

Continuous Transparent Authentication with User-Device Physical Unclonable Functions (UD-PUFs) based on Mobile Device Touchscreen Interactions

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

Ian T. Richardson
Electrical & Computer
Engineering
Iowa State University
Ames, IA, USA
ian.t.rich@gmail.com

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

ABSTRACT

A mobile device user continually interacts with many sensors through the natural user interface (UI) of apps. These interactions are unique for each (user, device) pair forming a user-device biometric. A physical unclonable function (PUF) can be realized from the touch screen pressure variability. We illustrate how a sequence of these pressure values from discrete touchscreen interactions may be used to uniquely characterize a user-device pair. These touch screen interactions Markov models can be integrated into a continuous authentication layer. Based on the most recent sequence of touch screen interactions, the continuous authentication layer can assign a probability that these interactions came from the authenticated (user, device) pair. Continuous authentication may help protect access to a mobile device from a malicious party by detecting the anomalies early. The effectiveness of this scheme is described in terms of how definitively can one user 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 hold sensitive data such as financial and health care data. Securing mobile devices poses unique challenges and opportunities. Traditional user authentication is a single challenge-response model based on a password. A mobile device, however, has a greater number of available device sensors that have been used to capture the uniqueness of a specific user, device interaction for enhanced traditional user authentication. Traditional user authentication however leaves large gaps in between successive user authentications. If the device is compromised in between two successive user authentications, there is a long time available to the adversary for inflicting damage. This paper proposes a continuous authentication framework in which an authentication framework running in the background is continu-

ally monitoring user activities. Moreover, this monitoring is based on the combined biometric of the user and the device silicon captured as a physical unclonable function (PUF).

Traditional physical unclonable functions (PUFs) which generate a unique signature of a given hardware device are not sufficient to guarantee the identity of a user. A mechanism that combines the biometric identity of a human user and the device silicon in a way that is not mathematically separable is more robust in the mobile world. UD-PUFs (user-device entangled physical unclonable functions) [?] serve just that role. Integration of such UD-PUFs into a continuous authentication environment is the main problem addressed in this paper.

A UD-PUF is a function of both the device silicon and the user behavior. Such a function must show significant change in the output given a small change in any component of a user-device pair somewhat like a hash function. Hence the UD-PUF must be based on a property or properties which vary significantly and identifiably among mobile device hardware and users. The best candidates to use will be properties which offer the most variability, and properties which are most easily exposed in the android operating system.

System efficiency and response times imposes additional constraints on properties to base a UD-PUF on. 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 intensive to be run on a mobile device, or the system has low accuracy then it is useless for any practical application.

In the continuation authentication world, a composite sensor vector [3] has been used to establish a user identity. Other user biometrics such as inter key stroke timing have also been used. In this paper, the continuous authentication is based on a combined user-device identity. The continuous authentication signature will fail if either the biometrically correct human being or the biometrically correct device component

is removed. This paper builds the continuous authentication framework on the user touch screen interactions. We establish that the user touch screen interactions do differentiate one user from another. Note that there are advantages to basing a framework on a single source of information. A failure of the touchscreen to function also constitutes device failure. This system is more robust because it will only fail upon failure of the device touch screen. Compare this to other systems whose functionality depends on many components. A failure in any one of these components disrupts the authentication scheme making these systems more prone to failure.

Continuous authentication frameworks' basis premise is that a user behavior over time gravitates towards predictable. It can be frequently modeled as an n -Markov model. It states that the user tokens of length n repeat themselves with certain frequency. Hence if we can record history of n -token sequences, they could help us classify the user's current behavior. The tokens are touch screen pressure values that are continually generated as the user interacts with an app through a soft keyboard.

Consider an authentication scheme which seeks to achieve authentication over the duration of a device's use, but requires that some action must actively be taken by the user. Say that this system requires the user to complete some gesture in response to an image on the screen. This sort of authentication may interrupt the user during important tasks, or otherwise cause the device to be more difficult to use. This may lead to great frustration on the part of the user at having to reauthenticate. This scenario exposes a clear benefit of continuous authentication which runs in the background. If the authentication system utilizes actions which are already being performed by the user to perform the authentication then the user can go about their business completely unaware and uninterrupted by the authentication system. This means additional security is possible without causing undue stress on the user or making interaction with the device an unpleasant experience.

