Simple defences against vibration-based keystroke fingerprinting attacks

Rushil Khurana[1] and and Shishir Nagaraja[2]

[1] IIIT Delhi, India
rushil09040@iiitd.ac.in
[2] University of Birmingham, UK
s.nagaraja@cs.bham.ac.uk

Abstract. Smartphones are increasingly equipped with sensitive accelerometers that can analyse acoustic vibrations on a physical surface. This allows them to gain a covert understanding of the surrounding environment by combining accelerometer sampling with sophisticated signal processing techniques. In this work, we analyse keyboard-sniffing attacks based on acoustic (vibration) covert channels, launched from a malicious application installed on a smartphone. An important requirement of such attacks is access to reliable acoustic signals that can be distinguished from the noise floor by applying appropriate signal processing techniques. Our analysis indicates that state-of-the-art attack techniques are fragile; injecting randomised noise (jamming) via the vibration medium into the accelerometer, reduces the efficiency of the attack from 80% to random guessing. We conclude that our work presents an important step towards disabling the covert channel and ensuring full security.

1 Introduction

The accelerometer sensors of modern mobile devices are getting increasingly powerful (around 100Mhz). After circumventing weak access control systems, if any, a malicious application on a mobile device such as a smartphone, can analyse incoming accelerometer signals to covertly gain an understanding of activities in its physical surroundings in an unauthorised manner.

Past work [2, 3, 5, 7, 6], on exploiting accelerometers to sniff keystrokes required direct physical contact with the keyboard. However a smartphone based accelerometer can be used to recover keystrokes to some extent **without** direct contact with the keyboard: Marquardt et al. [1] devised an attack that uses accelerometer readings from a smartphone to recover text being typed on a victim's keyboard. The keyboard and the smartphone are separated by a few inches on the same table. Accelerometer readings are collected by a mobile application running on the smartphone. Machine learning techniques are then applied to decipher English dictionary words. Marquardt et al. report an accuracy of recovering around 80% of the typed words.

Since the Marquardt attack is based on analysing acoustic vibrations, it is vital to examine their attack model in the context of **all** acoustic vibrations that

are present in a reasonable setting including the keystrokes' acoustics and then test the effectiveness of the attack. Our contributions are as follows:

- We propose simple and usable defences based on injecting acoustic (vibration) noise into the covert channel to reduce the effectiveness of sophisticated signal processing techniques.
- We analyse and report on the practicality of the Marquardt attack within the original threat model [1]. We found that it attack effectiveness is much lower than a laboratory setting.

The Marquardt attack exploits weak OS access control mechanisms which allow smartphone applications to access the accelerometer sensor without explicit user permission. However, a major challenge they needed to overcome in comparison to past work is that the sampling rate of the accelerometer sensor present in a mobile device is a full two orders of magnitude less than that of the devices used in the previous works [2]. As a result, naive approaches involving direct mapping between signal features to keyboard input do not work. To overcome this challenge, they chose to apply the well known technique of bigram analysis where the statistical characteristics of the vibration signal can be uniquely mapped to two consecutive keypresses instead of a single keypress.

Vibration based side-channels pose an important threat to user security as they can leak confidential information. Such attacks leverage the fact that most users tend to place their mobile devices next to the keyboard. The attacker installs a keylogger via a social malware attack [8] on to the victim's mobile phone, which can record and relay stolen information (keystrokes) to the attacker.

The use of machine learning techniques to analyse vibrations caused by keystrokes or other sensitive information has given rise to fresh concerns about user security. However, we find that it is fairly straightforward to induce error into such attack techniques. In this paper, we analyse the robustness of the Neural Network technique, a supervised machine learning technique. We find that the attack is surprisingly ineffective when dealing with low levels of noise. Careful application of periodic acoustic noise alone can bring the classifier accuracy close to that of random guessing. It is clear that the attack is rendered completely ineffective by the application of pseudorandom noise.

We start by describing the Marquardt attack and explore the effectiveness of the attack under various common-day scenarios where random noise accompanies the acoustic vibrations produced by full-size desktop keyboards.

