Gesture-Based Peer-to-Peer Pairing Authentication for Assymmetric Internet of Things Devices

Joe Chen
joe.chen@rice.edu

Zilong (Gino) Liao
zl15@rice.edu

Heng-Yi (Henry) Lin
henry.hy.lin@rice.edu

1. INTRODUCTION

In an Internet of Things (IoT) environment, mobile devices may need to pair or authenticate themselves to other devices. However, unlike the traditional internet, an IoT environment does not typically have a centralized certificate authority, making it difficult for one device to determine if another device is authentic. Furthermore, these IoT devices are often resource-constrained, meaning traditional cryptographic defenses that support confidentiality, integrity, and authenticity difficult to implement [6, 20].

One potential solution to this problem is to use biometrics—especially motion and gestures—in order to validate the identity of the device. Prior work has shown that impostors has a low probability of imitating a gesture calibrated to another person successfully [4]. Furthermore, motion recognition is suitable for IoT systems which feature small sensors and low powered devices because motion recognition can achieve high accuracy with just an accelerometer [18].

Some prior work have looked at motion sensor data fusion across different devices to detect pairing, for example detecting device collision when tiling two tablets together [7]. Existing work in gesture recognition and event detection focuses on gestures on the same device or similar devices (e.g. two tablets, gestures on a Wiimote [13]). However, pairing in an IoT environment is usually needed for two asymmetric devices with different types of hardware sensors (e.g. a smartwatch with a smartphone).

In this project, we analyze the use of gestures as a biometric for peer-to-peer authentication in an IoT scenario, where sensor devices are asymmetric.

2. ATTACK & DEFENSE MODELS

Our project analyzes the following three key defense models for authentication.

- *Model 1:* This system model contains three entities: a legitimate prover, a verifier, and an attacker. The legitimate prover is non-malicious and wants to pair with the verifier. Both the legitimate prover and the verifier are owned by the same person (e.g. a smartphone and a smartwatch). However, the attacker is a malicious prover and wants to also pair with the verifier. To distinguish between the attacker and legitimate prover, the verifier uses gesture recognition to distinguish between the two parties.

  Since the legitimate prover and verifier are owned by the same person, the user performs any generic gesture while holding both devices as shown in Figure 1. Accelerometer data is used to read the gesture, and a matching gesture authenticates the prover.

  In contrast, the attacker must mimic the gesture of the verifier in order to trick the verifier. Our hypothesis is that we can keep false negatives (rejecting legitimate provers) and false positives (accepting attackers) to less than 10%. This is modest compared to existing works due to the hardware asymmetry.

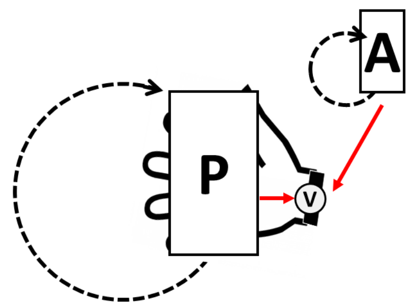

Figure 1: System model for defense model 1. The verifier (V) and the prover (P) are synchronized in motion. The attacker (A) must try to mimic the motion of the verifier in order to trick the system.

- *Model 2:* This system model contains the same three entities as in Model 1. In this model, the verifier has previously calibrated a single secret gesture with the user that is needed to pair with the device as shown in Figure 2. Devices that want to pair with the verifier must produce accelerometer data that matches this gesture with no prior knowledge about the gesture. Since this gesture is a predefined secret, only the prover needs to collect accelerometer data during the proof of authenticity.

  An attacker may try to trick this model in the follow-

ACM ISBN XXXX.
DOI: XXXX

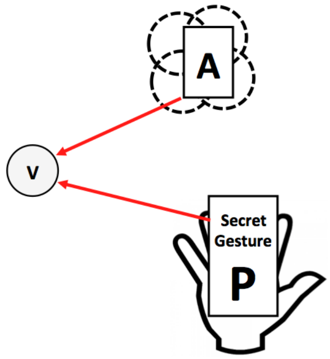

Figure 2: System model for defense model 2. The verifier (V) and the prover (P) are not synchronized in motion, and the verifier has previously established a secret gesture for pairing. The attacker (A) can brute force a gesture if it has no knowledge of the secret gesture, or it may imitate a gesture that it sees that a legitimate prover used.

ing two ways. First, if no information about the secret gesture has been leaked to the attacker, it will attempt a brute force attack and attempt several common gesture shapes, such as a circle, line, or even just shaking the device. Second, an attacker may learn information about which gesture is the secret gesture by watching a legitimate prover validate themselves to the verifier. These visual clues then reveal what the actual gesture is, and the attacker scenario reduces to the same as in Model 1.

