The Applicability of Ambient Sensors as Proximity Evidence for NFC Transactions

Carlton Shepherd[†], Iakovos Gurulian[†], Eibe Frank[‡],
Konstantinos Markantonakis[†], Raja Naeem Akram[†], Emmanouil Panaousis[§] and Keith Mayes[†]
[†]*Information Security Group Smart Card Centre, Royal Holloway, University of London, Egham, United Kingdom*
[‡]*Department of Computer Science, University of Waikato, Hamilton, New Zealand.*
[§] *University of Brighton, Lewes Road, Brighton, BN2 4GJ, East Sussex, UK.*
Email: {k.markantonakis, r.n.akram, keith.mayes}@rhul.ac.uk,
{Iakovos.Gurulian.2014, Carlton.Shepherd.2014}@live.rhul.ac.uk,
e.panaousis@brighton.ac.uk, eibe@waikato.ac.nz

***Abstract*—Near Field Communication (NFC) has enabled mobile phones to emulate contactless smart cards. Similar to contactless smart cards, they are also susceptible to relay attacks. To counter these, a number of methods have been proposed that rely primarily on ambient sensors as a proximity detection mechanism (also known as an anti-relay mechanism). In this paper, we empirically evaluate a comprehensive set of ambient sensors for their effectiveness as a proximity detection mechanism for NFC contactless-based applications like banking, transport and high-security access controls. We selected 17 sensors available via the Google Android platform. Each sensor, where feasible, was used to record the measurements of 1,000 contactless transactions at four different physical locations. A total of 252 users, a random sample from the university student population, were involved during the field trials. After careful analysis, we conclude that no single evaluated mobile ambient sensor is suitable for proximity detection in NFC-based contactless applications in realistic deployment scenarios. Lastly, we identify a number of potential avenues that may improve their effectiveness.**

1. Introduction

Contactless smart cards are susceptible to relay attacks [1]–[3], as are NFC-enabled mobile phones [4]–[7]. A relay attack is a passive man-in-the-middle attack in which an attacker extends the distance between a genuine payment terminal (point-of-service) and genuine contactless smart card (or NFC-enabled mobile device). This attack can enable a malicious user to access services for which the genuine user is eligible, such as paying for goods or accessing a building with physical access controls.

Quantifying the number of fraudulent activities where relay attacks are employed (on both smart card and NFC mobile phones) is a challenging task. Evidence exists, however, that academic work regarding attacks on smart cards has been adopted by real-world criminals [8]. In the domain of contactless smart cards, a potentially effective countermeasure has been distance bounding protocols [9, 10]. For NFC-enabled phones, anti-relay mechanisms – at least in academic literature – have comprised largely of ambient sensing (Section 2). In this paper, we investigate the ambient sensors available through the Android platform and construct a test-bed environment (Section 3) to evaluate their effectiveness as proximity detection mechanism for NFC-based contactless transactions (Section 4). The aim of this work is to provide empirical evidence of each ambient sensor's suitability as a proximity detection mechanism (Section 4.2).

1.1. Operational Environment

In this paper, we focus solely on NFC based contactless applications that emulate traditional contactless smart cards, particularly in the banking and transport sectors. In such domains, the evaluation of ambient sensors must adhere to the operational environment stipulated by industry standards and specifications. We list the salient ones as follows:

1) Proximity: Two devices are considered to be in proximity of each other if they are physically present within a distance of 3-5cm [11]–[13].
2) Transaction Duration: The transaction must complete within 500ms. In accordance with the EMV specifications, the maximum permitted time in which a contactless payment transaction should complete is 500ms [14]–[17]. From the banking sector's point of view, this time limit will be reduced gradually to 400ms from 2016 onward [14, 17, 18]. For transport-related transactions, the performance requirements are stricter, where transaction times should not exceed 300ms [18, 19].

1.2. Evaluation Scope

The suitability of a proximity detection mechanism for critical applications, such as banking, transport and (high-security) access control – the main focus of this paper – is based on its ability to uniquely pair measurements taken from a payment terminal and a payment instrument (in this case, a mobile handset) for a maximum duration of 500ms.

DOI 10.1109/SPW.2017.29

This is to establish confidence that the two devices are truly in close proximity (≈ 3-5cm) to each other. This measurement pair should be unique in a manner such that no other measurements can be paired successfully with the terminal's or payment instrument's measurements. The uniqueness of each measurement pair provides the effectiveness of an ambient sensor-based proximity detection.

In this paper, we continue to refer to contactless mobile payments due to the associated financial repercussions, and the attention this may attract from malicious actors. However, the discussion in this paper is equally relevant towards the deployment of NFC based contactless mobile solutions in other industry sectors, such as transport and access control.

There are three primary contributions of this paper:

1) A test-bed architecture and implementation used to evaluate various sensors on Android devices.
2) A data analysis framework and methodology for evaluating ambient sensor measurements under industry specifications.
3) An empirical evaluation of the effectiveness of ambient sensors as a proximity detection mechanism. This evaluation provides a foundation towards deciding which sensors to deploy in a target environment.

The implementation of the test-bed, data analysis and collected data sets are made available at: https://github.com/AmbientSensorsEvaluation/Ambient-Sensors-Proximity-Evaluation.git

2. Ambient Sensing in Mobile Payments

In this section, we briefly describe mobile phone-based contactless payments, relay attacks and a generic architecture for deploying ambient sensing as a proximity detection mechanism for countering relay attacks.

2.1. Contactless Mobile Devices and Relay Attacks

In NFC-based mobile contactless transactions, a mobile handset is brought into the radio frequency range (<3-5cm) of a payment terminal through which it can initiate a dialogue. During this, physical contact is not necessary and, in many cases, a second factor of authentication, e.g. biometrics or Personal Identification Number (PIN), is not required [12]. This renders it difficult to ascertain whether the genuine or relay device is in close proximity of the terminal. It should be noted, however, that even the use of a PIN or biometric may not thwart relay attacks effectively (notably the Mafia fraud attack [20]).

In a relay attack [5, 21, 22], shown in Figure 1, an attacker must present a malicious payment terminal to a genuine user and a masquerading payment instrument (mobile phone) to a genuine payment terminal. The goal of the malicious actor is to extend the physical distance of the communication channel between the victim's mobile phone and the payment terminal. The attacker has the potential to gain access to services using the victim's account if it successfully relays messages without detection. Existing literature, discussed in Section 2.3, argues that ambient sensor based proximity detection is an effective countermeasure to relay attacks. In this paper, we only evaluate the effectiveness of ambient sensors when used to detect the proximity of two devices – in the absence of a relay attack.