2 Keystroke fingerprinting attack using neural networks

In the following section, we describe the Marquardt attack for fingerprinting vibrations caused by keyboard usage. Consider a desktop computer user operating the computer through a keyboard placed on a desk. Now consider a smartphone placed on the same desk a few inches away from the keyboard. The Marquardt keysniffing attack leverages a acoustic covert channel between the keyboard and

a malicious application running on the smartphone. Vibrations induced by keypresses on to the desk shared by the smartphone and the computer keyboard are captured by the smartphone accelerometer. The collected information is then analysed by the malicious smartphone app (attacker) which applies a machine learning technique on the sampled vibration signals to infer the words typed in by the computer user (defender).

2.1 Marquardt's Keypress Fingerprinting Model

The sampling rate of accelerometers in mobile devices is too low to map the maximum amplitude of a vibration signal to a unique keypress (on a nearby keyboard). Therefore, the Marquardt adopts a bigram approach towards modeling keypresses; instead of a single keypress, a pair of keypress events is modelled together. Let E_i and E_j be sequential keypress events. The following features have been used to characterise the event (E_i, E_j):

1. **Keyboard Position:** For each event E_i, $pos(E_i)$ is a feature that describes the relative position of E_i to a central line dividing the keyboard into two parts- *left(L)* and *right(R).*
2. **Distance Between Successive Keypress Events:** For each pair of successive keypress events, $dist(E_i, E_j)$ is a feature that describes the distance between the two keypress events for a given pair. For a pre-determined value α, $dist(E_i, E_j)$ is either *near(N)* or *far(R)* where $N < \alpha$ and $R \geq \alpha$.

Each pair of successive keypress event (E_i, E_j) is represented by $pos(E_i)||pos(E_j)||dist(E_i, E_j)$ where $||$ represents feature concatenation. Any word can thus be represented by a sequence of concatenated features for each event-pair appearing in the word. For example, let $\alpha = 3$ and consider a partition of a QWERTY keyboard. All keys on the left of 't', 'f' and 'v' (inclusive) are assigned to the first partition named **L**eft. The remaining keys are assigned to partition named **R**ight. The word **rope** can then be represented as:

RO . OP . PE
LRF.RRN.RLF

It is clear from the above example that a word of n letters can be broken down into $n-1$ constituent character representation in the attack model. This is a compact representation of words. The corresponding text can be extracted by processing it as shown in the following section.

2.2 Attack

The attack consists of two phases, supervised learning and analysis which we outline in multiple steps as follows.

Data Collection: When a key is pressed, the mobile application records the surface vibrations produced in the process and stores a three dimensional vector

(x, y and z axes) per sample in a log. The log thus generated, is a dump of raw accelerometer readings obtained using the victims phone.

Feature Extraction: Next, simple statistical metrics are computed over the collected data to yield a compact representation consisting of the following feature vector corresponding to each keypress.

$$Keypress(E_i) = \\ (mean, kurtosis, variance, min, max, energy, rms, mfccs, ffts)$$

Feature Labelling: To train the system, the above two steps are performed using a chosen dictionary of words. Next, each word in the training dictionary is broken down into a LR/NR (Left-Right Near-Far) representation. The model prepares a training data set by labelling feature vectors extracted in the previous step as either left(L) or right(R) for individual keys. For key-pair samples the feature vectors of each of the constituent letters is concatenated together and then this composite vector is labelled as either near(N) or far(F). After this step, we will have a training set containing labelled feature vectors.

Neural Network Setup: Two neural networks are created from the training set obtained — one for classifying left-right feature vectors (hereon, referred to as L/R classifier) and the near-far feature vectors (hereon, referred to as N/F classifier). After training using the labelled data from the dictionary; these two neural networks can be used to classify and label accelerometer readings from the log obtained from the victim as L/R or N/F.

Word Matcher: The word matcher assigns a score against each word in the dictionary to each of the word representation it obtains from the previous steps. The scored dictionary words are then sorted on the basis of their scores and the top k results are presented as predictions for the given representation.

