On the Effectiveness of Ambient Sensing for Detecting NFC Relay Attacks

Iakovos Gurulian*, Carlton Shepherd*, Eibe Frank†,
Konstantinos Markantonakis*, Raja Naeem Akram*, 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
Email: {Iakovos.Gurulian.2014, Carlton.Shepherd.2014, k.markantonakis, r.n.akram, keith.mayes}@rhul.ac.uk,
eibe@waikato.ac.nz

***Abstract*—Smartphones with Near-Field Communication (NFC) may emulate contactless smart cards, which has resulted in the deployment of various access control, transportation and payment services, such as Google Pay and Apple Pay. Like contactless cards, however, NFC-based smartphone transactions are susceptible to relay attacks, and ambient sensing has been suggested as a potential countermeasure. In this study, we empirically evaluate the suitability of ambient sensors as a proximity detection mechanism for smartphone-based transactions under EMV constraints. We underpin our study using sensing data collected from 17 sensors from an emulated relay attack test-bed to assess whether they can thwart such attacks effectively. Each sensor, where feasible, was used to record 350-400 legitimate and relay (illegitimate) contactless transactions at two different physical locations. Our analysis provides an empirical foundation upon which to determine the efficacy of ambient sensing for providing a strong anti-relay mechanism in security-sensitive applications. We demonstrate that no single, evaluated mobile ambient sensor is suitable for such critical applications under realistic deployment constraints.**

***Index Terms*—relay attacks; ambient sensing; mobile security; contactless transactions; near-field communication (NFC);**

I. Introduction

Near-Field Communication (NFC) [1] has opened smartphone platforms to a range of application domains, particularly those based previously on smart cards. Through card emulation, users may use their smartphones in a range of payment, transport and access control applications – leading to the deployment of services such as Google Pay and Apple Pay. The Android platform also provides Host-based Card Emulation (HCE) [2], which enables any application to take advantage of card emulation mode via NFC. Deloitte estimated that 5% of the 600-650 million NFC-enabled mobile phones were used at least once a month to make a contactless payment globally in 2015 [3]. In the same year, 12.7% of smartphone users in the USA were actively using contactless mobile payments according to Statista, while the value of such transactions is projected to grow to $114 billion (USD) by 2018[1]. Similar trends are being observed in other domains where mobiles are used to deliver smart card-type services, like transportation and access control [4].

Contactless transactions, however, are vulnerable to relay attacks – a passive man-in-the-middle attack in which an attacker extends the distance between a genuine payment terminal (point-of-service) and contactless smart card or NFC-enabled mobile device. This attack enables a malicious user to access services for which the genuine user is eligible, e.g. accessing a building with physical access controls and purchasing goods. Such attacks on contactless smart cards have been extensively studied in related literature [5]–[8], and were quickly shown to be applicable to NFC-enabled smart phones in [9]–[11]. In a range of past proposals, the sensors found in smartphones have been suggested as a strong countermeasure against relay attacks [12]–[17].

In this work, we present an empirical study that evaluates smartphone sensors as a relay attack detection mechanism under the practical time constraints stipulated by EMV. Legitimate data was collected from two devices that were in close proximity (<3cm) to each other, while two devices, placed 1.5m apart, assisted in the collection of illegitimate data. We then conducted a two-fold evaluation based, firstly, on similarity-based threshold analysis and, secondly, machine learning. The primary contributions of this paper are:

- Evaluation Test-bed: We established a reproducible[2] test-bed environment that was used to collect field data for legitimate and illegitimate transactions concurrently. Section IV describes the theoretical model and practical implementation of the test-bed.
- Anti-Relay Attack Analysis: The effectiveness of an ambient sensor in countering a relay attack was studied using the illegitimate transaction data. An effective sensor should be able to reject sensor values that were recorded on distant devices, while still accepting legitimate transactions. In our experiments, we selected a distance of 5ft (1.5m) between the devices to emulate a close-range relay attack.

II. Ambient Sensing and NFC Transactions

In this section, we briefly discuss NFC-based smartphone transactions, how relay attacks are conducted, and the deployment of ambient sensing as a countermeasure.