For the second attack, we hypothesize that we will again achieve less than 10% false negatives for legitimate provers. However, we hypothesize 0% false positives for attackers if the secret gesture is not within the library of brute force attempts (i.e. the gesture recognition algorithm works).

- *Model 3:* The final model has the same three entities. The user has already successfully paired one device with the verifier and sent training data of a library of gestures to the verifier. This library of gestures is collected at the already-paired device and simply stored at the verifier. In order to pair a new device to the verifier, the prover must recreate these gestures with the new device.

  During the pairing process, the verifier will challenge the prover to a subset of the gesture library. As shown in Figure 3, the verifier sends visual prompts about what the gestures should be during the challenge process (i.e. the library of gestures is public). However, to trick the system, an attacker must produce the gesture in the same way as the intended user. The attacker may use the same device as the user or different devices. Even with device asymmetry, we want attacker to be denied and the actual user to be authenticated.

  We hypothesize that there is a inverse correlation between number of gesture challenges and number of false positives (i.e. more challenges reduces the success of an attacker. However, we also hypothesize there is a direct correlation between number of gesture challenges and number of false negatives (i.e. there are more opportunities for the user to fail). We seek to find a threshold that minimizes the number of false positives and false negatives in this model.

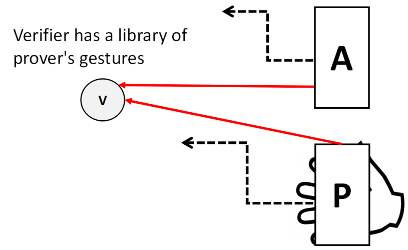

Figure 3: System model for defense model 3. The verifier (V) and the prover (P) are not synchronized in motion, and the verifier challenges the prover with a series of gestures in its pre-calibrated library. The attacker (A) receives the same prompts, but must mimic what the prover would do for each gesture.

2.1 Experimental Platform

Our experiments focus on three different mobile devices with 3-axis accelerometers: iPhone 6 (iOS), Nexus 5 (Android), and a Nintendo Wiimote. For ease of implementation, each device communicates with a laptop running MATLAB, and the laptop takes and compares the accelerometer data from each device.

The iPhone and Nexus devices communicate with MATLAB via WiFi through the MATLAB sensor hardware support package (iOS[1], Android[2]), and the Wiimote communicates with MATLAB via Bluetooth through an open source program called WiiLab[3].

For gesture recognition, we implement uWave [12, 13], which was developed in the Rice Efficient Computing Group. The algorithm uses dynamic time warping to obtain the distance between two time series accelerometer data to characterize how closely two gestures match. Their algorithm simplifies the time series data such that even simple 16-bit microcontroller can do the computation. The uWave authors have provided their original source code in C, and we have converted the gesture recognition modules into MATLAB implementations.

3. PROGRESS

3.1 Current Accomplishments

- Successfully extracted accelerometer data from all three mobile devices
- Basic gesture functionality of uWave converted to MATLAB programming environment.

[1]http://www.mathworks.com/hardware-support/iphone-sensor.html
[2]http://www.mathworks.com/hardware-support/android-sensor.html
[3]http://netscale.cse.nd.edu/twiki/bin/view/Edu/WiiMote

Pairer \ Gesture	Letter ‘e’	Letter ‘s’
Gino	383	433
Henry	884	3304
Joe	3300	5574

Table 1: DTW distance from the original calibration time sample.

- Initial distance measurements between non-malicious and malicious provers for gesture recognition on a signle device.

3.2 Future Milestones

- *April 6:* Rigorous Model 3 experiments completed. Scripting for Model 1 completed (remove axis/orientation bias).
- *April 13:* Rigorous Model 1 & 2 testing completed.
- *April 22:* Finalize project report/conduct follow up experiments.
- *Final Exam Period:* Final project presentation

3.3 Generalized Division of Labor

- *Joe:* Lead for MATLAB software development and wiimote integration.
- *Zilong:* Lead for Android software development.
- *Heng-Yi:* Lead for iOS software development.

4. EXPERIMENTAL RESULTS

4.1 Gesture Recognition on a Single Device

As an early proof-of-concept, we first test our assumption that we can detect when different users are performing the same gesture. All three team members perform the same gesture with the same iPhone device. We classify Gino as the legitimate user, while Joe and Henry are the attackers. Our two test gestures are an ‘e’ and ‘s’ from the lowercase English alphabet. Since this is an early experiment, each user only takes one accelerometer measurement for each gesture, but more data will be collected for the final paper.

After the data is collected, we run our implementation of the uWave algorithm. We first quantize the 3-axis acceleration data and then calculate the dynamic time warping distance between the two time series to check if the gestures match.