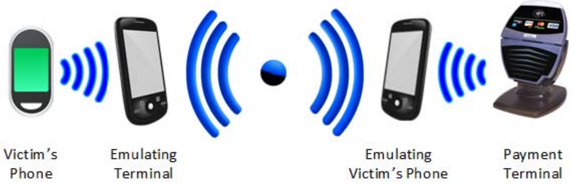


Figure 1: Overview of a Relay Attack

2.2. Ambient Sensors for Proximity Detection

An ambient sensor measures a physical attribute of its surroundings, such as temperature, light and sound. Modern smartphones and tablets are equipped with one or more of these sensors. The physical environment surrounding a smartphone (or a payment terminal) can potentially provide a rich set of attributes that might be unique to that location – the sound and lighting of a quiet, brightly-lit room, for example – and such information might be useful for proximity detection.

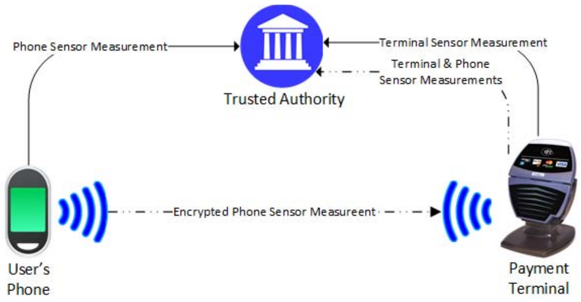

Phone Sensor Measurement
Trusted Authority
Terminal Sensor Measurement
Terminal & Phone Sensor Measurements
Encrypted Phone Sensor Measureent
User's Phone
Payment Terminal

Figure 2: Generic Deployments of Ambient Sensors as Proximity Detection Mechanism

Three ways exist in which a sensing-based proximity detection mechanism could be potentially deployed; Figure 2 illustrates the entities involved.

1) **Independent Reporting**. Both the smartphone and payment terminal collect sensor measurements independently and transmit these to a trusted authority (depicted as solid lines in Figure 2). The authority compares the sensor measurements, based on some predefined comparison algorithm with a set margin of error (threshold), and decides whether the devices are within sufficient proximity.

2) **Payment Terminal Dependent Reporting**. The smartphone encrypts its sensor measurements with a shared key between itself and the trusted authority, and transmits the encrypted measurements to the payment terminal. Next, the terminal sends the smartphone's measurements and its own to the trusted authority for comparison (shown as a double-dot-dash line in Figure 2).
3) **Payment Terminal (Localised) Evaluation**. The smartphone transmits its own measurement to the payment terminal, which then compares it with its own measurements to decide whether the smartphone is in proximity.

The overall deployment architecture falls under one of the above scenarios irrespective of how the user interacts with the payment terminal.

2.3. Related Work

Drimer et al. [23] and Ma et al. [24] showed how location-related data, namely using GPS (Global Positioning System), can be used to determine the proximity of two NFC mobile phones. Ma et al. use a ten second window with location information collected every second, which was subsequently compared across various devices. The authors report a high success rate in identifying devices within close proximity.

Halevi et al. [25] demonstrated the suitability of ambient sound and light for proximity detection. Here, the authors analyse measurements collected for 2 and 30 seconds duration for light and audio respectively using a range of similarity comparison algorithms. Although the scenarios are identical, the transaction duration does not conform to industry requirements for NFC-based contactless mobile transactions (Section 1.1). While the authors do not specify the number of transactions recorded at each location, the experiments show a high success rate of detecting co-located devices in various environments.

Varshavsky et al. [26] based their proximity detection mechanism on the shared radio environment of devices – the presence of WiFi access points and associated signal strengths – using the application of secure device pairing. This approach produced low error rates, recommending it as a proximity detection mechanism. While their paper did not focus on NFC-based mobile transactions, their techniques and methodology may still be applicable.

Urien et al. [27] use ambient temperature with an RFID/NFC authentication protocol for proximity detection. Using this method, they establish a secure channel by combining the timing channels in RFID, traditionally used in distance bounding protocols, in conjunction with ambient temperature. Their proposal, however, was not implemented and so there is no experimental evidence to evaluate its efficacy.

Mehrnezhad et al. [28] proposed the use of an accelerometer to provide assurance that the mobile phone is within proximity of the payment terminal. Their proposal requires the user to tap the payment terminal twice in succession, after which the sensor streams of the device and the payment terminal are compared for similarity. It is difficult to deduce the total time it took to complete one transaction in its entirety, but the authors use recording durations of 0.6–1.5 seconds.

TABLE 1: Related Sensing-based Anti-relay Mechanisms

Paper	**Sensor Used**	**Sample Duration**	**Contactless Suitability**
Ma et al. [24]	GPS	10 sec	Unlikely
Halevi et al. [25]	Audio	30 sec	Unlikely
	Light	2 sec	More Likely
Varshavsky et al. [26]	WiFi (Radio Waves)	1 sec	More Likely
Urien et al. [27]	Temperature	N/A	-
Mehrnezhad et al. [28]	Accelerometer	0.6 to 1.5 sec	More Likely
Truong et al. [29]	GPS Raw Data	120 sec	Unlikely
	Wifi	30 sec	Unlikely
	Ambient Audio	10 sec	Unlikely
	Bluetooth	12 sec	Unlikely
Shrestha et al. [30]	Temperature (T)	NA	Unlikely
	Precision Gas (G)	NA	Unlikely
	Humidity (H)	NA	Unlikely
	Altitude (A)	NA	Unlikely
	HA	NA	Unlikely
	HGA	NA	Unlikely
	THGA	NA	Unlikely

Truong et al. [29] evaluated four different sensors across recording durations of 10-120 seconds. Although the results were positive, such a long recording duration renders them unsuitable for realistic NFC-based mobile transactions. Moreover, the data collection set-up did not emulate a contactless transaction, either in the context of banking, transport or access control. However, the authors did discuss the impact of transaction duration on the real-world applicability of the results. For usability, transaction durations should be minimised – in the range of 5-15 seconds. They also concluded that measurements recorded beyond 10 seconds did not improve effectiveness.