[1]Statista: http://www.statista.com/statistics/244475/proximity-mobile-payment-transaction-value-in-the-united-states/

[2]Source code and collected data available at https://github.com/AmbientSensorsEvaluation/Ambient-Sensors-Relay-Attack-Evaluation

A. Relay Attacks on Contactless Transactions

In NFC-based mobile contactless transactions, a mobile handset is brought into the radio range (<3cm) of a payment terminal and a dialogue is initiated. During this transaction, 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 [18]. This makes it difficult to ascertain whether the device is genuinely in close proximity to the terminal. (Note that the use of a PIN or biometric may not counter a relay attack effectively – see the Mafia fraud attack [19]).

Fig. 1. Overview of a Relay Attack

In a relay attack [9]–[11], as shown in Figure 1, an attacker presents 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 – relaying each message across this extended distance. The attacker has the potential to gain access to services using the victim's account if it successfully relays messages without detection.

B. Ambient Sensors in Conventional Transactions

A substantial portion of work surrounding relay-attack countermeasures for contactless smart cards relates to distance bounding protocols [5], [6], [19]–[22]. However, these may not be feasible for NFC-enabled phones – at the current state of the art – due to their requirement of high time-delay sensitivity and specialised hardware [1], [12]. As such, alternative methods have been proposed to provide proximity detection, most of which use environmental and motion sensors present on modern mobile handsets. In Section III, we discuss how ambient sensors have been proposed to counter relay attacks in NFC-based mobile contactless transactions.

An ambient sensor measures a particular physical/environmental property of its immediate surroundings, such as temperature, light or sound; modern smartphones and tablets are equipped with such sensors. The physical environment surrounding a smartphone or payment terminal can potentially provide a rich set of features that are reasonably unique to that location – the sound in a library, for example – which could be leveraged for proximity detection. We illustrate a generic approach for deploying ambient sensing as a proximity detection mechanism for mobile payments in Figure 2, with the following variations:

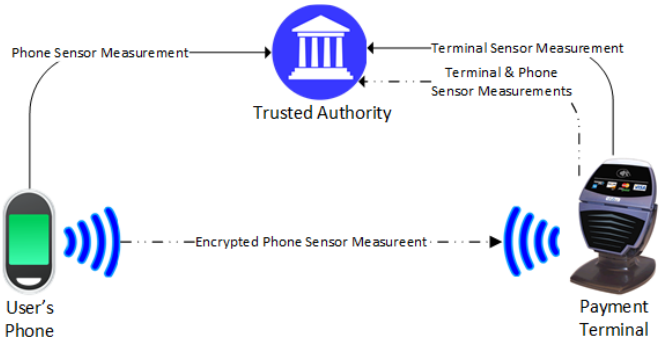

Fig. 2. Generic Deployment of Mobile Sensing for Proximity Detection

1) **Independent Reporting**. In this scenario, depicted as solid lines in Figure 2, both the smartphone and payment terminal collect sensor measurements independently of each other and transmit these to a trusted authority. This authority compares the sensor measurements, based on some predefined comparison algorithm with set margins of error (threshold), and decides whether the two devices are in proximity to one another.
2) **Payment Terminal Dependent Reporting**. This setup, shown as double-dot-dash lines in Figure 2, involves the smartphone encrypting the sensor measurements with a shared key (between smart phone and trusted authority) and transmitting the encrypted message to the payment terminal. The payment terminal sends its own measurements and the smartphone's to the trusted authority for comparison.
3) **Payment Terminal (Localised) Evaluation**. The smartphone transmits its measurement to the payment terminal, which compares it with its own measurements locally; the payment terminal then decides whether the smartphone is in close proximity.

Regardless of how a user interacts with the payment terminal, e.g. touching or tapping it with their device, the overall deployment architecture falls under one of the above scenarios. It can be observed that there is a potential fourth scenario in which a smartphone (payment instrument) could perform the comparison. In this study, our sole focus is on conventional transactions, requiring no specific interaction with the terminal, e.g. double-tapping, a gesture, or otherwise.

III. Related Work

We identify and summarise key pieces of related work that have addressed the proximity detection problem using ambient sensing as a strong countermeasure.

