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

ABSTRACT

This is an abstract for the paper. We should summarize the major points, present key results.

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—particularly 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. This asymmetry in device shape leads to different tendencies for holding the devices, and the difference in sensor accuracy and sampling rate also produces potential challenges for comparing gestures. Our key contributions are an experimental evaluation of gesture authentication on various devices including iPhones, Android devices, and a smartwatch.

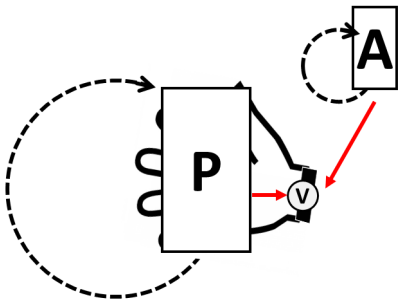


Figure 1: System model for the Simultaneous Gesture defense model. The verifier (V) and the prover (P) are synchronized in motion. The attacker (A) must try to mimic the motion of the verifier in order to trick the system.

The remainder of this paper is organized as follows. We start by defining our system defense and attack models in Section 2. Next, we describe our experimental platform in Section 2.1. Our experiments are divided into two sections based on the defense models: Gesture Library (Section 3) and Simultaneous Gestures (Section 4). In Section 5, we described related work in biometrics for authentication and gesture recognition. Finally, we conclude the paper in Section 6 and also describe future avenues of research for this method of authentication.

2. ATTACK & DEFENSE MODELS

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

- *Simultaneous Gesture Model:* 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

ACM ISBN XXXX.
DOI: XXXX

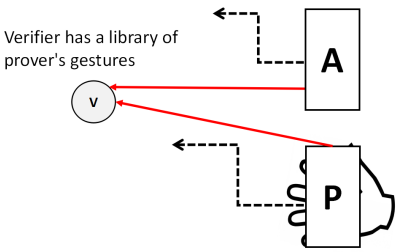


Figure 2: System model for the gesture library defense model. The verifier (V) and the prover (P) are not synchronized in motion, and the verifier challenges the prover with a series of gestures in its pre-calibrated library. The attacker (A) receives the same prompts, but must mimic what the prover would do for each gesture.

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.

- *Gesture Library Model:* 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 2, 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.

2.1 Experimental Platform

Our experiments focus on multiple mobile devices with 3-axis accelerometers: two iPhone 6, one iPhone 6s, Nexus 5 (Android). For ease of implementation, each device communicates with a laptop running MATLAB, and the laptop

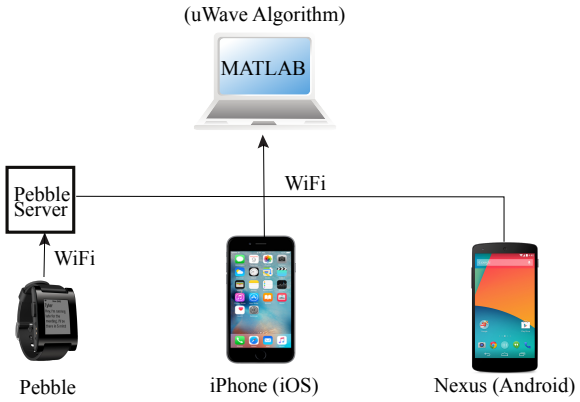


Figure 3: The hardware setup for the gesture-based authentication experiments. The iPhone and Android connect to MATLAB via WiFi. The Pebble smartwatch stores its data on a 3rd party server. The data is retrieved via WiFi and then imported into MATLAB.

takes and compares the accelerometer data from each device. We also compare performance against a classic Pebble smartwatch's accelerometer data. The data flow is shown in Figure 3.

The iPhone and Nexus devices communicate with MATLAB via WiFi through the MATLAB sensor hardware support package (iOS[1] and Android[2]). These devices are used for the Gesture Library model.

For the Simultaneous Gesture model, the Pebble smartwatch records its accelerometer data using one of its 1st party applications called CompassLog. The application sends the accelerometer data to an online server, and we download the data and compare the results against the iPhone/Nexus results in MATLAB. To simulate a powerful attackerl, the data collection for all devices, including the attacker, at the same time.

For gesture recognition, we implement uWave [12, 13], which was developed in the Rice Efficient Computing Group. The algorithm uses Dynamic Time Warping (DTW) 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.

We set each accelerometer to sample as quickly as the hardware allows us to sample—one sample per 100 ms for the iPhone devices, one sample per 20 ms for the Android device, and one sample per 40 ms for the Pebble smartwatch. Because these are relatively slow sampling rates, we choose not to apply any smoothing to the accelerometer data.