Shrestha et al. [30] used bespoke hardware known as Sensordrone, with a number of ambient sensors, but did not evaluate the commodity ambient sensors available on commercial handsets, did not provide the sample duration, and only mentioned that data from each sensor was collected for a few seconds. It is difficult to evaluate the proposed technique in the context of NFC contactless mobile transactions in the banking and transport sector under their specified requirements. The results related to barometric air pressure were similar to what we have calculated. Sensors like Precision Gas and Altitude are not available on commodity off the shelf Android smart phones.

Karapanos et al. [31] employed the use of sound as a supporting mechanism for two-factor authentication, using a recording duration of 5 seconds. Given that this deployment is not related to NFC-based mobile transactions, such a long duration can be justified.

We summarise the related work in Table 1, and use sensor sampling durations to determine whether a given approach is suitable for mobile contactless transactions. *'Unlikely'* are those proposals whose sample duration is so large that they may not be adequate for mobile services that replace contactless cards. Those whose durations are considered more reasonable are labelled as *'More Likely'* in Table 1. Note, however, that even schemes denoted as *'More Likely'* may not be suitable for time-critical domains, where strict time limits are imposed in which the transaction must be completed (see Section 1.1). In such domains, the goal is to serve people as quickly as possible to maximise customer throughput, in addition to determining whether a transaction is successful and, indeed, permitted. Here, optimal transaction durations are in milliseconds, rather than seconds.

In this paper, we do not repeat the experiment as the existing literature suggests, as their set-up does not conform to the industry requirements – especially the transaction duration (500ms). The only common aspect is that we have evaluated the ambient sensors for the same domain – proximity detection in contactless transactions.

3. Framework for Evaluating Ambient Sensors

In this section, we describe the test-bed that was developed to test, analyse and evaluate the effectiveness of using mobile sensors as a proximity detection mechanism. The results of the evaluation are presented in Section 4.

3.1. Test-bed Architecture

To empirically evaluate each sensor, we developed a experimentation test-bed – the details of which are discussed in this section. Two applications were implemented and installed on a pair of Android devices: one emulating a payment terminal (PT) and the other acting as the payment instrument (PI), a mobile phone. When the devices come sufficiently close, an NFC connection is established and both begin recording data using a specified sensor. After collecting measurements for 500ms, in line with the requirements specified in Section 1.1, each device stores the recorded data in a local database. During field trials, one mobile phone was fixed as a terminal and the second mobile phone was free of any restriction.

Figure 3 shows this in more detail. Bringing the two devices together (< 3cm) causes the PT application to send the first message to the PI over NFC, stating which sensor it uses in the transaction and a unique transaction ID. After this message is received by the PI, both applications initiate the process to record a sensor for 500ms.

After collecting the measurements, the PI validates the data it received from the terminal – whether the transaction ID and chosen sensor match that of the terminal (shown in message one in Figure 3) – and returns an acceptance or rejection message accordingly. This validation process ensures that both devices were recording data from the same sensor. Finally, PT performs the same process, ensuring that both devices used the same transaction ID and recorded from the same sensor. The measurement is rejected in the event that devices recorded data for differing transaction IDs or sensors. Upon validation, the devices save the measurements in their local databases. The database is designed to hold measurements for each transaction, which are used in the off-line analysis of each sensor.

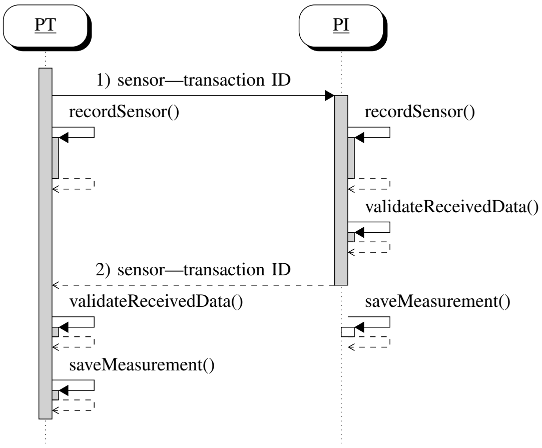


Figure 3: Measurement Recording Overview

3.2. Data Collection Framework

We test each sensor in four different locations around the university – the lab, cafeteria, dining hall and library – to account for the influence of different physical locations on sensor measurements. A field trial was conducted in each location with 252 participants that carried out a varying number of transactions. Each participant used the PI provided by us and was given free reign with how they interacted with the PT for each transaction, i.e. they could tap it once, hold it extremely close without touching, or tap and hold it to the device. This is to closely replicate the conditions in which they would conduct a regular contactless transaction. The data collection at each of the locations was collected over an eight hours period (0900-1700hours) with irregular gaps between transactions over the course of four days.

Four devices were used in the experiments, forming two PT–PI pairs. The first pair consisted of two Nexus 9 tablets, while the second pair comprised two Android smartphones: a Nexus 5, assuming the role of the payment terminal, and a Samsung Galaxy S5 mini (SGS5 mini), which acted as the payment instrument. The availability of the sensors on each device is listed in Table 2.

Some sensors, such as Bluetooth, GPS, Rotation Vector and WiFi – although present on the devices – returned no or very few data points within the 500ms timeframe (>99% sensor failure[1]). Two sensors, for humidity and temperature, are relatively uncommon among Android devices and none of our tested devices contained them. For completeness, we used two Samsung Galaxy S4 (SGS4) smartphones containing these to include in our evaluation. However, for these sensors, measurements were recorded for less than 6% of the transactions when recorded for 500ms. Consequently, having

[1] Detailed in Section 4.1.2 and Table 4

TABLE 2: Sensor Availability

Sensors	Nexus 9 (1)	Nexus 9 (2)	Nexus 5	SGS5 mini
PI-PT Pair: Nexus 9 (1) → Nexus 9 (2)				
Accelerometer	✓	✓	✓	✓
Bluetooth	$*$	$*$	$*$	$*$
GRV†	✓	✓	$*$	✓
GPS	$*$	$*$	$*$	$*$
Gyroscope	✓	✓	✓	✓
Magnetic Field	✓	✓	✓	✓
Network Location	✓	✓	✓	✓
Pressure	✓	✓	✓	✗
Rotation Vector	$*$	$*$	$*$	$*$
Sound	✓	✓	✓	$*$
WiFi	$*$	$*$	$*$	$*$
PI-PT Pair: SGS5 mini → Nexus 5				
Gravity	○	○	✓	✓
Light	$*$	$*$	✓	✓
Linear Acceleration	○	○	✓	✓
Proximity	✗	✗	✓	✓
Unsupported				
Relative Humidity	‡	‡	‡	‡
Ambient Temperature	‡	‡	‡	‡

✓: Working properly. ✗: Not present on device. $*$: Technical limitations.
‡: Evaluated using Samsung Galaxy S4. ○: Returned only zero-values.
† Geomagnetic Rotation Vector.

failed to record any data points in such a large number of cases, we omitted these sensors from subsequent analysis. A minimum of 1,000 transactions were recorded for each sensor – comprising measurement pairs for which both PT and PI have valid sensor data.