2.3 Attack efficiency before application of defences

As our first experimental step, we reproduced the Marquardt keyboard sniffing attack [1]. It involved two experiments to measure the accuracy of the two classifiers involved, and two experiments to measure the recovery of text. We obtained comparable accuracy results as the original authors. Marquardt et al. recovered 80% of the text from a self-built context-aware dictionary in the top 5 guesses. In comparison, we were able to recover 76% of the text.

3 Effectiveness of Marquardt's attack under random noise

Given the relatively high accuracy of the attack, we devised a series of further experiments to analyse the Marquardt attack in a practical setting of a motivated defender. Specifically, we measured the susceptibility of the attack torandomised noise. For each of the following experiments, we measured the accuracies of the two classifiers involved in the attack. Their accuracy is a direct indicator of the rate of text recovery.

Sampling Rate	100	80	64	50
Trained at 100	88% (L/R), 76% (N/F)	85%, 70%	76% , 67%	76% , 64%
Trained at 80	85% (L/R), 73% (N/F)	85%, 76%	79% , 67%	73% , 64%
Trained at 64	79% (L/R), 70% (N/F)	79% , 70%	79% , 73%	67% , 64%
Trained at 50	70% (L/R), 64% (N/F)	70% , 67%	67% , 64%	61% , 58%

Table 1. Table showing results of accuracy of L/R and N/F classifier measured with a single Harvard Sentence. Each column entry is the accuracy of each of the classifiers trained at a sampling rate of the corresponding row header and tested at the corresponding column header.

3.1 Variation across keyboards

Marquardt et al claim that their attack does not require violating the physical security of the victim (unlike previous attacks). This implies that the attacker is unaware of the target's keyboard make and model. Therefore, we examine change in attack efficiency when the training keyboard is different from the keyboard on which the attack is applied.

Keyboards	K1	K2	K3	K4	K5
Trained K1	88%(L/R),76%(N/F)	76%,64%	70%,58%	52%,47%	52%,50%
Trained K2	76%(L/R),67%(N/F)	91%,70%	73%,61%	58%,50%	52%,52%
Trained K3	70%(L/R),61%(N/F)	73%,61%	85%,73%	64%,52%	58%,52%
Trained K4	58%(L/R),50%(N/F)	61%,55%	61%,58%	85%,70%	73%,73%
Trained K5	58%(L/R),52%(N/F)	55%,52%	64%,55%	76%,70%	79%,70%

Table 2. Table showing results of accuracy of L/R and N/F classifiers measured with a single Harvard Sentence. Each column entry is the accuracy of each of the classifiers trained with the data collected from the keyboard of the corresponding row header and tested with data collected from the keyboard of the corresponding column header.

We used the following keyboards in our set:

- K1: a HP KB-0316 keyboard.
- K2: a SK-1688 keyboard.
- K3: an Intex keyboard-M/M Rolex.
- K4: an iBall KB279 keyboard.
- K5: a Wipro SK-2030 keyboard.

We trained the attack classifiers on acoustic vibrations from one of the keyboards from the set followed by an attack targeting every other keyboard in the set. We evaluate attack efficiency by measuring classifier accuracy. The results are shown in Table 2.

Keyboards K1 and K5 have identical layouts but differ in terms of spacing between adjacent keys. Thus training on K1 and attacking K5, results in decreased attack efficiency. Similarly, keyboards K1 and K2 nearly identical spacing between keys resulting in similar attack efficiency.

3.2 Impact of sampling rate on attack efficiency

Apart from the accelerometer sensor itself, the sensitivity and resolution of the accelerometer is also dependent on the firmware settings and the mobile operating system's data handling and delivery mechanisms. We assume a broad spectrum of available accelerometer sampling rates and measure the accuracy of the attack. We do this by training the attack framework with a particular sampling rate and test it against data collected at a higher or lower sampling rate. We noted the accuracy of the L/R and N/F classifiers at various sampling rates, this is presented in Table 1. Each column in the given table indicates the accuracy achieved by the L/R and N/F classifier when Experiment 1 — Using a single Harvard sentence was conducted. The phone was trained with the sampling rate in the corresponding row header and tested with a sampling rate of the corresponding column header.

