User-Device Physical Unclonable Functions (UD-PUFs) based on Mobile Device Touchscreen Pressure

Timothy M. Dee
Electrical & Computer Engineering
Iowa State University
Ames, IA, USA
deetimothy33@gmail.com

Akhilesh Tyagi
Electrical & Computer Engineering
Iowa State University
Ames, IA, USA
tyagi@iastate.edu

ABSTRACT

Described in this document is a physical unclonable function (PUF) utilizing the variability derived from the pressure with which users interact with their mobile device touchscreens. We illustrate how a sequence of these pressure values from discrete touchscreen interactions may be used to uniquely characterize a user-device pair. This characterization method may find many applications in protecting access to a mobile device from a malicious party. As a result, the effectiveness of this scheme is described in terms of how one user may be differentiated from another.

1. INTRODUCTION

Mobile devices are ubiquitous in the modern world. These devices are becoming progressively more important for many applications with security sensitive data. Securing mobile devices poses unique challenges and opportunities compared to traditional data security where it is difficult for an attacker to access the physical device on which the data is stored or from which the sensitive data may be accessed. Although there is increased probability that a device may be compromised, there is also greater number of available sensors to measure the variability in the way users interact with mobile devices compared compared to conventional computing technology.

The reality that an attacker may be able to gain access to a physical device makes securing any data stored on or accessed by a mobile device significantly more challenging. Traditional physical unclonable functions (PUFs) which can only be used to uniquely to a given hardware device are no longer sufficient to guarantee the authenticity of a user. This motivates an extension of the traditional PUF known as a user-device physical unclonable function(UD-PUF). This UD-PUF entangles the physical characteristics of the user in combination with the device to enable a more secure authentication scheme.

A UD-PUF is a function of both the hardware of a given device and the user of that device. Such a function must change significantly given unique user-device pairs, thus we must identify a property or properties which vary among mobile device hardware and a property or properties which vary among users of a given device. The best candidates to use will be properties which present with the most variability, and properties which are most easily exposed to the android operating system.

Identifying properties of device and user which can be used to construct our UD-PUF is insufficient; the system must also be practical under normal use conditions for mobile devices. The system must be non-intrusive to the user, fast enough to run on mobile devices, and accurate when authenticating users. If a user must spend a long time authenticating they are not able to use this time to accomplish the task for which they are being authenticated, this detracts from the efficiency of using a mobile device to complete the task. If the proposed system is too computationally intense to be run on a mobile device, or the system is only accurate some low percentage of the time then it is useless for any practical application. For these reasons we propose a system which operates under a continuous authentication scheme.

While other solutions have shown that a user may be distinguished by using data from several different sensors, this paper establishes that variability in the way in which users interact with the touch screen may be enough to differentiate one user from another. In other words there is a single source of information which is also tied to the usability of the device. A failue of the touchscreen to function also constitues a failure in the device as a whole. Thus the system is robust because it will only fail upon failure of the device. Compare this to other systems whose functionality depends on many components. A failure in one of these components disrupts the authentication scheme making these systems more prone to failure.

The structure of the paper is as follows. Section 2 discusses some related work. Touchscreen pressure and what it measures are described in section 3. Reasons for choosing a given modeling scheme for users and devices is discussed in section 4. Section 5 provides some implementation details with respect to how touchscreen pressure is used in the modeling scheme. Data collection methods are discussed in section 6. The authentication scheme is discussed in section 7. The

results including final numbers and suggestions in practical applications are articulated in section 8. Conclusions are presented in section 9. Finally section 10 discusses future work in this area.

2. RELATED WORK

SenSec, a similar authentication scheme to the one proposed in this paper, completed at Carnegie Mellon used information from from accelerometers, gyroscopes and magnetometers to construct a model of a user. [3] This method can be classified as a UD-PUF as each accelerometer, gyroscope and magnetometer will have some variance inherent in the manufacturing process for these devices [**?**]; the input to these devices will also vary significantly by user [**?**]. As a result the output is a function which varies with changes in user or device, a UD-PUF.

Another noteable aspect of the SenSec system is their method of modeling users with an n-order Markov Chain. This system presents several benefits which, according to sensec include, relative simplicity and scaleability. [3]

[2] [1]

3. TOUCHSCREEN PRESSURE

If our goal is be be able to distinguish a user's interaction with a given device from this same user on a different device and from different users on any device than our description of the user will need to be a product of both the user and the device. In the android operating system there exists a pressure function which returns a value proportional to current at sides of phone for a given touch screen interaction. [3] This is the value which will serve as the basis for our scheme and will henseforth be referred to as touch pressure.

The pressure function is significant because it's value not only depends on the characteristics of the device but also on the way in which a user interacts with a given device. The effect of a given device on the touch pressure value will differ significantly due to variations implicit in the manufacturing process for the touchscreens of these devices. [**?**] Our supposition is a given user will interact with a touchscreen in such a way as to cause significant variations in the touch pressure values when compared to other users on the same touchscreen [**?**]. Given this, we have chosen touch pressure as the basis for our UD-PUF.