The Android operating system (OS) returns data captured by a sensor in time intervals set by the application. To prevent unnecessary power consumption, however, the OS returns sensor values to the application only when the values have changed from the past measurement. Note that the sound sensor (microphone) captures data in a continuous, uninterrupted stream; in this instance, the applications converted the recorded amplitudes into sound pressure levels (in decibels) before storing the values in their respective databases. For Bluetooth, the OS returns data every time a new Bluetooth device is discovered nearby; with WiFi, this is after the device has scanned and detected the presence of nearby access points.

The recorded sensor measurements were stored in XML form in each database. A new child element was created containing the sequence ID of the measurement, the timestamp (initialised to zero at the start of the transaction), along with the data for each returned measurement. The sequence ID consisted of the date and time the transaction occurred, the location in which it was captured, and a transaction ID. The transaction ID is a random, 7-byte string generated by the terminal used to link the measurements of each device to produce a PT–PI pair. Occasionally, the NFC connection was disrupted, primarily when the devices were moved apart before the transaction was completed. To address this, the transaction ID was used in conjunction with the sequence ID to detect and exclude these measurements prior to analysis.

4. Ambient Sensor Evaluation

In this section, we describe our analysis and evaluation methodology based on two methods. The first evaluates sensor data using threshold-based similarity metrics used in existing literature, while the second employs the use of machine learning. The evaluation mechanisms discussed in this section are based on those used in existing literature (see Section 2.3) where they have been shown to effectively detect the proximity of two devices. Finally, we present the results of our individual sensor analysis.

After retrieving the databases from PT and PI, the set of all transactions, T, was produced using the shared IDs generated during data collection. Each transaction can be represented as a set of PT and PI values, PT_i and PI_i, with the same shared ID, i.e. $T_i = (PT_i, PI_i)$. Note that each device measures each sensor at potentially different time intervals (accounting for clock variances), which may produce an unknown total number of measurements for each device per transaction. That is, the number of measurements in PT_i is not necessarily that of PI_i.

4.1. Method 1: Similarity Analysis and Evaluation Criteria

A Python application was developed for analysing the transaction measurements from the application databases, using the SciPy library [32] for numerical computation.

$$2r \arcsin\left(\sqrt{\sin^2\frac{\phi_2-\phi_1}{2}+\cos\phi_1\cos\phi_2\sin^2\frac{\lambda_2-\lambda_1}{2}}\right) \tag{1}$$

$$MAE(PT_i, PI_i) = \frac{1}{N}\sum_{j=0}^{N}|PT_{i,j} - PI_{i,j}| \tag{2}$$

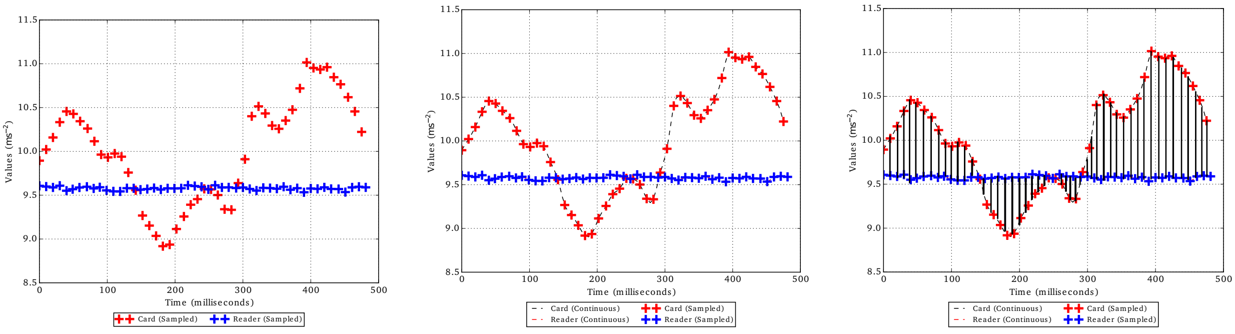
$$corr(PT_i, PI_i) = \frac{covariance(PT_i, PI_i)}{\sigma_{PT_i} \cdot \sigma_{PI_i}} \tag{3}$$

$$M = \sqrt{x^2 + y^2 + z^2} \tag{4}$$

To compare (PT_i, PI_i), we measure the similarity of or distance between the two. This is measured differently according to sensor type due to the differences in coordinate systems and dimensions used across sensors. Measurements are recorded in three dimensions for the accelerometer, for example, while location returns a longitude-latitude pair on Earth. Due to this, we devised three methods of dealing with the diversity of reported measurements.

For network location, we used the Haversine formula (Eq. 1), which measures the geographic distance between two latitude and longitude pairs, $\{(\phi_1, \lambda_1), (\phi_2, \lambda_2)\}$. In Eq. 1, '$r$' represents the radius of Earth.

For the remaining sensors, distance and similarity respectively were measured using the *Mean Absolute Error* (MAE, Eq. 2) and *Correlation Coefficient* (Eq. 3), as used in [28], between the signals of PT_i and PI_i. This was performed after linear interpolation to mitigate the effects of inconsistent clocks between devices (Figure 4). Furthermore, certain sensors – the accelerometer, gyroscope, magnetic



(a) Step One (b) Step Two (c) Step Three

Figure 4: Linear Interpolation to Mitigate the Effect of Missing and Inconsistent Sampling Rate – An Example from Accelerometer-based Transactions

field, rotation vector and GRV sensors – produce a vector of values comprising x, y and z components. In these instances, the vector magnitude (Eq. 4) was used as a general-purpose method for producing a single, combined value prior to computing the MAE and correlation coefficient. Next, MAE was computed by applying Eq. 2 directly, while for correlation, this was found after measuring the covariance and computing the standard deviations, σ_{PT_i} and σ_{PI_i}, of the data points in PT_i and PI_i.

