Keystrokes Recovery from Smartphone's Accelerometer

Jennifer Guo	Yi-Hsien (Stephen) Lin	Akshay Mittal	Wathsala W. Vithanage
jjguo@	yihsien@	akshay@	wathsala@

1. What is the problem we are working on? Why work on it?

As smartphones become prevalent, and is equipped with continuously increasing computational power, memory capacity and high quality sensors, smartphones with malicious softwares can impose a great security threat and privacy violations to us. Our team have read an interesting paper published in 2011 about keystroke learning through decoding the unique vibrations of each keys using smartphones accelerometers. This threat is especially interesting because different from other sensors such as cameras and microphones, softwares can access accelerometers without users permission. The paper used a neural network for the learning model and conducted the experiment on an iphone 4, which gave them accuracy rate around 80%. Although 80% accuracy is quite impressive already, in order to correctly reconstruct the original keystrokes, a decent amount of human labor is still required to analyze the output of the keystroke learner, which reduces the threat this attack might bring. However, the paper didnt quite justify their choice of the learning model, nor did they compare their learning model with other possible candidate models such as HMM. We are curious that if applied with different learning models, combined with the state of the art accelerometers that are more sensitive and noise resistant, will it be possible for us to boost the accuracy significantly to nearly 100%. Should we succeed, this threat is definitely severe and must be addressed seriously.

2. What data are out there that can help us solve it?

This is a relatively new domain of research and has not been explored in detail. Marquardt et. al [1] have developed the prediction strategies using their own dataset built using iPhone 4 and unfortunately this dataset is not public. We plan to build our training using an accelerometer on an Android device, and will construct the dataset ourselves. This would require building an simple Android application which starts recording the vibrations that it receives and transfers them to the computational device (laptop) of our choice. In the learning phase, we will type all the letters of the English alphabet, a through z, 200 times each (in no particular order) and the accelerometers reading will be collected to form a 4600 key-pressing events dataset. This will be followed by feature extraction from the signals - we plan to leverage the features which are used by Marquardt et. al i.e. <mean, kurtosis, variance, min, max, energy, rms, mfccs, ffts>, and extend it further. We next plan to construct the training dataset of signals of keystrokes for a dictionary of words. This will be done by combining the feature vectors for pairs of characters occurring in the words. This task of constructing the dataset should not take much time as compared to typing the different words of a dictionary and labelling the corresponding vibration signal that is generated for each letter. This is because typing is error-prone and it is non-trivial to incorporate the effect of using Backspace with the wrongly typed-letters in order to construct the accurate vibrations imprint for a particular word.

3. What methods have others tried to solve this problem?

Marquardt et. al [1] have tried a dual model of neural networks to predict the words typed on the keyboard placed near a iPhone 4. They construct a neural network for each individual characters vibration signal and a second neural network for the signal obtained for pairs of characters which constitute a word. They achieve 80% accuracy in recovery of the typed content. Their recovery model will be our main focus of comparison. Many researchers have focussed on reconstruction of the typed words based on the acoustic signals emanated from the keyboard and collected using the conveniently available microphones in the mobile devices. These works have shown the ability to

recover greater than 80% of keyboard presses given substantial training Asonov et. al [2], without training Zhuang et. al [3] and based on acoustic dictionaries Berger et. al [4]. While such approaches are certainly more within the reach of the adversary, they are extremely difficult to scale to large deployments as they require that a microphone is physically located near all potential targets at all times plus if they are used using mobile devices then permissions must be granted to the corresponding application which desires to use the functionality of the microphone.

4. What methods do we plan to explore and/or develop?

Since we try to map accelerometer readings into keystrokes we will have to label our training inputs. Therefore we will be using supervised learning algorithms and our choices are HMMs and neural networks. The goal of having two different models is that, we will be able to compare their performance in the evaluation step. In the training phase we will capture vibrations over the surface using the accelerometer in the mobile phone and extract features from the signal received by the mobile phone accelerometer. When generating the training data we will install a keystroke logger in a computer that the user uses to generate keystroke sequences based on a dictionary. At the same time on the same surface an Android based mobile phone will be kept so that it will log accelerometer readings into a text file in parallel. Once all the keystrokes are typed by the user, we will download the accelerometer log to a PC where we perform all the training, testing and evaluation. Before extracting data we may have to use a filter to eliminate any noise thats being generated by the sampling process. Cleaned data will then be splitted into two groups, the training set and the test set. From both training and the test sets we will extract features mentioned previously and use them to train and evaluate HMM and the neural network separately. In the evaluation stage we will compare the two models for accuracy. Please note that our goal here is only to demonstrate the plausibility of this attack which is based on machine learning, therefore we may not develop an Android app that streams accelerometer readings to a remote site to construct keystrokes in real-time. We avoid this step simply due to the limited time we have and strictly not due to any technical difficulty.

5. What does success mean? How will we evaluate success?

Character based accuracy - (1) boolean metric (2) distance metric. Word based accuracy - (1) boolean metric (2) distance metric. We will first evaluate the accuracy based on individual letters where we differentiate only between correctly recognized and incorrectly recognized letters. From that we can calculate an overall recognition accuracy. For a more fine grained metric we will also consider the distance the incorrectly recognized letters are apart from the actual letter. Thus letters that are spatially further away will get a bigger error penalty than letters closer to the actual letter. As an example for the letter D, the letters E,R,F,C,X,S would be only 1 distance away, while the distance to the key K would be 5. After the letter based accuracy, we will also measure based on word accuracy. For that again first we just use a boolean metric where its either correctly or incorrectly recognized. After that we will also include a distance metric to calculate the distance between the predicted and actual word. We will also consider the effect of increasing the training data and how it will affect accuracy.

6. What challenges do you expect to face?

The first challenge we expect to face is with the construction of the dataset. We specifically pick Android platform for the ease of application development to utilize the accelerometer sensor as compared to that of iOS platform. The second challenge would be collection of the feature vectors from the vibration signals. This would involve cleaning the noise which might be present - the noise can emanate from the vibrations obtained by the hand motion/sliding on the keyboard, mouse usage intermixed with typing, to name a few. The third challenge would be to fit the training dataset with appropriate model. Since this is a relatively new field, we will try techniques like neural networks, hidden markov models, and will attempt to compare the accuracies obtained. If the fit is not good, then it would be a non-trivial challenge to extract features in a different manner and fit with either the same models or explore new models.

References

[1] Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. (sp)iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers. CCS '11. ACM.

[2] Dmitri Asonov and Rakesh Agrawal. Keyboard acoustic emanations. In *IEEE Symposium on Security and Privacy*, 2004.

[3] Li Zhuang, Feng Zhou, and J. D. Tygar. Keyboard acoustic emanations revisited. *ACM Trans. Inf. Syst. Secur.*, 2009.

[4] Yigael Berger, Avishai Wool, and Arie Yeredor. Dictionary attacks using keyboard acoustic emanations. In *In Proceedings of Computer and Communications Security (CCS*, 2006.