RingAuth: User Authentication Using a Smart Ring

Jack Sturgess[a], Simon Birnbach[b], Simon Eberz[c] and Ivan Martinovic[d]
Department of Computer Science, University of Oxford, Oxford, U.K.
fi

Keywords: Authentication, Smart Ring, Smartwatch, Wearable, Mobile Payment, Access Control, Gesture, Biometrics.

Abstract: We show that by using inertial sensor data generated by a smart ring, worn on the finger, the user can be authenticated when making mobile payments or when knocking on a door (for access control purposes). We also demonstrate that smart ring data can authenticate payments made with a smartwatch, and *vice versa*, such that either device can act as an implicit second factor for the other when worn on the same arm. To validate the system, we conducted a user study (n=21) to collect finger and wrist motion data from users as they perform gestures, and we evaluate the system against an active impersonation attacker. We develop payment authentication and access control models for which we achieve equal error rates of 0.04 and 0.02, respectively.

1 INTRODUCTION

Mobile payment systems (also known as tap-and-pay systems), such as Google Pay, have become pervasive. These systems enable the user to provision payment cards to a virtual wallet on a smartphone and then facilitate cashless and contactless payments with NFC-enabled point-of-sale terminals. The functionality of mobile payment systems has been extended to wearable devices, such as smartwatches. When paired with a smartphone, a smartwatch can access and store the same virtual wallet and make payments even when the smartphone is not present.

We are starting to see commercial smart rings enter the market. Sleep tracking rings, such as Thim, use inertial or heart-rate sensors to infer different stages of sleep and monitor sleeping patterns. HelioS uses an ultra-violet radiation sensor to monitor exposure to sunlight to infer vitamin D intake and warn the user about sunburn. Amazon Echo Loop uses a microphone and speaker to enable the user to interact with the Alexa virtual assistant and to make phonecalls through a paired smartphone. Blinq uses inertial sensors for fitness monitoring and provides a discreet panic button in the shape of a fake gemstone that triggers an application on a paired smartphone to send a help request containing geo-location information to contacts or social media, ideal for night joggers.

[a] https://orcid.org/0000-0001-5708-3052
[b] https://orcid.org/0000-0002-2275-7026
[c] https://orcid.org/0000-0003-4101-4321
[d] https://orcid.org/0000-0003-2340-3040

Mastercard K-ring is a smart ring that enables the user to make payments via NFC, similar to a contactless payment card. While the K-ring does not communicate directly with the user's smartphone, a payment application running on the phone can be linked to the same account to which the ring is linked such that payment authorisation requests can be sent from the payment provider to the phone for the user to authorise. As smart ring technology evolves and ring-based services that require user authentication, such as payment and access control systems, grow in feature-richness, there is a growing need for new authentication factors to support them.

Given that wearable devices are often designed with continuous healthcare or fitness use-cases in mind, they tend to have an inertial measurement unit (IMU) consisting of (at least) an accelerometer and a gyroscope. Works in behavioural biometrics have shown that inertial sensor data in smartphones and smartwatches can be used to infer gait or gestures. These systems require some initial calibration (*i.e.*, a cumbersome enrolment phase), but then continuously authenticate the user with reasonable effect.

Contributions. In this work, we show that inertial sensor data generated by a smart ring can be used to authenticate the user when making certain gestures, such as tapping a payment terminal or knocking on a door. Specifically, we contribute the following:

- We propose a novel, smart ring-based authentication system and we conduct a user study (n=21) to evaluate it. Using only inertial sensor data, we show that a single tap gesture performed by

Sturgess, J., Birnbach, S., Eberz, S., Martinovic and I.
RingAuth: User Authentication Using a Smart Ring.
DOI: 10.5220/0013494900003979
In *Proceedings of the 22nd International Conference on Security and Cryptography (SECRYPT 2025)*, pages 96-107
ISBN: 978-989-758-760-3; ISSN: 2184-7711

a user while making a payment with a smart ring can implicitly authenticate the user. We also show that smart ring data can authenticate a smartwatch user, and *vice versa*, allowing either device to act as an implicit second factor for the other.

- We show that our approach can also be applied to an access control context, in which a single knock gesture on a door (measured by sensors on either device or embedded in the door itself) can explicitly authenticate the user.
- We show that our authentication models are resistant against an active impersonation attacker.
- We make our data available (Sturgess, 2023).

2 SYSTEM DESIGN

2.1 Design Goals

In this paper, we investigate the use of inertial sensors on a smart ring for user authentication purposes. We consider both the use of the smart ring to authenticate the user for its own services and to support authentication on other devices and the use of other devices to support authentication for services on the smart ring.

First, we select the context of mobile payments and we consider the tap gesture made when the user taps an NFC-enabled device against a point-of-sale terminal to provide a gesture biometric and we pose the following questions:

- Can tap gestures made with a smart ring, as measured by the inertial sensors of the smart ring, be used to implicitly authenticate the user during a transaction?
- Can tap gestures made with a smart ring, as measured by the inertial sensors on a *smartwatch* worn on the same arm, be used to implicitly authenticate the user (in case the smart ring does not have inertial sensors of its own) during a transaction?
- Can implicit authentication in a smartwatch system be improved by incorporating (the inertial sensor data of) a smart ring as a second factor?

Second, we select the context of access control and we consider knock gestures on a closed door, as measured by inertial sensors on worn devices and mounted on the door. We pose the question:

- Can knock gestures, as measured by inertial sensors, be used to authenticate the user implicitly (with regular knocks) or explicitly (with user-chosen secret knocks)?

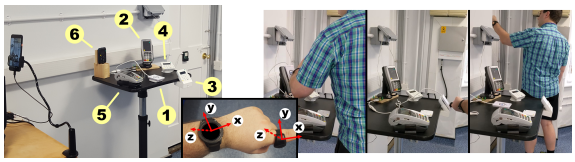


Figure 1: The equipment used in our experiments: six fixed terminals, an NFC reader (here on Terminal 1), a Raspberry Pi for timestamp collection, a Raspberry Pi attached to the door, and a fixed camera. Inset: our smart ring and smartwatch worn on the left arm with sensor axes shown (the z-axes point upwards through the screens). Right: the attacker's view of a victim interacting with (from left) Terminals 2 with the watch, 3 with the ring, and the door.

Table 1: Details of the terminals used in our payment experiment; the indices correspond to the labels in Figure 1 and 'F' is the freestyle terminal. *Height* is measured from the floor to the lowest point of the terminal; *Tilt* is the inclination at the lowest point of the terminal; and *Distance* is measured from the front of the stand to the foremost point of the terminal.