4.1.1. Calculating the FPR, FNR and EER. We compute the $MAE(PT_i, PI_i)$ and $corr(PT_i, PI_i)$ for each successful transaction. Next, we calculate the False Positive Rate (FPR), False Negative Rate (FNR) and Equal Error Rate (EER) of each sensor by the testing MAE and $corr$ values of genuine pairs[2], (PT_i, PI_i), against the MAE and $corr$ values of unauthorised pairs (PT_i, PI_j) with a threshold, t. An ideal similarity metric, V, would produce $V(PT_i, PI_i) < t$ and $V(PT_i, PI_j) > t$ for all possible pairs. We constructed these unauthorised pairs by exhaustively matching each PT_i with every PI_j measurement belonging to another transaction ($i \neq j$). The FPR and FNR are calculated using Equation 5, where FP, FN, TP and TN represent the total number of False Positives, False Negatives, True Positives and True Negatives respectively for a given threshold.

$$FPR = \frac{FP}{FP + TN} \qquad FNR = \frac{FN}{FN + TP} \tag{5}$$

4.1.2. Individual Sensor Results. The aim of our evaluation is to investigate to what extent legitimate and illegitimate transactions can be identified using these similarity metrics. For a transaction between two co-located devices, $MAE(PT_i, PI_i) \approx 0$ and $corr(PT_i, PI_i) \approx 1$, while for a PT and a PI device in differing locations, i.e. (PT_i, PI_j), the distance and correlation should be sufficiently large. What is considered 'sufficient' is determined through finding a suitable threshold, t, which permits all legitimate transactions while denying those which are illegitimate, i.e. $V_i(PT_i, PI_i) < t$ and $V_{ij}(PT_i, PI_j) > t$, as mentioned previously. For each individual sensor, we aim to find an optimal value of t, its error rate and reliability, e.g., whether it collected measurements consistently and correctly across 1,000 transactions.

TABLE 3: Optimum Thresholds and Associated EERs

Sensors	**Optimum Threshold**$_{MAE}$	EER_{MAE}	**Optimum Threshold**$_{corr}$	EER_{corr}
Accelerometer	0.784	0.434	0.596	0.458
Ambient Temperature	–	–	–	–
Bluetooth	–	–	–	–
GRV	0.499	0.384	0.556	0.486
GPS	–	–	–	–
Gyroscope	0.614	0.443	0.636	0.441
Magnetic Field	76.12	0.323	0.495	0.384
Network Location	8.532	0.369	N/A*	N/A
Pressure	2.787	0.270	0.329	0.492
Rotation Vector	1.281	0.498	0.011	0.466
Relative Humidity	–	–	–	–
Sound	8.22	0.417	-0.022	0.488
WiFi	–	–	–	–
Gravity	9.93e-3	0.429	0.596	0.424
Light	182.1	0.488	0.020	0.496
Linear Acceleration	1.361	0.496	-0.020	0.426
Proximity	N/A†	N/A	N/A	N/A

*Insufficient data to calculate correlation
†All transactions contained the same value for both devices.

We generate FPR and FNR curves for MAE and $corr$ for every sensor for which we were able to collect data. The point of intersection for these curves provides an optimal threshold for MAE and $corr$ based on its associated EER, i.e., the rate at which the acceptance and rejection errors are equal.

Practically speaking, in a wide-scale deployment of an ambient sensing proximity detection mechanism, a single threshold should be defined. The terminal (or third party) would store this threshold (Section 2.2), and if the similarity of the terminal's and device's sensor readings was within this, then the transaction would be assumed to be legitimate, i.e., both devices in close proximity. However, setting a threshold of this nature invariably incurs some rate of false positives and false negatives. The intersection of FPR and FNR provides us with the proportion of potentially invalid transactions which might pass as genuine (false positives)

[2] Those collected during field trials.

TABLE 4: Usability and Reliability Analysis

Sensors	Total Transactions	Transaction Failures	Sensor Failures
Accelerometer	1025	13 (1.26%)	0 (0%)
Bluetooth	101	1 (0.99%)	99 (99%)
GRV	1019	8 (0.78%)	0 (0%)
GPS	101	1 (0.99%)	100 (100%)
Gyroscope	1022	11 (1.07%)	0 (0%)
Magnetic Field	1027	17 (1.65%)	0 (0%)
Network Location	1053	15 (1.42%)	960 (96%)
Pressure	1018	10 (0.98%)	0 (0%)
Rotation Vector	1023	14 (1.36%)	0 (0%)
Sound	1047	4 (0.38%)	0 (0%)
WiFi	100	0 (0%)	100 (100%)
Gravity	1165	143 (12.27%)	0 (0%)
Light	1057	37 (3.50%)	0 (0%)
Linear Acceleration	1175	159 (13.53%)	3 (0.3%)
Proximity	1071	58 (5.41%)	0 (0%)
Ambient Temperature	50	0 (0%)	47 (94%)
Relative Humidity	50	0 (0%)	47 (94%)

and the proportion of genuine transactions being rejected (false negatives). The goal of a malicious entity would be to carry out relay attacks such that the sensor measurements at the terminal and mobile phone remained within the predefined threshold. A threshold with a higher FPR provides a large working space to the attacker, whereas a higher FNR will reduce the usability of the scheme, potentially frustrating consumers by rejecting legitimate transactions. Table 3 lists the optimum thresholds and associated EERs for each tested sensor.

Besides investigating the EERs of sensors and the effect this has on their suitability for NFC mobile services, we evaluate the reliability and potential usability of the selected sensors. Table 4 presents our findings regarding the proportion of failed transactions and sensor failures. To collect 1,000 transactions for each sensor, as explained in Section 3.2, we requested 252 users to present the PI to the PT as many times as they preferred. We established walk-in counters at four different locations of the university campus; students walking nearby were invited to assist us in the trial. Demographic data about the students was not collected, as sensors are not used for user identification, but simply to assure that two devices were in close proximity to each other during a transaction. At times, transactions were not registered during this process, usually due to the user moving the handset away too quickly, and was the primary cause of transaction failures[3] (no shared measurements between the PT and PI) represented in Table 4. The rate of sensor failures, in the same table, represents the situation when the transaction was successfully completed on both the PT and PI, but where one or both devices failed to record any data in the 500ms timeframe. The percentage of transaction failures relates to the total transactions, while sensor failures are measured with respect to the number of successful transactions. The transaction failure rates represent the difficulty in using the sensors by the user, while the sensor failure rates reflects their reliability.