The structure of the paper is as follows. Section 2 discusses related work. Touchscreen pressure and the corresponding UD-PUF model are described in section 3. The n -Markov model and its parametrization are discussed in section 4. Data collection methods are discussed in section 6. The authentication scheme is discussed in section 5. The results including final numbers and suggestions in practical applications are articulated in section 7. Conclusions are presented in section 8. Finally section 9 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 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 notable aspect of the SenSec system is their method of modeling users with an n -Markov model. This system presents several benefits which include relative simplicity and scalability. [3]

[2] [1]

3. TOUCHSCREEN PRESSURE

Many mobile devices today are equipped with capacitive touchscreens which have the ability to approximate, among other metrics, the pressure of a given interaction with the touchscreen. [?] When a user interacts with the touchscreen of their device they typically put their finger or fingers on the screen in order to indicate what operation they want the device to perform. This contact (or near contact) functions to increase the capacitance of the screen in a predictable way. The touchscreen is split into many rows and columns; capacitance is measured indirectly as a function of the current increase in the row and column containing the touch.??

If our goal is to 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 the touchscreen for a given touchscreen interaction. [?] This measurement is a function of both the device silicon and the way the user interacts with the touchscreen making this measurement a reasonable choice for use in our system. This is the value which will serve as the basis for our scheme and will henceforth be referred to as touch pressure.

The pressure function is significant because its 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.

In our system, we record the touch pressures generated by the user when they are using the system's soft keyboard. Once we have collected a sufficient number of touches, authentications may begin to be performed on new data. Figure 2 demonstrates that as few as 6000-8000 touches may be used to achieve accuracies higher than 80%. A typical user can enter information into a device's soft keyboard at a rate of xx words per minute. ???. A word is on average five letters long leading to a rate of $xx \times 5$ touch interactions per minute. This means in the average use case a user will generate enough information to begin being authenticated in about x minutes of continuous use. Many applications utilize this as a method of input meaning that in the average use case there will be a constant influx of new data to be analyzed.

4. MODELING A USER-DEVICE PAIR

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 separate Markov model is then constructed from a sequence of touches which need to be compared to the model. The model coming from the later sequence of touches is created with the same distribution as the base model. This affects the n -sized touches sequences which are added to the authentication model. Sequences containing touches which fall outside the base model's distribution are thrown out. The probabilities computed for each of these models are then compared and a percent difference is derived from these comparisons. We then choose to authentication only those models which have achieved a low enough percent difference.

In comparing the base and auth Markov models each n -token sequence in the auth model is compared to the base model. For a given sequence in the auth model the base model is searched for a matching sequence. If a matching sequence is not found then the sequence in the auth model being considered is decided to be maximally different. If a matching sequence is found then the absolute value of the difference between probabilities computed for that sequence are considered to be the difference. The aggregate difference is the average of these difference values.

6. DATA COLLECTION AND ANALYSIS

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.

The results presented here were derived from four user-device pairs each having a large number of touches. Each user was asked to use two different devices to collect data. The size of the data sets collected was approximately 20000-25000 for each user-device pair.

7. RESULTS

We predicted that an increase in the number of touches used to build the model would provide better accuracy in authentications. In addition,