Terminal	Height (cm)	Tilt (°)	Distance (cm)
1	100	0	5
2	120	60	25
3	95	45	-10
4	105	30	15
5	110	15	10
6	115	90	30
F	picked up from centre of stand		

2.2 System Model

For our payment model, we consider a system model in which the user is wearing both a smart ring and a smartwatch on the same arm and is using them to make an NFC-enabled payment at a point-of-sale terminal in a typical setting (*e.g.*, a shop). We assume that a payment consists of the user making a *tap gesture* by moving a device towards the terminal until it is near enough to exchange data via NFC. We assume that each tap gesture ends when the NFC contact point is found, as this is when the payment provider would decide whether to approve the payment.

For our access control model, we consider a system model in which the user is wearing both a smart ring and a smartwatch on the same arm and is knocking on a door with that hand as a means to authenticate to an access control system. This is a *knock gesture*.

We assume that the devices have an accelerometer and gyroscope and that we have access to their data. We use data from the inertial sensors only. We assume that the user's biometric templates are stored securely on the wearable devices. When we combine data from multiple devices into a single sample for classification, as we do in some of our models, we assume that

one device shares its feature vector components wirelessly with the other in a secure manner. We do not make any assumption on how the wider system might act on the authentication decision; however, an example would be to have the NFC module of the tapping device disabled by default and only enabling it if the user is locally authenticated by the gesture biometric.

2.3 Threat Model

We consider an adversary that has possession of a legitimate user's smart ring (or smartwatch, or both, as appropriate), has unlocked it, is wearing it, and is attempting to use it to make a payment at a terminal or to gain entry to a locked door via an access control system. The adversary may have (maliciously) stolen the device(s) or (benignly) borrowed it. Our goal is to authenticate the legitimate user and to reject the adversary by using gesture biometrics. We consider the following two types of attack:

- *Zero-effort attack*: for any given victim, all other users are considered to be passive, zero-effort attackers.
- *Observation attack*: an active attacker who sees (and hears) the victim perform gestures (*e.g.*, via a hidden camera) and then attempts to mimic him.

We focus on how gesture biometrics can be used to defend against these attacks. We do not consider threats to other components in the system, tampering of devices or biometric templates, malware, or denial of service attacks.

3 EXPERIMENT DESIGN

3.1 Experiment Overview

To evaluate the extent to which finger and wrist motion data can be used to authenticate users, and to compare the two, we designed and conducted a user study to collect data. We set up six point-of-sale terminals on an adjustable stand fixed at a height of 100 cm, an ACR122U NFC reader connected to a Raspberry Pi for timestamp collection, a second Raspberry Pi with an accelerometer and a gyroscope attached to a closed door, and a smartphone fixed in position for video recording. For our wearable devices, we used a smart ring and a smartwatch (detailed below), both commercial off-the-shelf devices and worn together on the same arm by the user. We collected motion data from the wearable devices and the door-mounted Raspberry Pi (with all clocks synchronised) as the user performed gestures. Figure 1 shows our apparatus.

For our payment experiment, we affixed an NFC tag to the front of each wearable device and we affixed the NFC reader to the front of each point-of-sale terminal in turn. For each terminal, the user stood in front of the stand and performed tap gestures on a terminal using a wearable device, as if making real payments, with a short spacing delay between each tap gesture. The use of NFC tags and the NFC reader ensured consistent NFC communication between the wearable devices and the terminals. Each NFC tag stored the name of the wearable device to which it was affixed; each time NFC contact was made during a tap gesture, the Raspberry Pi captured its timestamp and the name of the triggering device. This was later used to segment the inertial sensor data collected from the devices to retrieve the data for each tap gesture.

For our access control experiment, the user stood in front of the closed door and performed knock gestures on it using his device-wearing hand. The user pressed the button on the smart ring before and after each knock gesture to capture bounding timestamps. These timestamps were later used to segment the inertial sensor data collected from all devices to retrieve the data for each knock gesture.

3.2 Point-of-Sale Terminals

To immerse the user in a real-world payment setting as much as possible, we captured tap gestures using *seven* point-of-sale terminals. Six terminals were fixed in position (as detailed in Table 1) and one was movable—we call this the 'freestyle' terminal. To inform the placement of the fixed terminals, we surveyed supermarkets and restaurants to find popular or standardised terminal positions (in terms of height, tilt angle, and distance from the user). We set one of the fixed terminals (Terminal 3) to match the position of the terminal on a train station barrier in the UK, which is widely standardised and represents another setting where mobile payments are made. For the freestyle terminal, the user picked up the NFC reader from the centre of the stand with his other hand and performed a tap gesture on it, returning it after each gesture, as if a shopkeeper had handed a terminal to a customer.

3.3 Wearable Devices and Sensor Modules

For our smart ring, we used a Genki Wave. This ring is worn on the index finger and has a button that can be pressed with the thumb—we utilised this for timestamp collection in our access control experiment. We

wrote a data collection script in Python using the open source library provided by the developers to interface with the ring over Bluetooth LE.

For our smartwatch, we used a Samsung Galaxy Watch running the Tizen 4.0 operating system. We built a data collection app and installed it on the smartwatch using the Tizen Studio IDE.

From each of these wearable devices, we collected timestamped data from four inertial sensors directly or derived from their MEMS sensors. The *accelerometer* measures change in velocity. The *gyroscope* measures angular velocity. The *linear accelerometer* is derived from the accelerometer and the magnetometer to exclude the effects of gravity. The *gyroscope rotation vector* (GRV) is a fusion of sensor readings to compute the orientation of the device. We collected this data with sampling rates of 100 Hz for the smart ring (which we downsampled to 50 Hz), 50 Hz for the smartwatch, and 30 Hz for the door-mounted Raspberry Pi.

The inertial sensor axes are fixed relative to the frame of each device (as shown in Figure 1). Motion measured along the x-axis corresponds with pushing or withdrawing the arm; the y-axis, with waving the arm from side to side; and the z-axis, with movements up- and downwards through the screen.

3.4 Tap and Knock Gestures

To collect tap gestures for our payment experiment, we had each user perform the following types of tap gesture on each of the seven point-of-sale terminals:

- *Ring Tap*: the user tapped the smart ring against the terminal and moved it around, if necessary, until NFC contact was made to simulate a *ring*-based payment.
- *Watch Tap*: the user tapped the smartwatch against the terminal and moved it around, if necessary, until NFC contact was made to simulate a *watch*-based payment.