Table 1 shows the results for the three users. As the legitimate user, Gino’s gesture difference is smallest out of all the users. Henry is able to achieve a small distance for the letter ‘e’, but the distance is still two times larger than Gino’s. Joe’s distance is an order of magnitude larger than Gino’s for both gestures. These early results support the idea that we can detect when different users are making the same gesture. However, we plan to conduct more rigorous experiments in the coming weeks.

5. RELATED WORK

Our related work is divided into three key categories: (1) gesture recognition algorithms and implementations on a single accelerometer device, (2) device pairing via accelerometer data, and (3) other biometric recognition authentication techniques.

5.1 Gesture & Motion Recognition on a Single Device

Several efficient gesture-recognition algorithms already exist. For accelerometer-based gesture recognition, Ali et al. [2] introduce and evaluate five cutting-edge algorithms, including Dynamic Time Warping (DTW), Hidden Markov Model (HMM), Support Vector Machine (SVM), K-Nearest Neighbor (k-NN), and Artificial Neural Networks (ANN). They evaluate these algorithms in both user-dependent and user-independent cases, and their result shows that DTW and k-NN achieve the best accuracy among other algorithms in both cases.

Jiayang et al. [12, 13] present an algorithm called uWave which is based on a single accelerometer. uWave quantizes the acceleration data to reduce computational load and uses DTW to measure similarities between two time series of accelerometer data. Template adaptation deals with gesture variation over the time. To enhance the accuracy of user-independent recognition for some scenarios, Hussain and Rashid [8] propose an enhancement based on modified uWave algorithm with hardware-accelerated DTW. Their implementation decouples the axes of gesture data while performing DTW calculation to reduce the mismatch due to the variation of temporal speeds on different axes between two users.

Aside from uWave, Ahmad and Shahrokh [1] also propose a gesture recognition system that uses only one 3-axis accelerometer. The system temporally compresses the acceleration time series to filter out variations not intrinsic to the gesture itself and reduces the size of the acceleration signals for next step of dynamic time warping. Then, the system uses affinity propagation to find a good set of exemplars from all data points. Finally, they implemented compressive sensing to recognize a repetition of a gesture.

For the best user experience, gesture recognition should be in real-time and easy to use. Instead of using a button to indicate start time and end time of a gesture motion, Zoltan [15] proposes an automatic segmentation method and uses two classification algorithms: HMM and SVM to to give high accuracy. This system has great performance and low response time.

Considering the quadratic time and space complexity with DTW algorithm and the need of larger training sets with HMM, Zhe Ji et al. [10] propose a new algorithm which uses FastDTW instead of DTW and HMM. Their algorithm is divided into two parts: preprocess and classification. In the preprocessing step, the raw data is first filtered by a series of low-pass filters and its amplitude is normalized. Then, it is resampled to a fixed length. After preprocessing the raw data, FastDTW, which has linear time and space complexity, is used to calculated the alignment between two time series. Their work shows that the performance drop is not necessary and is avoidable while replacing lightweight FastDTW to DTW for reducing computational demand.

5.2 Device Pairing via Sensor Data

In addition to recognition of a gesture, gesture-detection can also be used as a form of multifactor authentication to pair two separate devices together. Hinckley proposed one of the earlier forms of synchronous gesture authentication. By detecting an impulse when two tablets are pushed together, Hinckley pairs the two tablets, allowing the user to tile both devices together as one large screen [7]. Vinteraction uses a combination of accelerometer and vibrator data to transmit private data between two devices in physical contact. The vibrations serve as the secret shared channel between the two devices [19]. Mayrhofer et. al. have a user hold two mobile devices and shake randomly to establish a shared secret key. The shaking motion produces enough entropy to create a key that is difficult to predict [14].

Jiang et. al. propose near-field vibration (NFV) to group multiple devices together at once. By propagating the vibrations of a smartphone through a table on which all group devices are placed, the smartphone can automatically pair with all of the devices in the group [11].

In a non-security based scenario, Duet explores combining sensor information for both a smartphone and a paired watch to create more sophisticated controls based on hand gestures [5]. PickRing compares gyroscope data across a ring and a gyroscope to detect when a user picks up a smart device. However, the authors do not analyze the security of this approach against a malicious prover [17].

5.3 Other Biometric Recognition Authentication Techniques

Besides motion and gesture, various biometric characteristics could also be used for recognition as Jain et al. [9] specified. These characteristics include: DNA, ear, face, facial thermogram, hand thermogram, hand vein, fingerprint, gait, hand geometry, iris, palmprint, retina, signature, and voice. In some applications, biometric traits are further exploited for machine-to-machine recognition authentication instead of conventional personal recognition.

One example of this idea is an authentication scheme based on heartbeat data (ECG) proposed by Rostami et al.[16]. It requires that the controller of implantable medical devices (IMDs) contacts the patient's body to control the IMD. Specifically, the controller can only gain access to IMDs if the ECG readings on the both devices are approximately the same.