3. GESTURE LIBRARY EXPERIMENTS

To test the Gesture Library Model, we collect gesture data from all three group members. Our experimental library

[1]http://www.mathworks.com/hardware-support/iphone-sensor.html
[2]http://www.mathworks.com/hardware-support/android-sensor.html

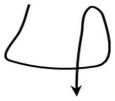
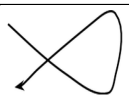
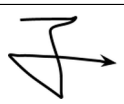
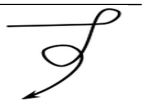
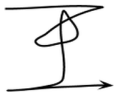
1	2	3	4
			
5	6	7	8
			
9	10	11	12
			
13	14	15	16
			
17			
			

Table 1: Gestures for authentication

consists of the 17 different gestures shown in **??**. These gestures are a mix of European characters, Asian characters, and simple shapes, each with varying complexity.

Gino is the legitimate user who performed calibration on his iPhone 6 device. To emulate a strong attacker, Henry attempts authentication using a second iPhone 6 device. Joe is a weaker attacker who attempts authentication on an iPhone 6s. Joe is less familiar with Asian characters, and thus is weaker than Henry because of the selecting gestures. Gino also attempts authentication on his second device, the Nexus 5, as a legitimate user.

For each gesture, Gino takes 30 calibration attempts on his iPhone 6. Afterwards, everyone attempts 10 new authentications for each gesture each day. In the next sections, we discuss calibration and set thresholds that minimize false negatives (rejecting Gino, the legitimate user), while also minimizing false positives (accepting Henry or Joe, the attackers).

3.1 Calibration Mechanism

Our calibration mechanism is a brute force search that calculates the DTW distance of one attempt time series to another for the same gesture. The calibrated time series selected is the attempt with the smallest average distance to all of the other attempts. This approach is non-scalable ($O(N^2)$ where N is the number of calibration attempts). However, scalability is not the focus of this project because we can make calibration a one-time event at the beginning and choose not to update the history for the purposes of this experiment. For calibration, we use the full 30 attempts per gesture data set unless otherwise noted.

3.2 Gesture Distance Consistency

First, to prove that gesture-based authentication works, we monitor the consistency of the DTW from the calibration time series over the period of 7 days for both the legitimate user and the attackers. We hypothesize that the average distance for all the attempts for a single gesture on a single day from the attackers stays at least one standard deviation away from the average distance for the intended user. Furthermore, we hypothesize that the intended user's average distance should stay within one standard deviation of the previous day's average.

DATA HERE

3.3 Threshold Setting

3.3.1 *Technique*

The threshold for rejecting or accepting a user can be set using a variety of techniques. For this project, we use a simple threshold (D_t) based on the calibration data above as a proof of concept. Each attempt from calibration has an average DTW distance from all of the other calibration attempts. Based on this distance, we can set a threshold tailored to gesture i using Equation 1

$$D_t = m \cdot d_i + \frac{1.96 \cdot \delta_i}{\sqrt{N}} \tag{1}$$

where m is a constant multiplicative factor, d_i is a mean distance value based on the distance data from calibration for gesture i, and δ_i is the associated standard deviation.

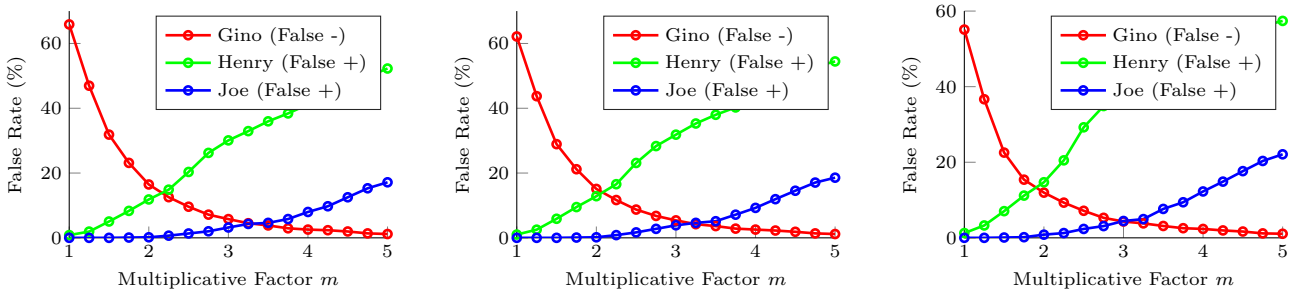
Intuition suggests that choosing the minimum recorded DTW distance for d_i yields an overly strict system that minimizes false positives but also introduces additional false negatives to the system. Likewise, choosing the maximum recorded DTW distance for d_i produces an overly lenient system that minimizes false positives but introduces false negatives.