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

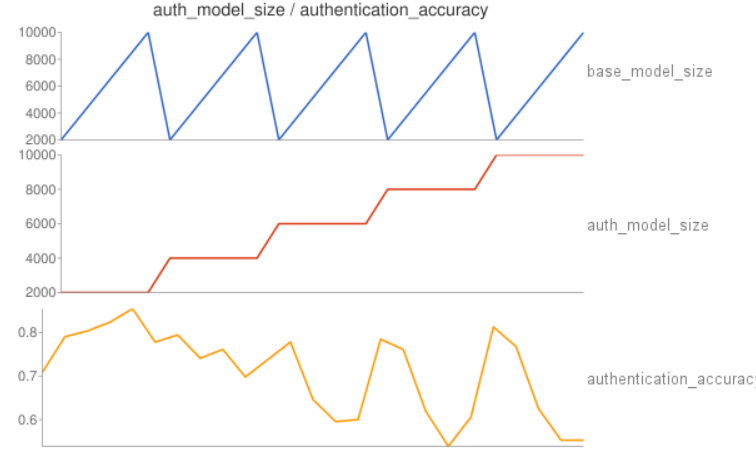


Figure 2: Authentication accuracy is a function of both the base model size and authentication model size.

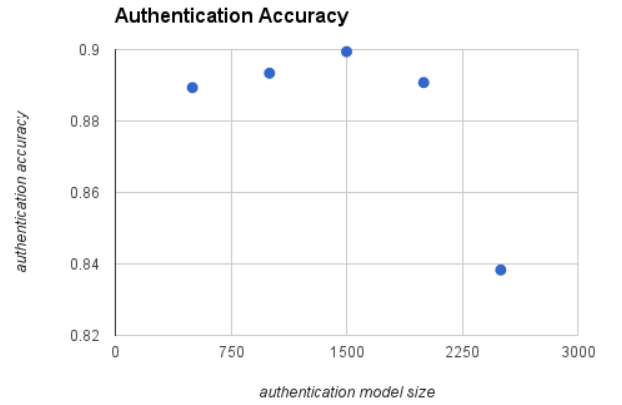


Figure 3: Depicts the result of many model comparisons done around the area of best results in Figure 2.

which result in a given authentication accuracy are aligned with that authentication accuracy on the horizontal axis in the chart.

In some instances, increased numbers of touches did not result in higher authentication accuracies. These lower accuracies seem to correspond to areas where This could be a result of the way the model decides to include n -sized touch sequences in the authentication. In this case, an auth model which is different will authenticate with low probability because sequences are punished for not existing in the base model if they exist in the auth model. If there are not many n -token sequences in the base model then high differences will be represent for models which are only slightly different, because many sequences will not exist in the base model. This sequences will be considered maximally different and constitute a high proportion of the number of n -token sequences.

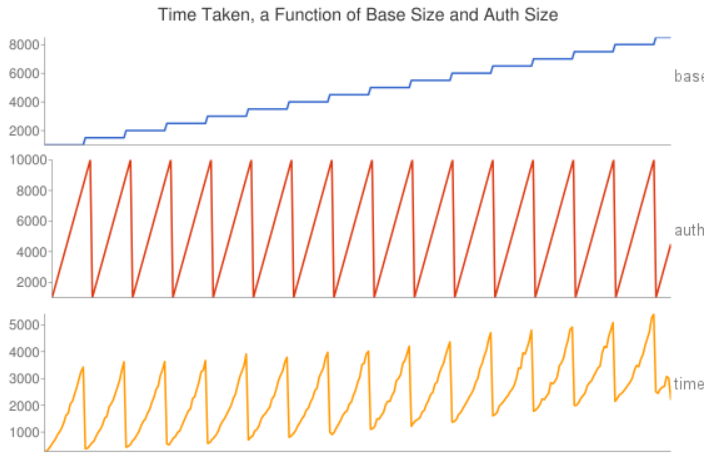


Figure 4: Depicts the time taken on a Nexus 7 tablet as a function of base model size and auth model size. Time taken is measured in milliseconds while model sizes are measured by number of touches.

To establish that the results in Figure 2 hold for large numbers of comparisons many more comparisons were 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.