To collect knock gestures for our access control experiment, we had each user perform the following types of knock gesture on the closed door:

- *3-knock*: the user knocked on the door three times.
- *5-knock*: the user knocked on the door five times.
- *Secret knock*: the user created a knock pattern consisting of between three and six knocks.

3.5 User Study

To collect our data, we conducted a user study that was approved by the relevant research ethics committee at our university. We recruited 21 participants that included staff, students, and members of the public. Each participant attended three data collection sessions and performed 10 of each of the gestures detailed in Section 3.4 in each session. The first two sessions were separated by a break of 5 minutes and the final session occurred on a different day. We collected $17 \times 10 \times 3 = 510$ gestures from each user.

In each experiment, the participant was asked to stand facing the terminals or the door; aside from this, we did not prescribe any constraints on positioning as we wanted the user to interact comfortably as though acting in a real-world setting. The first three gestures of any type were performed in silence to familiarise, then the researcher engaged the participant in light conversation to simulate the distractions of a real-world environment. This sometimes elicited additional hand and body movements if the user gesticulated naturally.

Impersonation. To evaluate the robustness of our approach against an observing attacker, all 21 participants consented to having their gestures recorded and participated in an impersonation exercise. The first 3 were recorded as victims and the latter 18 impersonated them; then, at the end of the study, the first 3 returned to impersonate the other 18. This design enabled us to compare the susceptibility of different victims and the skill of different attackers (also known as *wolf and lamb analysis*). We recorded the following six gestures of each participant: smart ring and smartwatch tap gestures on Terminals 2 and 3, 5-knock gestures, and secret knock gestures. The camera was fixed in position, as if hidden, and so the amount of observable information was controlled; Figure 1 shows the attacker's view of the terminals and the different information observable for each type of gesture. For each attack, the attacking participant watched (and heard) a short video of the victim performing the gesture three times and then made three attempts to mimic him. The attacker wore the wearable devices on the same arm as the victim.

User Statistics. Of our 21 participants: 15 were male, 17 wore the devices on the left arm (the decision was led by the smartwatch; everyone who wore them on the right arm was female), 15 regularly wore a watch of some kind (7 of which wore a smartwatch), 13 had paid with a smartphone before, and 5 had paid with a smartwatch (*i.e.*, 71% of those who regularly wore one). 16 participants (76%) remembered their secret knock, with an average of 4.8 days between their sessions (those who did not remember had an average of 4.2 days between sessions).

4 METHODS

4.1 Data Processing

We collect time-series data from the inertial sensors on the wearable devices and those attached to the door. Each accelerometer, gyroscope, and linear accelerometer sample at time t is given in the form (t,x,y,z). Each GRV sample at time t is given as a quaternion in the form (t,x,y,z,w).

We express a single tap or knock gesture as a series of samples within a time window. In our payment experiment, we retrieve the tap gestures for each user by segmenting sensor data into 4-second windows using NFC contact point timestamps as endpoints. To find the optimum parameters for a tap gesture window, we compare (in Section 5) the performance of gestures bounded by a variety of window sizes and time offsets. We define the offset to be the time between the NFC contact point and the end of the window—*i.e.*, for an NFC contact point timestamp T_0, a window size s, and an offset o, we would retrieve a tap gesture with start time T_S and end time T_E, where $T_E = T_0 - o$ and $T_S = T_E - s$. In our access control experiment, we retrieve the knock gestures for each user using bounding timestamps captured when the user pressed the button on the ring before and after each gesture.

4.2 Feature Extraction

Whenever a gesture is retrieved, we reduce noise in the data by applying a Butterworth low-pass filter and then process the gesture in the following 19 dimensions. For each accelerometer, gyroscope, and linear accelerometer sample, we process the filtered x-, y-, and z-values, the Euclidean norm of the filtered values, and the Euclidean norm of the raw (unfiltered) values. For each GRV sample, we process only its four filtered values.

We reduce each gesture to a feature vector containing the following 220 members. For each gesture, we extract ten statistical features in each dimension—namely: *minimum*, *maximum*, *mean*, *median*, *standard deviation*, *variance*, *inter-quartile range*, *kurtosis*, *skewness*, and *peak count*. And, from each of the gesture's accelerometer, gyroscope, and linear accelerometer vectors, we calculate the *mean velocity*, *maximum velocity*, and *displacement* along each axis and the *Euclidean displacement*.

In previous works (Sturgess et al., 2022a; Sturgess et al., 2022b), we found similar feature sets to be ideal by starting with a larger set and pruning it down using normalised Gini importances to reject the least informative features, so we use the same approach here on similar grounds. This approach also yielded promising preliminary results in our access control model, so we used it for both models to allow for comparability and consistency throughout the paper.

In some of our models, we combine inertial sensor data from multiple devices. In these cases, the above features are extracted for each separate source and then concatenated together to form a linearly larger feature vector.

4.3 Classification

We construct random forest classifiers for each of our payment authentication and access control models in Python (Pedregosa et al., 2011) and we include 100 decision trees in each forest. Similar works (Acar et al., 2020; Sturgess et al., 2022b) have shown that random forests are efficient, robust against noise, and capable of estimating the importance of features. To mitigate the effects of randomness, and to ensure that our results are fair and unselective, we reconstruct each classifier ten times with different forest randomisation seeds and average the results.

Position-Agnostic Model. To evaluate the zero-effort attacker, we train a set of classifiers that are user-dependent and agnostic to the position of the terminal. This means that a separate template (and decision threshold) is generated for each user and that, for each tap gesture under test, the tap gesture samples used to train the classifier came from tap gestures made only against other terminals (*i.e.*, a leave-one-out approach). The user's tap gestures form the positive class and all other users' tap gestures form the negative class. As this is an authentication scenario, we ensure that the training data temporally precedes the testing data by taking the tap gestures collected in users' first data collection session as training data (analogous to the enrolment phase, where the user template is created) and those collected in the second session as testing data (analogous to the authentication phase).

Position-Known Model. To evaluate the observation attacker, we apply a similar design, except that we do not exclude tap gestures based on terminal. We assume that, since the attacker has already observed the victim using the terminal in question, the system has knowledge of the victim's tap gestures made against that terminal. We pair up every user as a victim with every other user as an attacker, one at a time, and train the classifiers, excluding all of the attacker's tap gestures. This enables us to generate the victim's decision threshold for that pairing, tuned to the EER (which we take as our baseline FAR, the *base-*

FAR), with no knowledge of the attacker, and then we test the attacker's impersonation samples against that tuned classifier to find his attack success rate (the *observation-FAR*). We compare these two FARs to measure the success of the observation attack.