Table 1 shows that the sampling rate impacts accuracy of the attack. A attack classifier trained at a certain sampling rate has somewhat reduced efficiency at lower sampling rates. Roughly, a 20% reduction in sampling rate reduces attack accuracy by 10% to 15%.

3.3 Defensive vibrations

So far we have considered the challenges faced during reproduction of the attack in a practical setting. We now consider automated defences against the entire class of covert-channel attacks that leverage vibrations induced on shared physical surfaces. Our approach to defence is to "sanitise" the physical surface supporting a user device using additional user-controlled devices that transmit **defensive vibrations** into the shared physical medium. Defensive vibrations can be introduced according to a variety of strategies. An optimal strategy would be to transmit well designed signals waveforms into the shared medium such that the user-dependent vibration signals are exactly cancelled out. A simpler and more obvious strategy is to introduce random vibrations which we explore in this section.

Therefore, in our next experiment we used a buzzer within a *defender* phone (of the type found on conventional mobilephones/pagers) to produce acoustic vibrations in various configurations (distances and angle) with respect to the locations of the attack smartphone. A defensive buzz (induced vibration from the buzzer) consists of switching the vibrator (defender's phone) on and off periodically for the duration of the attack. It depends on three parameters: relative location from attacker phone and keyboard, the signal amplitude (buzz volume), signal-on duration, and signal-off duration.

In our experiments, the attack phone and the keyboard were placed four inches apart while the defender phone was placed at various positions relative to them, as follows.

- Keyboard, attacker phone, and defender phone are placed in a straight line. Defender phone is placed between the keyboard and the attacker phone at a distance of one inch from the keyboard.

– Keyboard, attacker phone, and defender phone are placed in a straight line. Defender phone is placed one inch from the attacker phone.
– Keyboard, attacker phone, and defender phone are not placed in a straight line. Defender phone is placed one inch directly below the attacker phone.

At each of these positions, defensive buzzing was continuously applied for the duration of the attack at three different intensities from a Nokia-2100 phone — gentle (signal-on for 0.5 seconds signal-off for 2 seconds), medium (signal-on 0.5, signal-off 1 second), and aggressive (signal-on 0.5 seconds, signal-off 0.5 seconds). We trained the attack classifiers using a single Harvard sentence and calculated the accuracy of the L/R and N/F classifier at each of those points at each of the specified pace. We averaged out the results at each point for each of the specified intensity of defensive buzzing. The results are as shown in Table 3.

Buzzing intensity	L/R	N/F
Gentle	85%	76%
Medium	79%	73%
Aggressive	70%	61%

Table 3. Table showing results of accuracy of L/R and N/F classifier measured with a single Harvard Sentence while injecting noise with defensive buzzing at various paces. Each result in the column entry is averaged out from three pre-decided points.

We observe that aggressive buzzing is able to lower the classification rate to just 61%, which is ten percent better than a random guess. We believe that by better calibration, defensive buzzing can achieve much improved results.

3.4 Data Collector as a source of acoustic noise

We now consider active defences where the victim's phone (so far referred to as the attacker phone) also participates in the defence mechanism. A possible defence allowed by the Marquardt threat model is the direct injection of noise into the attacker's data on the smartphone itself. Therefore, we analysed the effects of playing music on attacker/victim smartphone on attack efficiency. Again, a completely plausible scenario.

In this experiment, we played ten songs each from the following categories: Rock, Pop, Jazz, Classical, Blues, Hip Hop, R&B and Bollywood. We tried to pick as diversified songs as possible from the sub genres of each category. We measured the efficiency of the classifier in each case. The results are as shown in Table 4.

For all categories, the accuracy was less than 50% which means that the classifier's detection rate at deciding the relative position of the keypress was worse than a coin toss.

Music Genre	L/R	N/F
Rock	44%	38%
Pop	47%	47%
Jazz	41%	38%
Classical	29%	41%
Blue	38%	44%
Hip Hop	29%	44%
R&B	38%	47%
Bollywood	29%	26%

Table 4. Table showing results of accuracy of L/R and N/F classifier measured with a single Harvard Sentence while playing a song. Each result in the column entry is averaged out for 10 songs from each category.

