authentication and the features used in [GSMS17] for handwriting authentication. Due to space limitations, we do not provide the detailed feature listings in this paper.

After feature extraction, we normalized these features using the min-max scheme to a range between 0 and 1, and performed classification using python scikit-learn framework. We tried out several classification algorithms to assess how they would perform with our data. The Support Vector Machine (SVM — with a polynomial kernel and C value of 1000), and Logistic Regression (LR) were selected after producing the best performance. Data from each user's first session was used for training while data from the second session was used for testing. The ratio of genuine to imposter samples was 1 to 2 for both training and testing datasets. The instances of impostors were randomly selected.

3.2 Implementation of Noise Injection-Based Defenses

Past studies have implemented several variants of noise injection. For example in [DBC16] [Da16], noise exhibiting three different probability distributions (i.e. uniform, laplace and gaussian distributions) was used. In [SMS16] on the other hand, only uniformly distributed noise was used. For purposes of signal obfuscation, uniformly distributed noise is in general more robust than other noise distributions since it offers the highest entropy over a given bound, making it the most challenging form of noise for any attack which seeks to reconstruct the original signal.

Like was done in [SMS16], our designs are hence based on uniformly distributed noise due to its strong defensive credentials. Our noise was drawn from a uniform distribution whose maximum and minimum accelerometer and gyroscope values were obtained from the minimum and maximum sensor values registered during Attacks #1 and #2. We refer to these minimum and maximum values as the base range that we later modify when studying the attack under varying settings. Uniform noise was injected at a random interval between 3 and 8 milliseconds.

Algorithm 1 summaries our implementation of noise injection. Like previous works (e.g. [DBC16]), our noise injection is offline — i.e., noise is injected into already collected sensor data. This approach is convenient for repetitive experiments such as ours, yet simulates the behavior of a sensor producing data that has been perturbed by noise. The algorithm inputs a stream of sensor data ω_{sensor}, its start timestamp ts_{start}, its end timestamp ts_{end}, the lower and upper bound of the time interval for noise injection $range_{ts}$, and the lower and upper noise bound $range_{noise}$; and outputs ω_{noisy_sensor} , a modified stream of sensor and noise data. The $Random()$ function draws a value randomly from uniform distribution within a given range while $MergeAndSort()$ merges the two given sensor data streams and sorts them according to dates. Lines #8 - #12 are used to generate a timestamped series of noise data ω_{noisy_data} that is later merged and sorted with the original sensor data according to the new timestamps. Line #15 produces the obfuscated sensor data, ω_{noisy_sensor} , which is used in our experimental results in the next section.

ALGORITHM 1: Noise Injection

```
Input: $\omega_{sensor}$ , $ts_{start}$, $ts_{end}$, $range_{ts}$ , $range_{noise}$
// $\omega_{sensor}$: Sensor data
// $ts_{start}$: Start timestamp of sensor data
// $ts_{end}$: End timestamp of sensor data
// $range_{ts}$: Lower and upper bound of time interval
// $range_{noise}$: Lower and upper noise bound
Output: $\omega_{noisy\_sensor}$
// Modified noisy sensor data

$\omega_{ts,noisy\_data} \leftarrow \emptyset$
$ts_{noise} \leftarrow ts_{start}$
while $ts_{noise} < ts_{end}$ do
    $ts_{noise} \leftarrow ts_{noise} + Random(range_{ts})$
    $\omega_{noisy\_data} \leftarrow \{ts_{noise}, Random(range_{noise})\}$
$\omega_{noisy\_sensor} \leftarrow MergeAndSort(\omega_{noisy\_data}, \omega_{sensor})$

return $\omega_{noisy\_sensor}$
```

4 EXPERIMENTAL RESULTS

4.1 Baseline Performance of the Authentication Applications

In this subsection, we present the baseline performance of the authentication applications *before noise injection is performed.* The results presented here will serve as a benchmark for assessment of the extent of performance degradation in the later subsections after noise injection is implemented.

Table 1 shows the baseline performance of user authentication. The performance is expressed in terms of the mean F-score and the standard deviation of F-scores across the user population. For both the SVM and Logistic Regression classifiers, user authentication saw mean F-scores above 80%. These results are comparable with findings from previous research (e.g., see [GSMS17, Pr14]), which implies that our noise injection evaluation should give a realistic reflection of the state-of-the-art.