Position-Specific Model. We train a set of classifiers in which each is trained and tested on tap gestures made on a single terminal. This model enables us to measure the effectiveness of our approach if implemented in systems with standardised fixed terminals, such as public transport systems.

Access Control Model. For knock gestures, to evaluate each of the attackers, we use a similar approach as with each respective payment authentication model above, except that we do not need to generalise the model over multiple terminals. We train separate models for each of the three types of knock gesture.

4.4 Performance Metrics

We define the *true positives* to be the number of times that the positive class (*i.e.*, the legitimate user) is correctly accepted; the *true negatives* to be the number of times that the negative class (*i.e.*, the adversary) is correctly rejected; the *false positives* to be the number of times that the negative class is wrongly accepted; and the *false negatives* to be the number of times that the positive class is wrongly rejected. The decision threshold, θ, is the score required for the classifier to assign to a sample the positive class. To tune, we adjust θ—if we increase θ, the model becomes more resilient to false positives, so a greater θ will favour security and a smaller θ will favour usability.

To quantify the performance of our models, we find the optimum θ where the false acceptance rate (FAR) equals the false rejection rate (FRR); this cross-over point is called the equal error rate (EER). The FAR gives us an indication of *security*, as it measures the likelihood that the negative class will be wrongly accepted. The FRR gives us an indication of *usability*, as it measures the likelihood that the positive class will be wrongly rejected. The EER is commonly used as a metric in authentication systems as it gives a balanced measure of system performance. To understand how well a classifier can distinguish between positive and negative classes, we plot a receiver operating characteristic (ROC) curve by comparing its true positive rate against its FAR for each θ. The area under this curve (AUC) gives us an indication of the classifier's ability to separate the classes, where a score of 1 shows perfect class separation, a score of 0.5 shows none, and a score of 0 shows that the classifier is always assigning the wrong class.

5 RESULTS

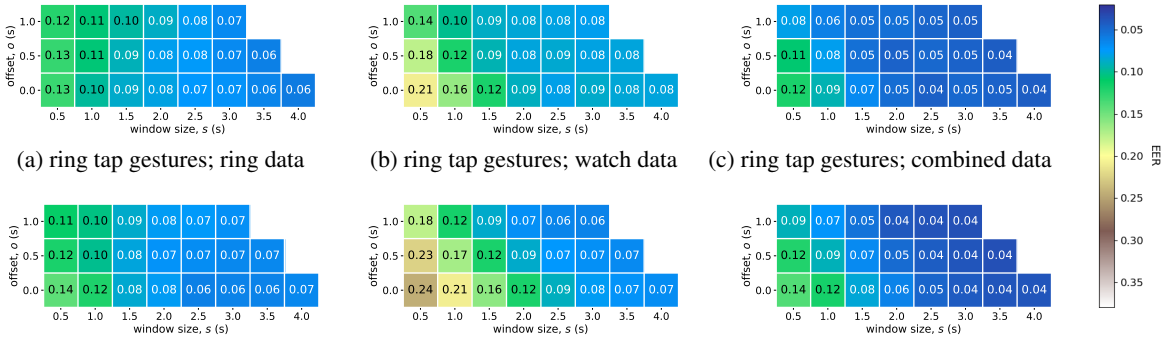
5.1 Zero-Effort Attack

Tap Gestures. Figure 2 shows the average EERs for our *position-agnostic* payment authentication models by window size and offset. With 21 users and 6 fixed terminals, each score is the average of scores from $21 \times 6 \times 10 = 1,260$ classifiers (see Section 4.3 for details).

For ring tap gestures, Figure 2a shows that when using data from the smart ring only our model achieves EERs as low as 0.06. Figure 2b shows that when using data from the smartwatch only, we get 0.08. This suggests that a smartwatch could perform as a reasonable authenticator for a smart ring, in a scenario where the smart ring does not have any inertial sensors of its own (such as the Mastercard K-ring). If the data from both devices are combined, Figure 2c shows that we achieve EERs as low as 0.04. The optimum parameters for ring tap gestures, by considering EERs and favouring a smaller window size for usability, are $\{s = 2.5, o = 0\}$. Figure 3a shows the ROC curves for these models by gesture and device in this optimum window; we achieve AUCs of up to 0.96.

When we use data from the smartwatch only, we see a pattern radiating from the bottom left corner of the grid. This shows that, when training only on data where the tapping device is near to the terminal (small window, just before the NFC contact point is found), the movements of the wrist are not discriminative between users. The same is not true when using data from the smart ring only; this suggests that the finger remains active during that time, even for watch tap gestures. Figure 2f shows that including a smart ring as an additional factor can improve the authentication of a user making watch-based payments (*cf.* Figure 2e).

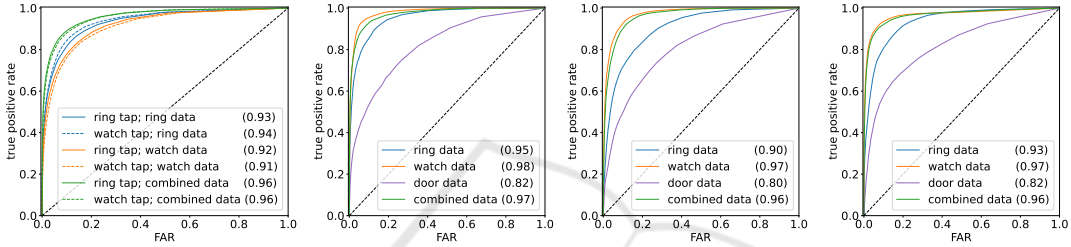
Knock Gestures. Across all of the participants in our user study, the average 3-knock gesture lasted 2.81 seconds, the average 5-knock, 3.32 seconds, and the average secret knock, 3.78 seconds. Table 2 shows the average EERs for our access control models. Each score is the average of scores from $21 \times 10 = 210$ classifiers. Figure 3 shows the average ROC curves. Unexpectedly, our 3-knock model performed best, achieving an EER of 0.02 and an AUC of 0.98 using data from the smartwatch only. The weaker results of the 5-knock model might be due to some participants sometimes miscounting the number of knocks when performing a 5-knock gesture, suggesting that it is a less user-friendly gesture and leading to messier data, whereas 3-knock gestures were performed effortlessly and more consistently.