4. MODELING A USER-DEVICE PAIR

Interactions between users and devices are complex. To interpret these actions in a meaningful way, in order to perform an authentication for example, it is necessary to simplify these interactions. The chosen model must provide sufficient entropy such that a model generated with a given user-device pair is not consistently reproducible by another user or on a different device. The modeling method must also be easily reproducible by the original user on the original device. A model having the necessary characteristics required for this application is a Markov Chain.

Markov Chains are useful in predicting systems whose behavior can be modeled in discrete states. The transitions between states can be identified to happen with some probability.

Historically the Marcov Chain has found applications in (statistics?)

Upon identification of an appropriate model the next step is to discover an optimal way in which it may be applied to the current problem. An interaction between a user and device can be described as a sequence of touch pressure values. Using a Markov Chain to describe this sequence is only reasonable if we suppose that a given touch pressure value depends on some number of preceding touch pressure values. [**?**]

5. TOUCH PRESSURE MODELING

The goal in modeling a system with a Markov model is to classify the system in terms of its transitions between states. If such a model is to be used to purposes of uniquely identifying a given system, than the model must be chosen in a way which exposes the uniqueness of the system. Our scheme uses a Markov model constructed from the sequence of touches entered by a user. Each touch contains information about the pressure and location of the interaction with the touchscreen.

Our Markov model calculates the probability of a given touch coming after a sequence of n touchs. n is a parameter to the model which should allow for increased accuracy in determining the true probability of a given touch to come after a preceeding sequence of touches when given more data to construct the model. The increased accuracy is due to a greater number of possible model states which decreases the liklihood that a user will have states which overlap a different user.

The probability of a touch coming after a sequence of touches can be expressed as the number of occurrences of touch after a given sequence by the total number of occurrences of that sequence.

6. DATA COLLECTION

Data for creating touch pressure models in this experiment was generated using a special keyboard application for the android operating system. Users would load the keyboard onto their device and continue using the device in the way they would normally. Some users were asked to play a typing game in order to help expedite the data collection process. After the users had generated at least ten thousand touches the data was collected from the user's device.

7. DIFFERENTIATING USER-DEVICE PAIRS

In distinguishing a particular user from another different user, it is necessary to develop a method of comparison between users. In our method of comparison we take the probability associated with a touch pressure coming after a sequence of preceding touch pressures for a particular user and compute the difference between this probability and the probability of the same touch pressure coming after the same sequence of touches for a different user. The average of these probability differences is taken to be the difference between two users. Once a comparison is established a natural extension is a system of authentication. This system needs to determine when two sets of touch pressure values came from the same user-device pair. When authenticating a user,

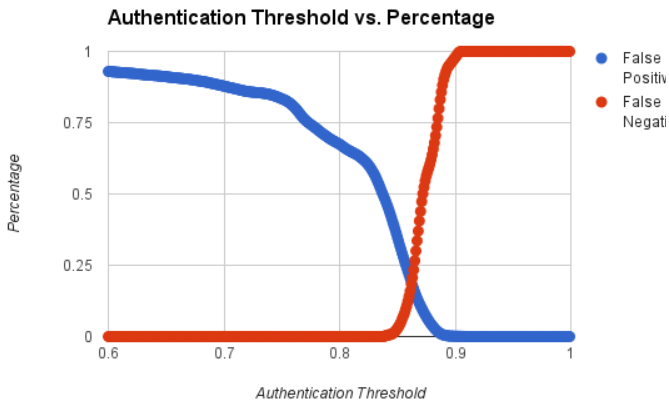

Figure 1: Describes how false positive and false negative percentages vary as the authentication threshold is adjusted.

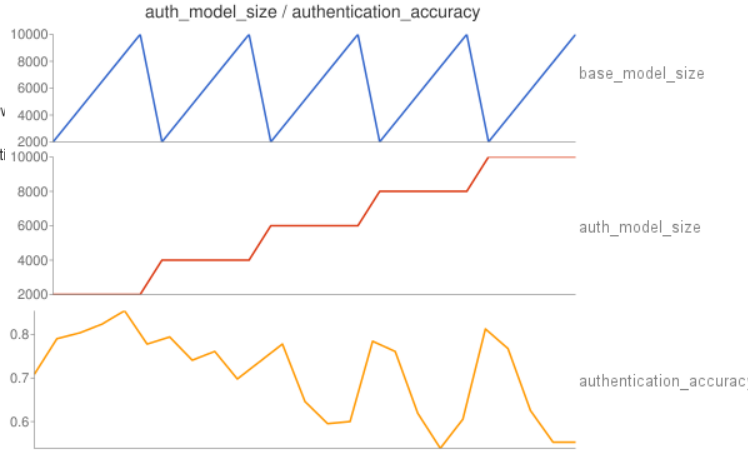

Figure 2: Authentication accuracy is a function of both the base model size and authentication model size.

we take one minus the average difference between the model constructed from the two sets of touch pressure values. Take this value to be the authentication percentage for a given set of touch pressure values against another. To determine how well this system does at differentiating users it is useful to develop metrics which describe the system's performance under conditions which are similar to it's potential real-world applications. Figure 1 illustrates how false positive percentage and false negative percentage vary based on where the threshold for authentication is set.