Figure 4 depicts the runtime of our system on a Nexus 7 tablet. Time taken is measured in milliseconds while model sizes are measured by number of touches used to construct the model. The time taken metric does not include the overhead associated with adding touches to either the base or auth models; it is assumed this will be done over time as the user enters data. In addition, adding touches is not a computationally intense activity as it is designed to be done in constant time. Time taken does include the probability computation for each of the models and the comparison between the models. This chart is a good representation of how each of the model sizes affect the runtime of the system and allows for the identification of an overall trend in the system's runtime. The overall time taken is trending upward as both base model size and auth model size increase; this suggests that the total number of touches used, that is the sum of base model size and auth model size, might be the most important factor in determining the runtime. The dependence on the total number of touches makes sense because the majority of the computations are used to determine the probabilities of the Markov model. A greater number of touches in a model requires more of these probability computations to take place. Also of note, base model size and auth model size seem to affect time taken differently. A change of k in the size of auth model seems to increase the total amount of time taken more than the same k -sized change in the base model.

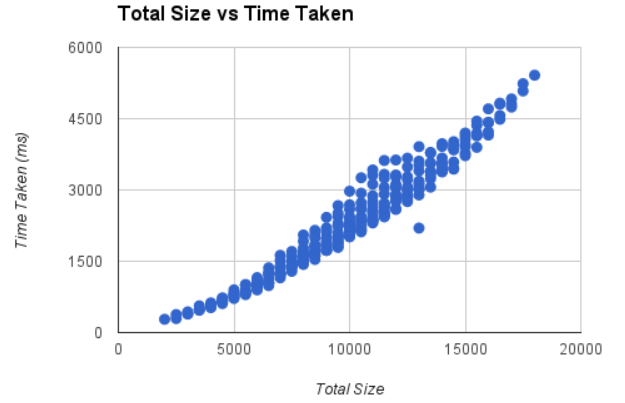


Figure 5: Depicts the time taken on a Nexus 7 tablet as a function of base model size plus auth model size. Time taken is measured in milliseconds while model sizes are measured by number of touches.

Figure 5 illustrates how the time taken depends on the total number of touches used in creating the models. That is to say total size is the sum of base model size and auth model size. The trend in the model suggests that the total amount of time taken will increase exponentially as the number of touches used to generate the model increase. This figure also supports the conclusion that the total size of the model in number of touches has a larger influence on the runtime than the sizes of either the base or auth models individually; in general, more touches will lead to an increase in run time.

Figures 2 and 3 establish that in general a greater number of touches used in the authentication will result in a greater accuracy. This manifests in the charts as the peaks indicating the highest authentication accuracy existing around areas where the largest numbers of touches are used to construct the models. Figures 4 and 5 demonstrate the performance tradeoff associated with using increased numbers of touches. As expected there exists an inverse relationship between performance in terms of speed and accuracy of authentication. That is, increased authentication accuracy comes at the expense of execution speed.

8. 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 distinct user-device combinations. This is useful in situations similar to theft where a mobile device is accessible to a party which may desire to use the device to compromise sensitive data. One shot authentication systems like passwords are not the ideal solution in this case. If the password has also been compromised or is weak such that it may be easily broken, then the attacker has access to the device indifferently. In the continuous authentication model presented here, if the device is compromised then the way the attacker interacts with the device will deviate away from the model which has been established for the authentic user. After the the newest sequence of touch interactions has become signif-

icantly different from the established model, then the device may be locked or some other arbitrary action taken.

The implementation of our system has both modeling and authentication components. The modeling component depends on the use of an n -Markov model constructed from a sequence of touches entered by the user. This allows for the likelihood that two sequences of touch pressure values came from the same user to be calculated in a probabilistic fashion. The authentication system constructs two models. One of the models coming from the most recent touches and the other coming from touches preceeding the most recent. A difference is derived between the models which is a function of the probability derived in the modeling phase. If difference is sufficiently large it can be said with reasonable certainty that the user from which the touch sequences originated for the most recent touches differs from the user who generated the touches preceeding those most recent.

Data for this approach comes from the user's interactions with the soft keyboard 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. In addition, this data is reliable over the lifespan of the device. The authentication system will persist as long as the touchscreen functions correctly. A failure of the touchscreen renders the mobile device useless; hence the system is more robust compared to systems which depend on multiple sensors contained on the mobile device. Mobile device touchscreen interactions are also ideal for use in an authentication scheme. They function both as a biometric of the human user and the silicon of the device; a change in either device or user will be detected as a result.

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.

9. FUTURE WORK

10. 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.