Authentication	**SVM**		**LR**	
	Mean	**Std**	**Mean**	**Std**
Gait	83.57	2.69	82.27	3.63
Handwriting	87.35	1.14	85.18	1.54

Tab. 1: Average F-Scores and standard deviation of F-scores registered across the user population for both handwriting and gait-based user authentication

4.2 Evaluating the Impacts of Noise Injection

In this subsection we explore the impacts of noise injection on gait and handwriting authentication applications. For each key observation made, we present: (1) the observation, (2) evidence in support of the observation, and, (3) the implications of the observation on the state-of-the-art. We will analyze the global and user-level impacts of noise injection based on the classes stipulated by the biometric menagerie [YD10, MTM14, Wa12]. We briefly introduce the biometric menagerie before presenting the main results.

The biometric menagerie: The biometric menagerie (also known as the ”Doddington zoo” [R.98, YD10]) is a framework through which the users of a biometric system are categorized into groups based on: (1) how they perform when matched against themselves, and, (2) when matched against others. The mechanism categorizes users into different animals based on how they perform. *Lambs* are the users who are vulnerable to impersonation, while *goats* are those users who are unsuccessful at authenticating against their own profiles. *Sheep* on the other hand are the users who exhibit good authentication performance while *wolves* are users who are exceptionally successful at impersonating others.

The lambs and goats are poor users (due to issues cited above with their authentication performance) while sheep are the good users. Because the dynamics of animal behavior in the menagerie provide fine-grained details on, and strongly influence, the performance of a biometric system, the biometric menagerie is at the heart of many performance enhancement and evaluation mechanisms in both benign and adversarial settings (e.g., see [Wa12]). We will use the biometric menagerie for a part of our noise injection analysis so our findings can easily be interpreted in the context of existing frameworks and past work built around the menagerie. The discussion on the impacts of noise injection follows.

Observation #1: At a noise threshold of 1 × the base range, both applications had *a statistically significant dip in F-score* relative to that seen when no noise was injected. On increasing the amount of noise further, gait authentication performed better than the handwriting authentication.

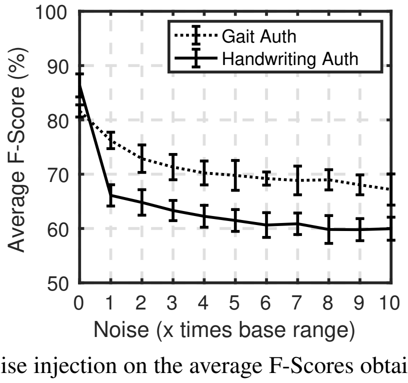


Fig. 2: Global impact of noise injection on the average F-Scores obtained for gait-based authentication and handwriting-based authentication. The error bars indicate one standard deviation, and are plotted at only whole number boundaries, 1, 2, 3, etc., to avoid cluttering the figure.

Authentication Application	**P-value**
Gait	8.9×10^{-5}
Handwriting	9.1×10^{-5}

Tab. 2: p-values obtained from the Wilcoxon Signed rank test under the null hypothesis of noise injection having no effect on application accuracy and the alternative hypothesis of noise injection causing a reduction in accuracy. The p values indicate strong evidence in favor of rejecting H_o (i.e., noise injection significantly reduced application performance for all three applications).

Evidence in support of Observation #1: Figure 2 and Table 2 summarize our evidence in support of Observation #1. The p values in Table 2 were obtained from a series of Wilcoxon Signed rank tests that were run with the following null and alternative hypotheses for each of the two applications. H_o: The F-scores seen before noise injection did not differ from those seen after the first threshold of noise injection (i.e., at Noise = 1× base range on Figure 2). H_a: The F-scores seen before noise injection were greater than those seen after the first threshold of noise injection.

At the 5% significance level, we rejected H_o for both applications, indicating that noise injection significantly reduced the application F-scores/performance. Note that we only ran the statistical tests to compare the performance before noise injection with that seen after the injection of a very low amount of noise since this is sufficient to showcase the minimum impact of noise injection. Also note that at each noise threshold, we compute 20 different F-scores via cross validation; hence the statistical tests run at each threshold are between two 20-dimensional vectors. Figure 2 shows the observed pattern over a wider range of noise thresholds, explaining the rest of Observation #1.