Here, authentication threshold refers to the value of authentication percentage one model must have against another for the models to be considered the same; two models which are the same are supposed to have been created from touches generated by the same user-device pair. False positive percentage measures what fraction of authentications between two sets of touch pressures which did not come from the same user-device pair, therefore these sets should not be considered the same, but did authenticate as being the same in our authentication system. False negative percentage is exactly the inverse of false positive percentage in that it describes what fraction of authentications between two sets of touch pressures which did come from the same user-device pair, but did not authenticate as being the same in our authentication system.

In Figure 1 there exists a clear intersection between false negative and false positive percentages. This intersection is significant; at this point the system neither biased toward allowing user-device pairs which should not authenticate to pass authentication nor toward disallowing user-device pairs, which should authenticate, from passing authentication. This point represents a balance in design, but the best authentication threshold will depend on the application of this system.

The implementation of the authentication system is as follows. A Markov model is constructed from a sequence of touches known to have come from the user. A seperate Markov model is then constructed from a sequence of touches which need to be compaired to the model. The probabilities computed for each of these models are then compaired and a percent difference is derived from these compairasons. We then choose to authentication only those models which have achieved a low enough percent difference.

8. RESULTS

In exploring the design space of this system we are trying to determine how well the system will perform in the end use case. Perhaps the best measure of the system's performance is how accurate the system is when authenticating users. Figure 2 depicts authentication_accuracy as a function of base_model_size, number of touches known to have originated from an authentic user, and auth_model_size, number of touches which are to be checked against the model generated from the base touches. We define authentication_accuracy to be the percentage of authentications for which our system makes the correct decision. In other words, an authentic user is authenticated and a non-authentic user is not authenticated. The size of base model and user model which result in a given authentication accuracy are aligned with that authentication accuracy on the horizontal axis in the chart.

To establish that the results in Figure 2 hold for large numbers of comparisons many more comparisons where done around the area of best results. The results of these tests are presented in Figure **??**. For the results depicted in the figure, user_model_size is held constant at ten thousand while auth_model_size is varied. The test was performed in this way because variations in auth_model_size seems to have a greater impact on authentication accuracy than variations in base_model_size. Approximately the same trend as seen in Figure 2 presents itself in Figure 3.

9. CONCLUSIONS

This paper presents an approach toward continuous authentication which utilizes the variability in the way users interact with the touchscreens of their devices to differentiate

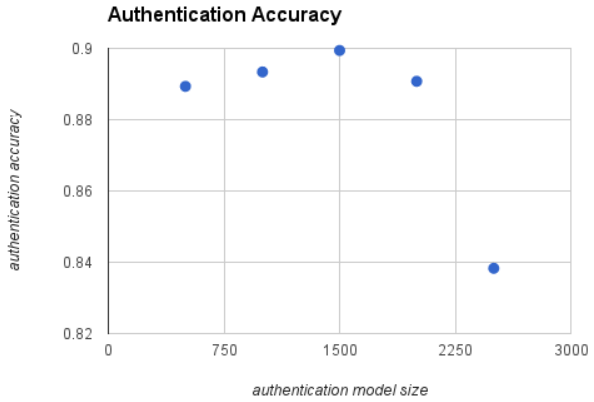

Figure 3: Depicts the result of many model comparisons done around the area of best results in Figure 2.

distinct user-device combinations. We model these interactions with an n-gram Markov model which allows us to describe the likelihood that two sequences of touch pressure values came from the same user in a probabilistic fashion. Data for this approach comes from the user's interactions with the touchscreen of their device. This data will be generated over time by the user; thus this scheme lends itself very well to a continuous authentication model.

Depending on the implementation of this system, varying the parameters of the modeling system can allow the implementer to tune the system to their specific purpose.

10. FUTURE WORK

11. REFERENCES

[1] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. K. Nguyen. Continuous mobile authentication using touchscreen gestures. In *Homeland Security (HST), 2012 IEEE Conference on Technologies for*, pages 451–456. IEEE, 2012.

[2] W. Shi, F. Yang, Y. Jiang, F. Yang, and Y. Xiong. Senguard: Passive user identification on smartphones using multiple sensors. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*, pages 141–148. IEEE, 2011.

[3] J. Zhu, P. Wu, X. Wang, and J. Zhang. Sensec: Mobile security through passive sensing. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 1128–1133. IEEE, 2013.