4.2. Method 2: Machine Learning Analysis

The $MAE(PT_i, PI_I)$ distance measure in Equation 2 and the $corr(PT_i, PI_i)$ similarity measure in Equation 3 give each pair of individual measurements $PT_{i,j}$ and $PI_{i,j}$ the same weight when PT_i and PI_I are compared. However, it is conceivable that not all time slots $PT_{i,j}$ and $PI_{i,j}$ are equally important when the task is to discriminate between genuine and unauthorised transaction pairs. Moreover, it is possible that discrimination becomes possible by modelling complex non-linear interactions between the individual differences $|PT_{i,j} - PI_{i,j}|$—interactions that cannot be captured by simple similarity measures.

To investigate this, we applied a collection of supervised machine learning algorithms to the problem, including algorithms that are able to model (in an approximate manner) arbitrary non-linear interactions given enough training data. The data for learning was created by treating each pair (PT_i, PI_i) for a particular sensor as a labeled observation $(\vec{x}, y)$, where the label y is either *genuine* or *unauthorised* and the feature vector $\vec{x}$ consists of the individual differences $|PT_{i,j} - PI_{i,j}|$ for the pair (PT_i, PI_i).

When applying machine learning to a classification problem such as this one, it is important to test the discriminative ability of the model inferred by the learning algorithm to a set of observations that have not been used for learning the model; the labelled data must be separated into a so-called *training set* and a *test set*. The learning algorithm is applied to the training set to build a model, while the test data is used to measure the model's discriminative performance. We use equal error rate to measure performance, using the confidence scores associated with the model's classifications to rank observations according to their estimated likelihood of being genuine transactions.

Given the number of observations available in our datasets, a single train-test experiment is not sufficient to establish a reliable estimate of equal error rate. A standard procedure in machine learning is to perform 10-fold stratified cross-validation, where the data is shuffled and split into 10 disjoint test sets each containing the same number of observations. The data is also stratified so that the proportion of genuine and unauthorised transactions is the same in each set. Then the algorithm is run 10 times, once for each test set, where the observations not in the corresponding test set are used for training the model, and the observations in the test set are used to measure its equal error rate. This yields 10 estimates of equal error rate, which are averaged to obtain the final performance estimate. To reduce the variance of the performance estimate even further, we repeat 10-fold cross-validation 10 times, each time shuffling the data before it is split into 10 test sets. This yields 100 estimates of equal error rate and we report the mean and standard deviation of these estimates for each learning algorithm and sensor in Table 5.

Table 5 shows results for the six learning algorithms we evaluated, including both parametric and non-parametric

[3] These failed transactions were not included in the data analysis and results represented in Table 4, which is based on the successful 1000 transactions.

TABLE 5: Estimated EER for machine learning algorithms, obtained by repeating stratified 10-fold cross-validation 10 times

Dataset	Random Forest	Naive Bayes	Logistic Regression	Decision Tree	Support Vector Machine	Multilayer Perceptron
Accelerometer	62.6±2.4	50.9± 2.6	52.6± 2.3	50.0± 0.0	**49.8**± 2.5	55.1± 2.5
GeomagneticRotationVector	**43.5**±2.1	44.7± 2.4	47.4± 3.1	50.0± 0.0	48.9± 3.6	45.0± 2.6
Gravity	87.4±1.8	57.9± 2.0	57.9± 2.4	**50.0**± 0.0	**50.0**± 2.6	74.6±11.2
Gyroscope	68.3±2.7	**49.9**± 2.4	54.3± 2.4	50.0± 0.0	51.1± 2.5	51.4± 2.5
Light	57.6±2.6	51.5± 2.4	53.3± 2.5	**50.0**± 0.0	50.8± 2.4	51.3± 2.8
LinearAcceleration	60.3±2.5	50.7± 2.7	54.3± 2.3	**50.0**± 0.0	**50.0**± 2.1	55.4± 2.8
MagneticField	**29.2**±2.1	31.9± 2.0	32.2± 2.0	41.5± 1.5	39.8± 4.6	32.9± 2.6
Pressure	10.3±1.0	10.7± 1.0	28.7± 1.3	**9.2**± 5.4	31.9± 4.5	11.4± 1.9
Proximity	49.9±3.1	53.7± 6.9	**47.6**±18.8	50.0± 0.0	54.3± 25.4	50.8±19.7
RotationVector	**27.6**±4.6	56.3±24.3	59.6±23.3	50.0± 0.0	51.3± 24.3	48.8±24.5
Sound	**28.8**±1.9	31.4± 2.2	31.0± 2.1	34.7±13.6	41.1± 4.1	30.6± 2.0

approaches, as implemented in the WEKA machine learning software [33]. We used default parameter settings for the learning algorithms unless otherwise specified. The random forest method [34] learns an ensemble classifier consisting of 100 semi-random decision trees from bootstrap replicates of the training data. This classification method is able to model arbitrarily complex interactions and is known to be a general-purpose approach that performs well without parameter tuning. The naive Bayes classifier fits a multivariate Gaussian distribution with a diagonal covariance matrix to the data for each classification (genuine vs. unauthorised), thus assuming conditional independence of the features in the data, and uses Bayes' rule to obtain class probability estimates. Logistic regression fits a linear model using maximum conditional likelihood. The well-known C4.5 [35] algorithm is used to grow decision tree classifiers. We also include linear classification using support vector machines, which are trained using the SMO [36] algorithm. A logistic regression model is fit to the output of the support vector machine to obtain class probability estimates. The last learning method in our collection is a multilayer perceptron, a type of artificial neural network, with one hidden layer containing 10 units, which is trained using the *MLPClassifier* method in WEKA.

The results in Table 5 are largely in line with those observed earlier; the lowest equal error rate for each sensor is shown in bold. No useful discriminative signal appears to be present in the accelerometer, geometric rotation vector, gyroscope, light, linear acceleration, gravity, and proximity data. Decision-tree-based methods give the best results for the remaining sensors. Magnetic field, rotation vector, and sound data provide some discriminative ability, but the equal error rate remains close to 30%. The best result is obtained on the pressure data, with an equal error rate of approximately 10%. Pressure was also the most informative sensor in the earlier experiments, with 27% equal error rate for the MAE distance metric. Although the result obtained using tree-based machine learning is substantially better, discrimination is still significantly too inaccurate to be used for authentication in practice.

5. Outcome and Future Directions

As discussed previously, the higher the EER, the greater the likelihood that an attack passes undetected and that a genuine transaction is rejected. Based on our analysis, it is difficult to recommend any of the sensors individually for a high security deployment application, such as banking and transport. These sensors, however, might be appropriate for low-security access control, but we recommend that a thorough analysis of the sensors and their performance in the chosen domain is performed prior to deployment.