Furthermore, a patent [3] submitted by Apple Inc. depicts a more general authentication scheme using biometric data for wireless pairing and communication between devices. The scheme is simply based on the comparison of biometric data which received and stored by the device or the host. Thus, it is applicable with any sort of distinctive biometric traits for machine-to-machine authentication once they embed related module targeting to any specific trait on their commercial devices.

6. REFERENCES

[1] A. Akl and S. Valaee. Accelerometer-based Gesture Recognition via Dynamic-time Warping, Affinity Propagation, & Compressive Sensing. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, Dallas, USA, March 2010.

[2] A. H. Ali, A. Atia, and M. Sami. A comparative study of user dependent and independent accelerometer-based gesture recognition algorithms. In *Distributed, Ambient, and Pervasive Interactions, 2014 International Conference on Human-Computer Interaction*, Crete, Greece, Jun 2014.

[3] Apple Inc. Wireless Pairing and Communication Between Devices Using Biometric Data. *US Patent US 20140068725 A1*, Mar 2014.

[4] J. G. Casanova, C. S. Avila, A. de Santos Sierra, G. B. del' Pozo, and V. J. Vera. A Real-Time In-Air Signature Biometric Technique Using a Mobile Device Embedding an Accelerometer. In *Proceedings of the Conference on Networked Digital Technologies*, Communications in Computer and Information Science, Prague, Czech Republic, Jul 2010.

[5] X. A. Chen, T. Grossman, D. J. Wigdor, and G. Fitzmaurice. Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*, New York, NY, 2014.

[6] Cisco Systems, Inc. Securing the Internet of Things: A Proposed Framework. http://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html. (Accessed: 2016-02-16).

[7] K. Hinckley. Synchronous Gestures for Multiple Persons and Computers. In *Proceedings of the Symposium on User Interface Software and Technology (UIST '03)*, Vancouver, Canada, 2003.

[8] S. M. A. Hussain and A. B. M. H. Rashid. User independent hand gesture recognition by accelerated dtw. In *Informatics, Electronics and Vision (ICIEV), 2012 International Conference on*, Dhaka, Bangladesh, Apr 2012.

[9] A. K. Jain, A. Ross, and S. Prabhakar. An Introduction to Biometric Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, Jan 2004.

[10] Z. Ji, Z.-Y. Li, P. Li, and M. An. A new effective wearable hand gesture recognition algorithm with 3-axis accelerometer. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2015 International Conference on*, Beijing, China, Aug 2015.

[11] Z. Jiang, J. Han, W. Xi, and J. Zhao. NFV: Near Field Vibration Based Group Device Pairing. In *Proceedings of the International Conference on Collaborative Computing: Networking, Applications, and Worksharing (CollaborateCom '15)*, Wuhan, China, Nov 2015.

[12] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. User Evaluation of Lightweight User Authentication with a Single Tri-axis Accelerometer. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*, Bonn, Germany, Sep 2009.

[13] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based Personalized Gesture Recognition and Its Applications. *Pervasive and Mobile Computing*, 5(6):657–675, Dec 2009.

[14] R. Mayrhofer and H. Gellersen. Shake Well Before Use: Authentication Based on Accelerometer Data. In *Proceedings of the International Conference on*

Pervasive Computing and Communications (PerCom '07), Toronto, Canada, May 2007.

[15] Z. Prekopcsák, P. Halácsy, and C. Gáspár-Papanek. Accelerometer Based Real-Time Gesture Recognition. In *Proceedings of the 12th International Student Conference on Electrical Engineering*, Prague, Czech Republic, May 2008.

[16] M. Rostami, A. Juels, and F. Koushanfar. Heart-to-heart (H2H): Authentication for Implanted Medical Devices. In *Proceedings of the SIGSAC Conference on Computer and Communications Security (CCS '13)*, Berlin, Germany, Nov 2013.

[17] K. Wolf and J. Willaredt. PickRing: Seamless Interaction Through Pick-up Detection. In *Proceedings of the Augmented Human International Conference (AH '15)*, Singapore, Singapore, 2015. ACM.

[18] R. Xu, S. Zhou, and W. J. Li. MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition. *IEEE Sensors Journal*, 12(5):1166 – 1173, Sep 2011.

[19] T. Yonezawa, J. Nakazawa, and H. Tokuda. Vinteraction: Vibration-Based Information Transfer for Smart Devices. In *Proceedings of the International Conference on Mobile Computing and Ubiquitous Networking (ICMU '15)*, Hakodate, Japan, Jan 2015.

[20] Z.-K. Zhang, M. C. Y. Cho, and S. Shieh. Emerging Security Threats and Countermeasures in IoT. In *Proceedings of the Symposium on Information, Computer and Communications Security (ASIA CCS '15)*, Singapore, Apr. 2015. ACM.