(a) ring tap gestures; ring data (b) ring tap gestures; watch data (c) ring tap gestures; combined data

(d) watch tap gestures; ring data (e) watch tap gestures; watch data (f) watch tap gestures; combined data

Figure 2: Average EERs for our payment models by window size and offset, for tap gestures made with the smart ring (top) and smartwatch (bottom), using data from the smart ring only (left), the smartwatch only (centre), and both combined (right).



(a) payment models (b) 3-knock gestures (c) 5-knock gestures (d) secret knock gestures

Figure 3: Average ROC curves for our payment models in optimum window $\{s = 2.5, o = 0\}$ and our access control models, using data from (i) only the smart ring, (ii) only the smartwatch, (iii) only the door-mounted sensors, and (iv) both or all three (as relevant) combined. The AUCs are given in parentheses.

Table 2: Average EERs for our access control models, using data from (i) only the smart ring, (ii) only the watch, (iii) only the door-mounted sensors, and (iv) all three combined.

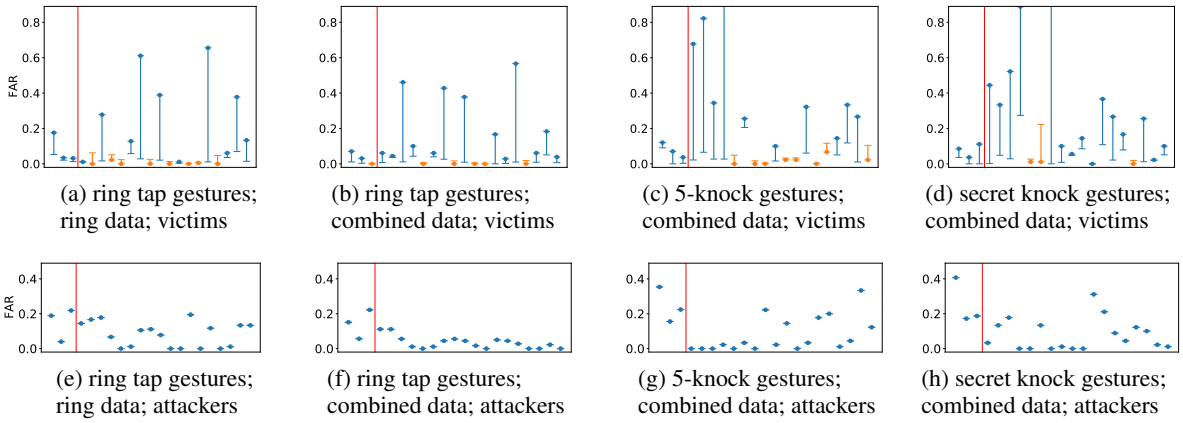
Knock Gesture	EER			
	Ring	Watch	Door	Combined
3-knock	0.06	**0.02**	0.17	0.03
5-knock	0.12	**0.04**	0.21	0.05
secret knock	0.09	**0.05**	0.19	**0.05**

When we use data from the smartwatch only, we achieve the best error rates and class separation across all models, including those based on combined data. This suggests that wrist movements are a key discriminator in knocking, to such an extent that other factors act as pollutants. When we use data from the door only, we achieve the poorest results; the lower sampling rate of the door-mounted sensors may have an impact, but the magnitude of the difference in results is likely explained by those sensors lacking knowledge of user movements. Despite this sparse data, the classifiers are still able to achieve a reasonable degree of class separation, as evidenced by their average AUC of 0.81.

5.2 Observation Attack

Figure 4 shows the results of our observation attack against our *position-known* payment authentication model and our access control model. Each of the first 3 users was the victim of 1,080 ring tap impersonation attempts (18 attackers × ring tap gestures on 2 terminals × 3 attempts at each gesture × 10), 540 5-knock attempts, and 540 secret knock attempts; each of the other 18 users, separated in the figures by a red line, was the victim of 180 (3 attackers), 90, and 90 attempts, respectively.

For ring tap gestures, the base model achieves average base-FARs of 0.03 when using data from the smart ring only and of 0.01 when using data from the smart ring and smartwatch combined; when attacked, the average success rates (observation-FARs) are 0.06 and 0.05, respectively (and similar results for watch tap gestures). Figures 4a and 4b show that a small number of our users are *lambs* (*i.e.*, users who are especially susceptible to impersonation), where their observation-FAR is significantly larger than their base-FAR. Figure 4b shows that the addition of the smartwatch data helps to reduce the largest FAR deltas, and the overall average observation-FAR, but also opens a new vector that increases the suscepti-



(a) ring tap gestures; ring data; victims
(b) ring tap gestures; combined data; victims
(c) 5-knock gestures; combined data; victims
(d) secret knock gestures; combined data; victims
(e) ring tap gestures; ring data; attackers
(f) ring tap gestures; combined data; attackers
(g) 5-knock gestures; combined data; attackers
(h) secret knock gestures; combined data; attackers

Figure 4: Results of our observation attack against our *position-known* payment models in optimum window $\{s = 2.5, o = 0\}$ and our access control models. The top row shows for each user as a victim the average FAR of the user-specific base model (flat line) and the average FAR when attacked (circle); if the latter is greater, then the victim's line is coloured blue, otherwise it is orange. The bottom row shows for each user as an attacker the average FAR achieved when attempting to impersonate other users. The red line separates the first 3 users from the other 18, indicating the two groups of users (see Section 3.5).

bility of some users. Figures 4e and 4f show that none of our users are *wolves* (*i.e.*, users who are especially skilled at impersonation); when using data from the smart ring only, some attempts got lucky against a random spattering of users, but when the smartwatch data were combined, the success rate dropped. This suggests that our system provides resistance against wolf behaviour, reducing the likelihood that an attacker could predictably impersonate a given victim—and so, in the wider system, this may act as a deterrent.

For knock gestures, when using data from the door-mounted sensors, smart ring, and smartwatch combined, we have average base-FARs of 0.05 for both the 5-knock and secret knock gestures and average observation-FARs of 0.08 and 0.09, respectively. Figures 4c and 4d show that we have a number of lambs, this time with greater FAR deltas. Knock gestures are notably weak against impersonation if they are loud and slow. For the secret knock, the fourth and sixth users after the red line have high base-FARs because those users chose common gesture fragments in their secret knocks. For the former (and the seventh user), the gesture was loud and slow, as evidenced by the large observation-FAR. For the latter, curiously, the observation-FAR is far lower than the base-FAR, suggesting that the gesture was difficult to mimic intentionally despite having commonality with other gestures. This gesture contained three fast knocks in the middle, which captured the attention of attackers only for them not to match the surrounding knocks. Figures 4g and 4h show that attentive attackers are able to achieve reasonable attack success rates, but this is due to the lambs being more vulnerable—whether by poor choice of secret or poor execution (*i.e.*, loud and slow)—rather than the attacker being especially skilful. The knock gesture has a greater dependence on user discretion than the tap gesture.