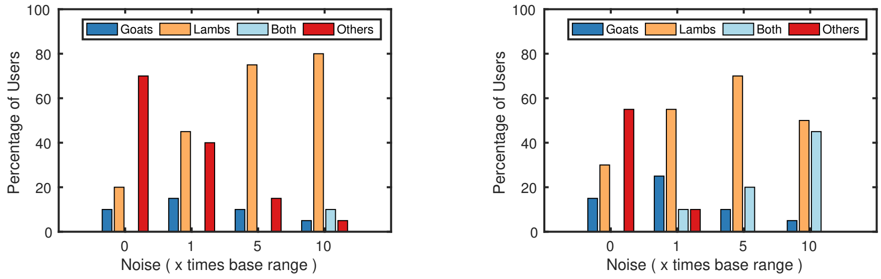
Implications of Observation #1 on research in this area: Observation #1 provides strong evidence against the notion that noise injection can thwart the attacks without significantly impacting the benign applications. The two applications studied exhibit variations in behavior, however, they all see a significant dip in F-score even with low amounts of noise.

In Observation #1, we have taken a coarse-grained view of application performance — i.e., we have studied each application in terms of a global mean F-score. In Observation #2, we take a deeper look at each application, exploring the impact of noise injection on *each user* in our authentication applications.

Observation #2: The decrement in global mean F-score seen due to noise injection mostly manifested as an increment in one of the two classes of poor performing users: the lambs.

Evidence in support of Observation #2: Figure 3 shows the evolution of the biometric menagerie under noise injection. Gait authentication attained a monotonic increment in the number of lambs as the amounts of noise injected increased (see Figure 3a). Figure 3a shows that about 80% of the user population had become lambs and about 10% of the animals had become goats when noise injection reached the highest threshold.

On the other hand, Figure 3b shows that handwriting authentication attained a similar trend (i.e., monotonic increment in the number of lambs as the amounts of noise injected



(a) Gait authentication

(b) Handwriting authentication

Fig. 3: Illustration of how noise injection morphs the biometric menagerie as the amounts of noise injected increase. The figure shows that noise injection increased the proportion of one of the two classes of poor performing users (i.e., the lambs) and not the other (i.e., the goats).

increased) except for the last noise threshold. At that last noise threshold, about 50% of the users became lambs and about 5% had become goats (see Figure 3b).

Overall, Figure 3 reveals that there were increments in the number of lambs at all thresholds of noise injection. The figure also shows that handwriting authentication attained slightly higher numbers of users who were simultaneously lambs and goats at different thresholds of noise injection.

Implications of Observation #2 on research in this area: To get a deeper understanding of the effects of noise and how it might better be tuned to co-exist with the benign applications, its necessary to look beyond the global application behavior and study the ”atomic” dynamics (in this case user-level dynamics) influencing the observed global behavior. For example, for the two biometric authentication applications studied in our investigations, observations on the animal transitions in the menagerie could be fed into well-known theoretical models of animal behavior in the menagerie (e.g., see [MTM14]). This could in turn enable fine-grained sensitivity analysis on the impacts of the noise, or how the system could be tuned to withstand it. A global mean accuracy number would not provide nearly as much information as this.

5 DISCUSSION AND CONCLUSION

We have shown that, contrary to what has been reported in recent works, noise injection is not a viable defense against side-channel attacks that use sensor data.

Application Impact: We found that the average F-score for the authentication applications was degraded by up to 23% for the base range of noise and dropped up to 30% with 10 times the base noise range. Even within the same authentication application, there was significant variation in impact as shown in Figure 3. These impacts could make using some applications difficult or impossible for users.

Alternate Noise Approaches: An alternate solution to globally injecting noise into all sensor streams would be categorizing apps according to noise tolerance. If an app needs high granularity data, it would be provided sensor data with less noise. However, this would require significant infrastructure changes to allow for app specific sensor noise making this solution infeasible.