To make the system more usable, we opt to split the difference to minimize both false positives and false negatives. However, if we must choose between more false positives and false negatives, we choose to err on the side of more false positives to make the system more usable to the end user. Therefore, we set d_i based on the 80th, 85th, and 90th percentiles of the calibration data. This lets us minimize the effects of any major calibration outliers while still being lenient. Furthermore, the multiplicative factor m is used to introduce some additional leniency because of day-to-day variation in the intended user's gesture. m is varied from 1 to 5 in steps of 0.25.

Our hypothesis is that we can set an authentication distance threshold that keeps the false positives and false negatives rates averaged over all gestures below 10% for all gestures.

3.3.2 *Results*

The optimal threshold settings are the ones that minimize both the false positives and false negatives. 4c shows the results for false positives and false negatives for the (a) 80th, (b) 85th, and (c) 90th percentiles. Based on these results, we can first conclude that our initial assessment of attacker strength is correct. Henry is a stronger attacker than Joe, so we focus on Henry and Gino for finding the best threshold. However, our hypothesis is invalidated because there exists no point there all false positives and false negatives are less than 10%. Nonetheless, we next aim to see what is the best



(a) 80th Percentile (b) 85th Percentile (c) 90th Percentile

Figure 4: False positive and false negative results for various thresholds using various thresholds. Although the global minima for each percentile occurs at a lower multiplicative factor, we err on the side of less false negatives such that the intended user (Gino) finds it more usable.

threshold multiplier and percentile setting that achieves the best performance.

Talk about the plots here.

3.4 Number of Calibration Iterations

Thus far, our evaluation is based on our full data set of 30 calibration attempts per gesture. In this section, we investigate the performance of our Gesture Library model for smaller subsets of calibration attempts: 5, 15, and 30. We hypothesize that we will have more false positives and false negatives for the same parameters because of the smaller data set.

DATA HERE.

3.5 Multi-Attempt Authentication

Placeholder

4. SIMULTANEOUS GESTURES EXPERIMENTS

In simultaneous gestures model, we assume the pebble smartwatch as the legitimate prover, an iPhone as the verifier and the other iPhone as the malicious attacker. Three devices starts and ends performing gestures at the same time while the prover and verifier are held together with all axes of accelerometers are correctly corresponded and the attacker is watching and trying to copy the gesture. We tried 4 gestures and each gesture was tested with 10 attempts. The gestures are gesture 5, 7, 11, 17 in **??**. Ideally, the prover should be authenticated because it moves together with the legitimate prover and attacker should be denied.

According to the distance matrix we calculated from the raw data, most distance from the prover to the verifier are less than 100 and only 5 samples have distance larger than 100. This shows that when two devices are held together, it is easy to get low distance result between them and this can be exploited for pairing small devices.

The distance between the attacker and the legitimate prover are all larger than 100 and most of them are more then 200. Compared with the distance result of prover, the distance of attacker is always at least 100 larger than that. It is easy to differ the attacker from the prover in Figure 5 so that our simultaneous gestures model works well.

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.

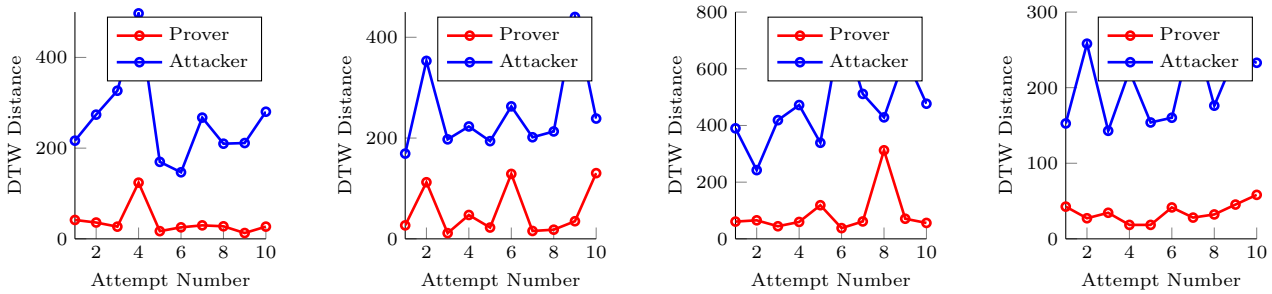
5.1 Gesture & Motion Recognition on a Single Device

Several efficient gesture-recognition algorithms already exist. For accelerometer-based gesture recognition, Ali et al. [2] introduce and evaluate five cutting-edge algorithms, including 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



(a) Gesture #5 (b) Gesture #7 (c) Gesture #11 (d) Gesture #17

Figure 5: DTW Distance from both attacker and prover to verifier, separated by gestures, for the Simultaneous Gestures model.

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. CONCLUSION & FUTURE WORK

Placeholder

7. 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.