5.3 Feature Informativeness

To investigate which features are most informative to our models, we sum the top five features, sorted by Gini importance, of each classifier. (Note that, *w.r.t.* the counts, there are six times more classifiers for the payment models.)

For ring tap gestures, Table 3a shows that our models favour GRV-derived features when using data from the smart ring only, but accelerometer- and gyroscope-derived features when using data from the smartwatch only, indicating that the finger moves to a position faster and remains in a position longer than the wrist, whose movements are smoother. The combined model, which achieved stronger results than the others as we saw in Figure 2, had twice as many members in its feature vector and ended up favouring a similar set, re-affirming the relative importance of these features.

For 3-knock gestures, Table 3b shows the dominance of features derived from the y- and z-axes of the wearable devices, which is to be expected as these measure the sideways and forward movements of the hand, respectively. Aside from these, two notable exceptions show greater importance: the x-axis of the smartwatch yields a single important feature, representing the maximum acceleration of the arm as it extends towards the door initially, and the median impact experienced by the door-mounted accelerometer, indicating that each user struck the door with consistent force.

Table 3: Modal top-five features by Gini importance summed over classifiers for our payment models in optimum window $\{s = 2.5, o = 0\}$ and our access control models, using data from (i) only the smart ring, (ii) only the smartwatch, (iii) only the door-mounted sensors, and (iv) both or all three (as relevant) combined. Features are given in the format sensor-axis-statistic; for combined models, the leading character indicates the device to which the sensor belongs (ring, watch, or door).

(a) payment models; ring tap gestures

Ring		Watch		Combined	
Feature	**#**	**Feature**	**#**	**Feature**	**#**
GRV-y-med	309	Acc-x-min	297	w-Acc-x-min	185
GRV-y-mean	269	Acc-y-max	208	w-Acc-y-max	151
GRV-x-mean	239	Acc-x-max	208	r-Gyr-z-mean	145
GRV-x-med	223	Acc-z-max	189	r-Acc-x-vmax	141
GRV-x-max	222	Gyr-z-vmean	138	r-GRV-x-max	135
GRV-y-max	217	Gyr-z-mean	126	r-Acc-x-mean	128
Gyr-z-mean	192	Gyr-z-min	124	r-GRV-y-mean	122
Acc-x-mean	162	Gyr-z-disp	119	r-Acc-x-med	122
GRV-z-max	157	Acc-x-mean	117	r-GRV-x-med	116
Acc-x-vmax	155	Acc-x-vmax	116	r-GRV-y-med	116

(b) access control models; 3-knock gestures

Ring		Combined	
Feature	**#**	**Feature**	**#**
GRV-w-med	24	d-Acc-y-med	22
Acc-y-disp	22	w-Acc-x-max	20
GRV-y-max	20	w-Gyr-z-var	17
GRV-w-mean	19	w-Gyr-y-med	17
GRV-z-med	19	w-Gyr-z-vmax	15
Acc-z-mean	18	w-Gyr-z-stdev	15
Acc-z-med	18	w-LAc-z-disp	15
Gyr-y-vmax	18	w-GRV-y-iqr	15
GRV-x-mean	18	w-GRV-y-med	14
LAc-x-disp	17	w-Gyr-y-min	12

5.4 Sensor Selection

We collected motion data from all of the inertial sensors available on our devices. Some devices are more limited in their offering—the accelerometer is the commonest sensor, as it is the smallest and cheapest. To assess the feasibility of our approach on devices with fewer sensors, we trained a set of sensor-specific models in which each classifier is trained and tested on data from a subset of sensors.

For models that use data from the smart ring only, we find distinctly poorer results whenever we remove the GRV sensor. With just an accelerometer, we see poorer results with an increase of approximately 0.04 in every EER. Conversely, for models that use data from the smartwatch only, we find that we achieve EERs of 0.05 in windows that achieved EERs of 0.08 in Figure 2c when using only the accelerometer and gyroscope; this suggests that the other sensors pollute the smartwatch classifiers, echoing similar findings in related work (Sturgess et al., 2022b). The combined models are broadly unchanged, relying on ring features when GRV is included and watch features when not.

5.5 Terminal Positions

We collected tap gestures performed against a range of terminals. Some systems, such as public transport systems, have highly standardised terminals (*i.e.*, dedicated terminals that can be found set at the same position in many instances). To compare the effectiveness of our approach in a general setting against a standardised setting, we trained a set of position-specific models in which each classifier is trained and tested on data from a single terminal.

Table 4 shows the average EERs for our *position-specific* payment authentication models when trained and tested on tap gestures from a single terminal. In general, we gain little improvement from restricting our system to a single terminal. We find that user comfort has a beneficial impact on authentication results. Terminals 5 and 6 show slightly improved results and these were the most comfortably positioned terminals for the majority of participants, who wore the devices on their left arm (indeed, if we reconstruct our models using data only from those users wearing the devices on their left arm, the relative gains are greater still). Likewise, the freestyle terminal shows improved results for watch tap gestures, since the watch was more awkward to tap than the ring and this terminal accommodated smoother movements—however, it had the opposite effect for ring tap gestures, likely because most users tilted and moved the terminal a shorter distance when interacting with it when using the ring than when using the watch, eliciting a sloppier punch gesture.

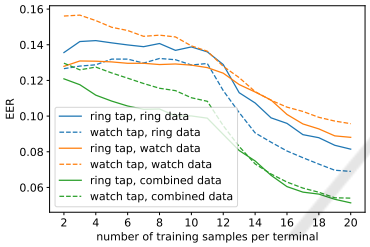
5.6 Enrolment Parameters

The enrolment phase of a biometric system requires the user to submit samples of the measured characteristic so that the initial template can be constructed. This can be particularly user-unfriendly in behavioural biometric systems due to the effort required to collect these samples. To evaluate how much we can expedite this process, we trained a set of models in which each classifier is trained on fewer user samples (*i.e.*, a smaller positive class).