Additional sources of noise: Calling/Receiving a phone and connecting via text messaging are among the most basic functionality of a mobile phone. In a real-world setting, the victim's phone could receive a text message, or receive a call, or buzz in response to event notifications. In workplaces, most of the phones are usually set to vibrate mode and even if not, phones often ring as well as vibrate. To understand the change in attack efficiency as a result of noise resulting from day-to-day operations, we ran the following experiments.

1. We generated a notification by text messaging the phone while the malicious application was running. This could be seen as any notification as usually the notification alert is similar (if not same) in android.
2. We called the victim phone while the attack was in progress. We however, did not receive the call as it would disturb the orientation of phone and the experiment is merely to understand the affect of vibrations produced by the phone. The attack obviously would fail if the victim picks up her phone from the table itself.

For each of the two scenarios, we conducted the experiment ten times and averaged out the results. In scenario one, the accuracy of the L/R classifier was noted to be 67% and the N/F classifier was only accurate 55% of the time. In scenario two, the accuracy of both the L/R classifier and the N/F classifier was noted to be 41% and 35% respectively. Therefore, suggesting that attack efficiency is very low if the victim receives a call during the attack.

4 Conclusions and future work

We have demonstrated that several simple defence options against the Marquardt attack are available to the motivated defender. Something as trivial as a periodic buzz can reduce the attack efficiency to 61%. Better defences are obtained by inducing low-frequency acoustic noise resulting in lowering attack efficiency to less than 50% (in a binary choice problem).

We have demonstrated that the attack can fail under various day-to-day scenarios. However, this is due to the limitations of the signal processing techniques which are not noise-tolerant. Overall, the application of machine learning techniques for statistical analysis of sensitive user data is fraught with difficulties. In particular, the ML technique used by Marquardt et al. is easily thwarted by random perturbations.

In future work, we will establish a comprehensive defence mechanism which can "cancel" the acoustic vibrations induced by a keyboard, hence moving towards a provably secure mechanism which can prevent the flow of data across the acoustic channels involved in the attack. The idea is to sanitise the nearby environment in a manner such that the sensor picking up data cannot differentiate between the instance of keyboard being typed on and the instance when it is not. That would ensure that irrespective of the attack mechanism or the algorithm used behind such an attack, it would be defended against.

References

1. Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. (sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers. In Proceedings of ACM Conference on Computer and Communications Security, CCS, 2011
2. Y. Berger, A. Wool, and A. Yeredor. Dictionary Attacks Using Keyboard Acoustic Emanations. In Proceedings of the ACM Conference on Computer and Communications Security (CCS), 2006.
3. L. Cai, S. Machiraju, and H. Chen. Defending Against Sensor-Sniffing Attacks on Mobile Phones. In Proceedings of ACM SIGCOMM Workshop on Networking, Systems, Applications on Mobile Handhelds (MobiHeld), 2009.
4. Mobile that allows bosses to snoop on staff developed BBC NEWS- Technology. Web. 27 July 2010. ⟨`http://news.bbc.co.uk/2/hi/technology/8559683.stm/`⟩.
5. E. Owusu, J. Han, S. Das, A. Perrig, J. Zhang. ACCessory: Password Inference usingAccelerometers on Smartphones. In HotMobile 2012 - The 13th International Workshop on Mobile Computing Systems and Applications, 2012.
6. D. Asonov and R. Agrawal. Keyboard Acoustic Emanations. In Proceedings of the IEEE Symposium on Security and Privacy, 2004.
7. I. S. on Subjective Measurements. Ieee recommended practices for speech quality measurements. IEEE Transactions on Audio and Electroacoustics, 17:22746, 1969.
8. S. Nagaraja and R. Anderson, The snooping dragon: social-malware surveillance of the Tibetan movement, In technical report UCAM-CL-TR-746, University of Cambridge, 2009.
9. R. Schlegel, K. Zhang, X. Zhou, I. Mehool, A. Kapadia, and X. Wang. Soundcomber: A stealthy and context-aware sound trojan for smartphones. In Proceedings of the 18th Annual Network and Distributed System Security Symposium, NDSS'11, 2011.