One potential reason that related research in this domain has achieved different results is due to the larger transaction durations and limited field trials in other work. The sample duration limit imposed during our experiments was in line with the performance requirements of an EMV application as discussed in Section 1.1, i.e. 500 milliseconds. Additionally, transportation is one of the biggest application areas of contactless smart cards, along with banking; in this domain, the recommended duration for a transaction is far lower, in the range of 300–400 milliseconds. Imposing a limit of 500 millisecond in our experiments is, therefore, an upper-bound of the operational requirements for two major areas where mobile-based contactless transactions may be applied.

One potential method of improving performance is to increase recording duration. This is indicated in related work, which yields more promising results, but uses recording durations far longer than the industry limits used in this study. This would be possible if users initiated recording in advance of the actual transaction, and we consider this as one of our future research directions. We do have reservations about this proposal, however. Firstly, it requires users to pre-empt transactions, which, in realistic situations, may require an additional task to be performed in advance before they can use their mobile device. Secondly, a user may have to initiate recording at a distance from the PT to potentially give PI more time to measure a larger sample. Both of the reservations involve an additional step to be performed by the user that would detract from usability, in a way the whole purpose of contactless transactions. This, however, does not provide a measurement of proximity as defined by banking and transport specifications. Furthermore, we do not agree with the argument that proximity detection is unnecessary because a PIN or biometric is required to use a payment application. In the relay attack variant known as a Mafia Attack [37], a malicious terminal is deployed by an attacker to trick genuine users to use their smartphones with it. In this scenario, a PIN or biometric cannot protect against relay attacks.

During our experiments, we realised that sensors and their associated platforms may not have the maturity required for a wide-scale deployment as a proximity detection mechanism for NFC-enabled phones. Variations in sensor availability, the sensor measurements themselves, whether the platform's sensing architecture is affected by other applications running simultaneously, and differences in minimum sampling rates, may vary across device manufacturers. We contend that mobile sensors have a considerable way to go before achieving the necessary interoperability, standardisation and performance requirements to enable an effective sensing-based proximity detection mechanism.

From the work carried out and the results presented in this paper, we can claim with a high degree of confidence that mobile sensors, at least in their current state on Google Android devices, are not suitable for use as an anti-relay mechanism. This is especially pertinent in the case of applications with high security requirements, such as banking, transport and access-control at highly sensitive sites. It may be argued that these sensors might be suitable for low-risk application that do not have stringent transaction time limits and distance bounding assurance requirements. However, the developer ought to consider the risks highlighted in this paper, i.e., EER and reliability rates. To this end, we provide EER tables (Table 3 and 5) that indicates the effectiveness of the respective sensor and its associated risk if it were deployed.

After carrying out the experimentation with selected handsets in this paper, we extended the test-bed to include additional handsets like the Samsung Galaxy S4 (Model: GT-I9505) with additional sensors. In these tests the outcome was similar to the initial test, providing further evidence for our results.

As part of our future research, we are currently experimenting with:

- Collecting and evaluation a large data set of actual relay attacks using these sensors, and investigating if and how a relay attack in the field is reflected in its sensor measurements.
- Combining sensor measurements with time slicing and sensor fusion: only one sensor is measured at a time, but over the duration of the transaction multiple sensors could be used.

6. Conclusion

The aim of the paper was to evaluate and analyse a range of sensors present in modern day mobile devices, and determining which sensors, if any, would be suitable as a proximity detection mechanism in the domain of NFC smartphone transactions. We listed 17 sensors accessible through the Android platform, before limiting it to those which are widely-available. In existing literature, only five sensors have been proposed as an effective proximity detection mechanism (listed in Table 1). In this paper, we extend this with ten additional sensors by evaluating their effectiveness as a proximity detection mechanism on NFC-enabled mobile devices. In total, we implemented and evaluated 17 sensors, but WiFi, Bluetooth, Ambient Temperature, Relative Humidity and GPS were dropped after exhibiting high failure rates in initial tests. The scope of our analysis focuses on NFC-enabled mobile devices that emulate traditional smart card services, such as transportation and banking. Any analysis or recommendation in this paper regarding these sensors is restricted to mobile contactless transactions that aim to substitute for the contactless smart card transactions in such high security applications.

The field of ambient sensors for proximity detection in NFC-based mobile services is expanding, as illustrated by the number of recent proposals. In this paper, we extend the discussion to a large set of ambient sensors. We evaluate the suitability of sensors proposed currently and investigate a range of sensors not yet explored as an anti-relay mechanism in related work. Table 2 shows that we have undertaken a comprehensive evaluation of ambient sensors for proximity detection (seventeen in total). In existing literature, only a subset of these sensors are proposed and evaluated as proximity detection mechanism (Table 1). Most of this existing literature has shown that the ambient sensors are effective. However, in our empirical evaluation with a real world operational environment for banking, transport and access control has shown to the contrary that these sensors do not provide an effective proximity detection mechanism.

It is neither evaluated nor claimed that similar results will be produced in other deployment scenarios where distance bounding and transaction time limits are not as stringent. It should be considered that the transaction time limit and operating distance are not set arbitrarily, but rather in compliance with industry-wide requirements, as stipulated by EMV and transport specification bodies. The experimentation and analysis carried out as part of this paper showed that none of the sensors were individually suitable to be deployed as a proximity detection mechanism for NFC-based mobile transactions. Finally, we release the source code of our test-bed publicly available, along with our collected data sets, for open scrutiny and further analysis.

References

[1] G. Hancke, K. Mayes, and K. Markantonakis, "Confidence in smart token proximity: Relay attacks revisited," *Computers & Security*, vol. 28, no. 7, pp. 615 – 627, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404809000595

[2] G. P. Hancke, "Practical Attacks on Proximity Identification Systems (Short Paper)." in *IEEE Symposium on Security and Privacy*. IEEE CS, 2006, pp. 328–333. [Online]. Available: http://dblp.uni-trier.de/db/conf/sp/sp2006.html#Hancke06

[3] Z. Kfir and A. Wool, "Picking virtual pockets using relay attacks on contactless smartcard," in *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*. IEEE, 2005, pp. 47–58.