Other alternative solutions would be (1) requiring mobile apps to request permission to use any sensor, a policy in common with other sensors on Android phones, such as GPS, and (2) implementing new permissions that allow certain apps to restrict sensor data e.g., a banking application could request permission to deactivate all accelerometer and gyroscope sensor usage on the phone while the user is typing in their pin.

Conclusion: In this paper, we have studied the impact of sensor noise injection on two widely studied biometric authentication applications (i.e., gait and handwriting authentication). We have found that both applications see significant degradation in performance after moderate amounts of noise are injected into the data stream. Further, we have found that different categories of users are affected differently by the noise injection. The paper calls for more rigorous research on the impact of noise injection on a wide range of applications before it can be universally deployed a defensive scheme in mobile and wearable devices.

6 ACKNOWLEDGMENT

This research was supported by National Science Foundation Award Number: 1527795.

References

[Da16] Das, Anupam; Borisov, Nikita; Chou, Edward; Mughees, Muhammad Haris: Smartphone Fingerprinting Via Motion Sensors: Analyzing Feasibility at Large-Scale and Studying Real Usage Patterns. arXiv preprint arXiv:1605.08763, 2016.

[DBC16] Das, Anupam; Borisov, Nikita; Caesar, Matthew: Tracking mobile web users through motion sensors: Attacks and defenses. In: Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS). 2016.

[Fr13] Frank, Mario; Biedert, Ralf; Ma, Eugene; Martinovic, Ivan; Song, Dawn: 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, 2013.

[GSMS17] Griswold-Steiner, Isaac; Matovu, Richard; Serwadda, Abdul: Handwriting watcher: A mechanism for smartwatch-driven handwriting authentication. In: Biometrics (IJCB), 2017 IEEE International Joint Conference on. IEEE, pp. 216–224, 2017.

[Mi12] Miluzzo, Emiliano; Varshavsky, Alexander; Balakrishnan, Suhrid; Choudhury, Romit Roy: Tapprints: your finger taps have fingerprints. In: Proceedings of the 10th international conference on Mobile systems, applications, and services. ACM, pp. 323–336, 2012.

[MTM14] Murakami, Takao; Takahashi, Kenta; Matsuura, Kanta: Toward optimal fusion algorithms with security against wolves and lambs in biometrics. IEEE Transactions on Information Forensics and Security, 9(2):259–271, 2014.

[Ow12] Owusu, Emmanuel; Han, Jun; Das, Sauvik; Perrig, Adrian; Zhang, Joy: ACCessory: password inference using accelerometers on smartphones. In: Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications. ACM, p. 9, 2012.

[Pr14] Primo, Abena; Phoha, Vir V; Kumar, Rajesh; Serwadda, Abdul: Context-aware active authentication using smartphone accelerometer measurements. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 98–105, 2014.

[R.98] R. Doddington, George; Liggett, Walter; Martin, A; Przybocki, Mark; Reynolds, Douglas: SHEEP, GOATS, LAMBS and WOLVES: a statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation. In: International Conference on Spoken Language Processing. 01 1998.

[Si16] Sitová, Zdeňka; Šeděnka, Jaroslav; Yang, Qing; Peng, Ge; Zhou, Gang; Gasti, Paolo; Balagani, Kiran S: HMOG: New behavioral biometric features for continuous authentication of smartphone users. IEEE Transactions on Information Forensics and Security, 11(5):877–892, 2016.

[SMS16] Shrestha, Prakash; Mohamed, Manar; Saxena, Nitesh: Slogger: Smashing Motion-based Touchstroke Logging with Transparent System Noise. In: Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks. ACM, pp. 67–77, 2016.

[Wa12] Wang, Z.; Serwadda, A.; Balagani, K. S.; Phoha, V. V.: Transforming animals in a cyber-behavioral biometric menagerie with Frog-Boiling attacks. In: 2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS). pp. 289–296, Sept 2012.

[WLRC15] Wang, He; Lai, Ted Tsung-Te; Roy Choudhury, Romit: Mole: Motion leaks through smartwatch sensors. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking. ACM, pp. 155–166, 2015.

[YD10] Yager, Neil; Dunstone, Ted: The Biometric Menagerie. IEEE Trans. Pattern Anal. Mach. Intell., 32(2):220–230, February 2010.