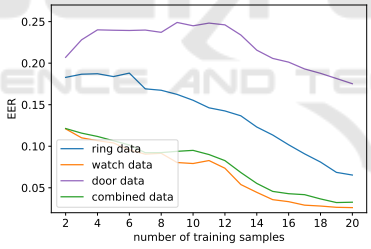
Figure 5a shows that our payment models can authenticate the user with EERs as low as 0.12 when trained on just twelve of the user's tap gestures

Table 4: Average EERs for our *position-specific* payment models in optimum window $\{s = 2.5, o = 0\}$, using data from the smart ring only, the smartwatch only, and both combined. Our *position-agnostic* results are included for comparison.

	(a) ring tap gestures		(b) watch tap gestures	
	EER		**EER**	
Terminal	**Ring**	**Combined**	**Watch**	**Combined**
1	0.07	0.04	0.08	0.05
2	0.09	0.07	0.09	0.07
3	0.09	0.07	0.09	0.07
4	0.08	0.03	0.06	0.05
5	**0.06**	**0.02**	**0.06**	**0.03**
6	**0.07**	**0.03**	**0.04**	**0.02**
F	0.10	0.09	**0.05**	**0.04**
agnostic	0.07	0.04	0.09	0.05



(a) payment models



(b) access control models; 3-knock gestures

Figure 5: Average EERs for our payment models in optimum window $\{s = 2.5, o = 0\}$ and our access control models if trained on fewer enrolment samples. For the payment model, each classifier is trained on six terminals.

(spread evenly over six terminals), which can be performed in less than a minute. Figure 5b shows that, when including watch data, our 3-knock access control model can authenticate the user with EERs as low as 0.12 when trained on just two 3-knock gestures (the access control models for the other knock gestures show a similar pattern), which can be performed in a few seconds. In both cases, we see that the error rates improve as more samples are included in the training set; this suggests that we can relax upfront sample collection requirements and then incorporate subsequent gestures with an update mechanism to improve the models over time as the system is used.

6 DISCUSSION

Power Consumption. Wearable devices are designed to facilitate always-on sensing (*e.g.*, for health monitoring). To measure the impact of our data collection in practical terms, we wore two of each wearable device in an identical state, but only collecting data from one of each. For the smart rings, there was no noticable difference in power consumption over 6 hours. For the smartwatches, our app consumed an additional 1.5% of battery capacity per hour.

Response Time. We calculated the computation time for classifying a single watch tap gesture, averaged over 10,000, to be 7.11 ms on a desktop computer with an Intel Core i5-6500 processor. Using a benchmarking tool, we found that a Samsung Exynos W920 (a modern smartwatch processor) performs 26 times slower, so we would expect an authentication decision to be made in roughly 185 ms on a smartwatch. Robust benchmarking is not yet available for smart ring CPUs.

3-knock Impersonation. Table 2 shows that we achieve our best overall results from the 3-knock gesture. In our preliminary testing, this was not the case, so we chose not to include it in the impersonation exercise of our user study due to time restrictions. This is regrettable, as 3-knock gestures in the observation attack may have yielded more interesting results than 5-knock gestures.

Secret Knock Feasibility. The three drawbacks of using an explicit gesture biometric are that it (i) takes time to perform, (ii) must be remembered, and (iii) must be performed discreetly. In our user study, 76% of participants remembered their secret knock, with an average of 4.8 days between their sessions, which suggests that a secret knock is memorable. The average length of a secret knock gesture was 3.78 seconds, although we influenced this by restricting its size to three to six knocks. This restriction limited the key space and resulted in each secret knock gesture consisting of 1-, 2-, or 3-knock fragments knocked at a certain cadence—despite this, we had only one collision, where two users chose the same. We found that these two users were not good at impersonating each other because once they heard the familiar gesture they performed their own version rather than listening to the subtle differences. In a larger user set, we would expect to see more collisions; however, this finding suggests that it would not necessarily weaken the system. The chief weakness of a secret knock is that its sound is difficult to conceal; however, as we show in Figure 4d, only a few of the users in our user study were highly susceptible to having their secret knocks impersonated.

Smart Door Feasibility. Figure 3 shows that we are reasonably able to distinguish between users even when we use data only from the door-mounted sensors. This is a promising result that presents the possibility of a smart door system that can authenticate (and perhaps identify, which raises privacy concerns) the user using only its own embedded inertial sensors, without requiring the user to hold or wear any devices. We leave this project to future work.

7 RELATED WORK

Tap Gestures. The use of inertial sensor data to authenticate tap gestures in tap-and-pay systems was proposed by Shrestha *et al.* (Shrestha et al., 2016) for smartphone-based systems, achieving F-measure scores of 0.93, and by Sturgess *et al.* (Sturgess et al., 2022b) for smartwatch-based systems, achieving F-measure scores of 0.88 and EERs of 0.07. The use of a smartwatch, due to the physiology of the arm, introduced the challenge that the sensor axes frequently change reference frames because the device changes orientation during the tap gesture. We found that the use of a smart ring sits between the two in terms of complexity: the smartphone requires no major change in orientation and is the trivial case, the smart ring requires only a single change because the finger is easily manoeuvred towards the terminal, and the smartwatch orientation is changed frequently during the tap gesture. We found similar results to related works in our smartwatch models and improved results with our smart ring and combined models.

Smartwatches. The use of inertial sensors on smartwatches have been used in a variety of authentication cases. For implicit authentication, Johnston *et al.* (Johnston and Weiss, 2015) showed that wrist motion data can be used to authenticate the user while walking with 10-second windows of data. For explicit authentication, some works have shown that various full arm (Yang et al., 2015), punch (Liang et al., 2017), and hand (Yu et al., 2020) gestures can be used to authenticate the user in a similar fashion. Nassi *et al.* (Nassi et al., 2016) showed that wrist motion data can be used to authenticate handwritten signatures and other authors (Ciuffo and Weiss, 2017; Griswold-Steiner et al., 2017; Griswold-Steiner et al., 2019; Wijewickrama et al., 2021) applied a similar approach to freestyle handwriting. We found optimum results with 2.5 seconds of data for tap gestures and 2.8 seconds for knock gestures.