[4] L. Francis, G. P. Hancke, K. Mayes, and K. Markantonakis, "Practical NFC Peer-to-Peer Relay Attack Using Mobile Phones." in *RFIDSec*, ser. LNCS, S. B. O. Yalcin, Ed., vol. 6370. Springer, 2010, pp. 35–49. [Online]. Available: http://dblp.uni-trier.de/db/conf/rfidsec/rfidsec2010.html#FrancisHMM10

[5] ——, "Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones." *IACR Cryptology ePrint Archive*, vol. 2011, p. 618, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/iacr/iacr2011.html#FrancisHMM11

[6] G. Madlmayr, J. Langer, C. Kantner, and J. Scharinger, "NFC devices: Security and privacy," in *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*. IEEE, 2008, pp. 642–647.

[7] M. Roland, J. Langer, and J. Scharinger, "Applying relay attacks to Google Wallet," in *Near Field Communication (NFC), 2013 5th International Workshop on*, Feb 2013, pp. 1–6.

[8] H. Ferradi, R. Geraud, D. Naccache, and A. Tria, "When Organized Crime Applies Academic Results." *IACR Cryptology ePrint Archive*, p. 20, 2015.

[9] G. P. Hancke and M. G. Kuhn, "An RFID Distance Bounding Protocol," in *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, ser. SECURECOMM '05. Washington, DC, USA: IEEE CS, 2005, pp. 67–73. [Online]. Available: http://dx.doi.org/10.1109/SECURECOMM.2005.56

[10] R. Trujillo-Rasua, B. Martin, and G. Avoine, "The Poulidor distance-bounding protocol," in *Radio Frequency Identification: Security and Privacy Issues*. Springer, 2010, pp. 239–257.

[11] "How to Optimize the Consumer Contactless Experience? The Perfect Tab," MasterCard, Online, March 2014.

[12] "Emv contactless specifications for payment systems: Book a - architecture and general requirements," EMVCo, LLC, Specification Version 2.5, March 2015.

[13] "EMV and NFC: Complementary Technologies that Deliver Secure Payments and Value-Added Functionality," Smart Card Alliance, White Paper, October 2012.

[14] "The Future of Ticketing: Paying for Public Transport Journeys Using Visa Cards in the 21st Century," VISA, Whitepaper, January 2013.

[15] "MasterCard Contactless Performance Requirement," MasterCard, Online, March 2014.

[16] "Emv contactless specifications for payment systems: Book d - emv contactless communication protocol specification," EMVCo, LLC, Specification Version 2.6, March 2016.

[17] "Transactions acceptance device guide (tadg)," VISA, Specification Version 3.0, May 2015.

[18] "Transit and Contactless Open Payments: An Emerging Approach for Fare Collection," Smart Card Alliance Transportation Council, White Paper, November 2011.

[19] M. Emms, B. Arief, L. Freitas, J. Hannon, and A. van Moorsel, "Harvesting high value foreign currency transactions from emv contactless credit cards without the pin," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 716–726.

[20] C. Cremers, K. Rasmussen, B. Schmidt, and S. Capkun, "Distance Hijacking Attacks on Distance Bounding Protocols," in *Security and Privacy (SP), 2012 IEEE Symposium on*, May 2012, pp. 113–127.

[21] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "Practical NFC Peer-to-peer Relay Attack Using Mobile Phones," in *Proceedings of the 6th International Conference on Radio Frequency Identification: Security and Privacy Issues*, ser. RFIDSec'10. Berlin, Heidelberg: Springer, 2010, pp. 35–49. [Online]. Available: http://dl.acm.org/citation.cfm?id=1926325.1926331

[22] R. Verdult and F. Kooman, "Practical Attacks on NFC Enabled Cell Phones," in *Near Field Communication (NFC), 2011 3rd International Workshop on*, Feb 2011, pp. 77–82.

[23] S. Drimer and S. J. Murdoch, "Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks." in *USENIX Security*, N. Provos, Ed. USENIX Association, 2007.

[24] D. Ma, N. Saxena, T. Xiang, and Y. Zhu, "Location-Aware and Safer Cards: Enhancing RFID Security and Privacy via Location Sensing," *Dependable and Secure Computing, IEEE Transactions on*, vol. 10, no. 2, pp. 57–69, March 2013.

[25] T. Halevi, D. Ma, N. Saxena, and T. Xiang, "Secure Proximity Detection for NFC Devices Based on Ambient Sensor Data," in *Computer Security – ESORICS 2012*, ser. LNCS, S. Foresti, M. Yung, and F. Martinelli, Eds. Springer, 2012, vol. 7459, pp. 379–396. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33167-1_22

[26] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, "Amigo: Proximity-Based Authentication of Mobile Devices," in *UbiComp 2007: Ubiquitous Computing*, ser. LNCS, J. Krumm, G. Abowd, A. Seneviratne, and T. Strang, Eds. Springer, 2007, vol. 4717, pp. 253–270. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74853-3_15

[27] P. Urien and S. Piramuthu, "Elliptic curve-based RFID/NFC authentication with temperature sensor input for relay attacks," *Decision Support Systems*, vol. 59, pp. 28 – 36, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923613002509

[28] M. Mehrnezhad, F. Hao, and S. F. Shahandashti, "Tap-Tap and Pay (TTP): Preventing Man-In-The-Middle Attacks in NFC Payment Using Mobile Sensors," in *2nd International Conference on Research in Security Standardisation (SSR'15)*, October 2014.

[29] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi, "Comparing and Fusing Different Sensor Modalities for Relay Attack Resistance in Zero-Interaction Authentication," in *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*. IEEE, 2014, pp. 163–171.

[30] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan, "Drone to the rescue: Relay-resilient authentication using ambient multi-sensing," in *Financial Cryptography and Data Security*. Springer, 2014, pp. 349–364.

[31] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, "Sound-proof: usable two-factor authentication based on ambient sound," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 483–498.

[32] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online]. Available: http://www.scipy.org/

[33] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Burlington, MA: Morgan Kaufmann, 2011.

[34] L. Breiman, "Random Forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[35] J. R. Quinlan, *C 4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA: Morgan Kaufmann, 1993.

[36] J. C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," in *ADVANCES IN KERNEL METHODS-SUPPORT VECTOR LEARNING*, 1998.

[37] M. Mehrnezhad, F. Hao, and S. F. Shahandashti, "Tap-Tap and Pay (TTP): Preventing Man-in-the-Middle Attacks in NFC Payment Using Mobile Sensors," Newcastle University, Department of Computing Science, Tech. Rep. CS-TR-1428, July 2014. [Online]. Available: http://www.cs.ncl.ac.uk/publications/trs/papers/1428.pdf