Tapped-based Authentication for Mobile Device Security

Lukasz Brodowski
54 Bradley Street A4
Plainville, CT 06062
(860) 970-6299
brodowski@my.ccsu.edu

Cameron Dziurgot
2 King Arthur Way Apt. 1
Newington, CT 06111
(203) 903-6265
cameron.dziurgot@my.ccsu.edu

Donald Moretz
(860) 882-4834
travis.moretz@gmail.com

ABSTRACT

Passkeys have been around since personal phones have existed. The typical numeric PIN offers a limited number of combinations and is relatively easy to crack by guessing or eavesdropping. A tap sequence used as a passkey offers an infinite number of different combinations making it unique to that person and that person alone. Once a passkey is saved, it can be compared using the Euclidean distance formula to any other passkey resulting in a dissimilarity score to the master key. If this score is within a certain threshold, that passkey will grant or deny the user access. Our results show that it is possible to authenticate using a unique tap sequence, and convenient enough to be used every day to unlock a personal device.

Keywords

Authentication; Security; Mobile devices; Tapping;

1. PROBLEM AND MOTIVATION

With the ever increasing number of smartphones, tablets, and touch screen device, users are becoming more concerned with safely securing their data. Having a passkey to open a device is only one level of security. Many devices only allow for a four-digit passkey which could result in 10,000 different passkey combinations. If an unsuspecting user is not careful when unlocking their device, the passkey can easily be seen by onlookers and unauthorized access can be granted to your device.

A passkey can be much more complex by adding not just the passkey tapped, but the duration of the presses, the duration of the gaps between the presses, and the area of the screen that one's finger contacts when pressing the screen. This results in a much harder time for an unauthorized user to gain access to your device. By recording all of these variables and storing them as attributes of a master key, the degree of complexity of a passkey would provide a much higher level of security. Even if someone saw the user tapping their sequence, they would have a much harder time replicating the exact presses. This would make every passkey unique to the given user, adding another level of security and preventing over the shoulder attacks as well as many other types of attacks. This type of passkey is also practically impossible to brute force since there are infinite possibilities in terms of tap sequences a user can choose.

2. BACKGROUND AND RELATED WORK

Passkeys have been around since the personal phone existed, the one that is most commonly used being a PIN (Marques, 2013). The idea of a tapping pattern being used as a way of authentication for smartphones has been described in an article named TapSongs in 2009[2]. In this study, only a binary input was used for the tap sequence. Adding more variables such as location on screen, area, and pressure can make it even more complex.

A tapping sequence is a set of timestamps recorded by the application. The first timestamp is recorded during the initial first press, and it acts as a zero. All other timestamps are relative to this initial press. The minimum requirements for tapping-based authentication include recording the relative time of the presses up/down, as well as coordinates of the press. This data is enough to calculate the interval of the press, as well as the interval between two presses. More information can be obtained from the press event, such as area and pressure adding more complexity to the vector.

One approach to comparing the two data vectors of the tap sequences is to use the Hamming distance, which measures the minimum number of substitution required to change one vector to the other[3]. For our application we used a variation of the Euclidean distance formula which is the sum of the squares of data sets. Our algorithm also took into account a standard deviation which allows for a margin of error when a user enters their passkey. A dissimilarity score is given from this algorithm which tells us how far the individual points on the two vectors from each other. Based on this score we can decide whether the correct sequence was entered.

3. APPROACH AND UNIQUENESS

A prototype of our tap sequence passkey authentication application was implemented in Java for an Android device. This lets us test the capability of mobile devices and their ability to quickly run an algorithm to compute the degree of dissimilarity between two different vectors. This allowed us to verify the practicality and usability of this approach while reliably rejecting impostors. Since the passkey only has to be compared to one master key, we found this to be a pretty fast feedback loop.

This type of approach is unique because every single passkey is distinct to its user due to micro variations and there are infinite possibilities to what type of sequences can be tapped because of this. Security not only comes from this, but if an onlooker was able to get ahold of the tap sequence they would have trouble replicating the exact sequence.

4. RESULTS & CONTRIBUTIONS

Our results show that all the different factors of a tap sequence (time pressed down, time between taps, and coordinates of each tap) make it extremely hard to replicate the exact same tap sequence. This is because of the micro changes in the tap sequence that even humans cannot control. Therefore, to make it easier we removed the coordinates of each tap from the algorithm. This allowed us to match the master sequence more reliably while still keeping the same acceptance rate.

When a user registers their master tap sequence, they enter it three times. The application then calculates a standard deviation between the three sequences. Once this average master sequence is created it can be used to compare it to any trial sequences. The algorithm that we use returns a dissimilarity score, which indicates how close the master sequence and the trial sequence are. The closer the score is to zero, the greater the similarity between the two sequences. This algorithm becomes even more efficient when you take into account the edge cases of where the number of taps between the master and trial sequence are not the same. It was found that more complex tap patterns such as a beat to a user's favorite song where much harder to replicate while still being relatively easy for the master user to authenticate.

After many more tests the next step that we took was to start logging the statistics of each trial or attempt. This let us calculate rates of false acceptance and false rejection of the authentication method. Gathering these statistics will better improve the acceptance rate by allowing us to tweak the similarity score acceptance to either be more generous or less forgiving. After many trials and many tweaks, the system proves to be a reliable way of verification for the average user.

5. ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

6. REFERENCES

[1] Marques, D., Guerreiro, T., Duarte, L., Carriço, L. (2013). *Under the table: tap authentication for smartphones.* In Proc. of BCS HCI '13, Swinton, UK

[2] Wobbrock, J.O. (2009). *TapSongs: tapping rhythm-based passwords on a single binary sensor*. In: Proc. UIST '09. ACM Press. 93-96.

[3] Norouzi, M., Fleet, D. J., & Salakhutdinov, R. (2012). *Hamming Distance Metric Learning.* Advances in Neural Information Processing Systems 25, 1-9.