A number of works have used inertial sensor data from a smartwatch to verify the user's continued interaction with another device to facilitate automatic de-authentication on that other device (as an improvement to timed lockouts). Mare *et al.* (Mare et al., 2014) showed that wrist motion data can be used to infer a sequence of interactions from the user that can be correlated with inputs on his workstation, such that he can be de-authenticated from the workstation if the correlation stops (however, the system was found to be vulnerable to spoofing attack as the attacker can control both streams (Huhta et al., 2016)). Acar *et al.* (Acar et al., 2020) showed that wrist motion data can be correlated with keystrokes in a similar manner. Other works (Lee and Lee, 2016; Mare et al., 2019) have correlated wrist motion data with interactions on a smartphone.

Smart Rings. Few authors have considered the use of smart rings in authentication use-cases. Sen *et al.* (Sen and Kotz, 2020) proposed the use of a smart ring that is capable of producing a vibration to bootstrap a communication channel with another device held in the same hand that has an accelerometer to detect the vibration. Liang *et al.* (Liang and Kotz, 2017) showed that inertial sensor data from a smart ring can be correlated with mouse movements to verify the continued interaction of the user with a workstation. For explicit authentication, Roshandel *et al.* (Roshandel et al., 2014) showed that finger motion data can be used to authenticate the user of a smart ring while air-writing a signature or tracing it on a flat surface with the finger. To the best of our knowledge, we are the first to propose the use of inertial sensors on a ring to authenticate the user via implicit gesture biometrics.

8 CONCLUSION

In this paper, we showed that inertial sensor data from a smart ring can be used to authenticate the wearer. In a mobile payment context, we showed that a smart ring user can be authenticated implicitly with a single tap gesture with an EER of 0.04. We also showed that inertial sensor data from a smart ring can be used to authenticate the user when making a smartwatch payment, and *vice versa*, opening the possibility for either device to be used as an implicit second factor to support the other. This is particularly applicable in cases where the other device lacks inertial sensors of its own. In an access control context, we showed that a smart ring user can be authenticated with a single knock gesture with an EER of 0.06 (or 0.01 with a smartwatch). Our results further show that a smart door, which has inertial sensors embedded inside it, may be able to authenticate or identify the user knocking on it without the need for other devices. We demonstrated that our models can be trained quickly

(on very few samples) and that they provide resistance against an active attacker who observed (and heard) the victim's gestures.

ACKNOWLEDGEMENTS

This work was supported financially by Mastercard; the Engineering and Physical Sciences Research Council [grant number EP/P00881X/1]; and the PETRAS National Centre of Excellence for IoT Systems Cybersecurity [grant number EP/S035362/1]. The authors would like to thank these organisations for their support and Genki Instruments for their collaboration and technical support.

REFERENCES

Acar, A., Aksu, H., Uluagac, A. S., and Akkaya, K. (2020). A usable and robust continuous authentication framework using wearables. In *IEEE Transactions on Mobile Computing (TMC)*.

Ciuffo, F. and Weiss, G. M. (2017). Smartwatch-based transcription biometrics. In *IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference*.

Griswold-Steiner, I., Matovu, R., and Serwadda, A. (2017). Handwriting watcher: A mechanism for smartwatch-driven handwriting authentication. In *IEEE International Joint Conference on Biometrics (IJCB)*.

Griswold-Steiner, I., Matovu, R., and Serwadda, A. (2019). Wearables-driven freeform handwriting authentication. In *IEEE Transactions on Biometrics, Behavior, and Identity Science*.

Huhta, O., Shrestha, P., Udar, S., Juuti, M., Saxena, N., and Asokan, N. (2016). Pitfalls in designing zero-effort deauthentication: Opportunistic human observation attacks. In *Network and Distributed System Security Symposium (NDSS)*.

Johnston, A. H. and Weiss, G. M. (2015). Smartwatch-based biometric gait recognition. In *IEEE International Conference on Biometrics Theory, Applications, and Systems (BTAS)*.

Lee, W. H. and Lee, R. B. (2016). Implicit sensor-based authentication of smartphone users with smartwatch. In *ACM Hardware and Architectural Support for Security and Privacy (HASP)*.

Liang, G. C., Xu, X. Y., and Yu, J. D. (2017). User authentication on wearable devices based on punch gesture biometrics. In *International Conference on Information Science and Technology*.

Liang, X. and Kotz, D. (2017). Authoring: Wearable user-presence authentication. In *ACM Workshop on Wearable Systems and Applications (WearSys)*.

Mare, S., Markham, A. M., Cornelius, C., Peterson, R., and Kotz, D. (2014). Zebra: Zero-effort bilateral recurring authentication. In *IEEE Symposium on Security and Privacy (S&P)*.

Mare, S., Rawassizadeh, R., Peterson, R., and Kotz, D. (2019). Continuous smartphone authentication using wristbands. In *Workshop on Usable Security and Privacy (USEC)*.

Nassi, B., Levy, A., Elovici, Y., and Shmueli, E. (2016). Handwritten signature verification using hand-worn devices. In *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. In *Journal of Machine Learning Research*.

Roshandel, M., Munjal, A., Moghadam, P., Tajik, S., and Ketabdar, H. (2014). Multi-sensor finger ring for authentication based on 3d signatures. In *Human Computer Interaction. Advanced Interaction Modalities and Techniques*.

Sen, S. and Kotz, D. (2020). Vibering: Using vibrations from a smart ring as an out-of-band channel for sharing secret keys. In *Journal of Pervasive and Mobile Computing*.

Shrestha, B., Mohamed, M., Tamrakar, S., and Saxena, N. (2016). Theft-resilient mobile wallets: Transrgessparently authenticating nfc users with tapping gesture biometrics. In *Annual Conference on Computer Security Applications (ACSAC)*.

Sturgess, J. (2023). Ringauth dataset. In *University of Oxford, DOI: 10.5287/ora-no9q44mzq*.

Sturgess, J., Eberz, S., Sluganovic, I., and Martinovic, I. (2022a). Inferring user height and improving impersonation attacks in mobile payments using a smartwatch. In *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*.

Sturgess, J., Eberz, S., Sluganovic, I., and Martinovic, I. (2022b). Watchauth: User authentication and intent recognition in mobile payments using a smartwatch. In *IEEE European Symposium on Security and Privacy (EuroS&P)*.

Wijewickrama, R., Maiti, A., and Jadliwala, M. (2021). Write to know: On the feasibility of wrist motion based user-authentication from handwriting. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*.

Yang, J., Li, Y., and Xie, M. (2015). Motionauth: Motion-based authentication for wrist worn smart devices. In *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*.

Yu, X., Zhou, Z., Xu, M., You, X., and Li, X. (2020). Thumbup: Identification and authentication by smartwatch using simple hand gestures. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*.