Drimer et al. [5] and Ma et al. [13] showed how location-related data, using a GPS (Global Positioning System), can be used to determine the proximity of two mobile phones. Ma et al. used a ten second window with location information collected every second, which was subsequently compared across various devices. The authors reported a high success rate in identifying whether the devices are in close proximity.

Halevi et al. [12] demonstrated the suitability of using ambient sound and light for proximity detection. Here, the authors analysed the sensor measurements – collected for 2 and 30 seconds duration for light and audio respectively – using a range of similarity comparison algorithms. Extensive experiments were performed in different physical locations, with a high success rate in detecting co-located devices.

Varshavsky et al. [17] 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 scenario of secure device pairing. In this work, they considered this approach to produce 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. [16] propose using ambient temperature with an elliptic curve-based RFID/NFC authentication protocol to determine whether two devices are co-located and to bootstrap a secure channel. The proposal combines the timing channels in RFID, traditionally used in distance bounding protocols, in conjunction with ambient temperature. Their proposal, however, was not implemented and has no experimental data to evaluate its effectiveness.

Mehrnezhad et al. [23] propose the use of an accelerometer to provide assurance that the mobile phone is within the vicinity 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 have provided a recording time range of 0.6–1.5 seconds.

Truong et al. [15] evaluated four different sensors. Similarly to previous studies, their sample rates were 10-120 seconds. Although results were positive, the sample duration made them unsuitable for NFC-based mobile transactions. Shrestha et al. [14] used specialised 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. This potentially renders the technique inapplicable to NFC-based mobile transactions.

Table I summarises past proposals, using sensor sampling durations to determine their suitability for NFC-based mobile phone transactions in banking and transportation. 'Unlikely' proposals are those whose sample duration is so large that they may not be adequate for mobile-based services that substitute contactless cards, while those with reasonably short sample ranges are labelled 'More Likely' in Table I. However, even schemes denoted as 'More Likely' may not be suitable for certain domains, such as banking or transport applications, where a strict upper bound is often present in which to complete the entire transaction. In these domains, the goal is to serve people as quickly as possible to maximise customer throughput, so time is critical in determining whether a transaction is successful and, indeed, permitted. An optimum transaction duration is 500ms rather than seconds.

TABLE I
RELATED WORK IN SENSOR-BASED ANTI-RELAY MECHANISMS

Paper	Sensor(s) Used	Sample Duration	Contactless Suitability
Ma et al. [13]	GPS	10 seconds	Unlikely
Halevi et al. [12]	Audio	30 seconds	Unlikely
	Light	2 seconds	More Likely
Varshavsky et al. [17]	WiFi (Radio Waves)	1 second	More Likely
Urien et al. [16]	Temperature	N/A	-
Mehrnezhad et al. [23]	Accelerometer	0.6 to 1.5 Seconds	More Likely
Thruong et al. [15]	GPS Raw Data	120 seconds	Unlikely
	WiFi	30 seconds	Unlikely
	Ambient Audio	10 seconds	Unlikely
	Bluetooth	12 seconds	Unlikely
Shrestha et al. [14]	Temperature (T)	Few seconds	Unlikely
	Precision Gas (G)	Few seconds	Unlikely
	Humidity (H)	Few seconds	Unlikely
	Altitude (A)	Few seconds	Unlikely
	HA	Few seconds	Unlikely
	HGA	Few seconds	Unlikely
	THGA	Few seconds	Unlikely

Shepherd et al. [24] questioned the effectiveness of ambient sensing as a proximity detection mechanism under short time frames ($<$ 1 second) – illustrating that a variety of sensors available via the Android platform perform poorly within an operating distance of $<$3cm. Both threshold- and machine learning-based analyses were employed using sensing data collected from mock transactions in the field. Similar results were also exhibited by Haken et al. [25] using sensors via the Apple iOS platform. While our study is similar, we focus on applying data collected from an emulated relay attack set-up under the assumption that $<$3cm provides little discrimination between measurements. A larger operating distance ($>$1.5m) may, however, provide greater evidence in identifying (il-)legitimate transactions.

Ambient sensing is also used in user-device authentication, key generation and establishment of secure channels [26]. These applications typically measure the environment for longer periods of time ($>$1 second) and, generally speaking, their primary goal is not proximity detection. As such, we do not discuss them in this section.

The use of ambient sensors for proximity detection in NFC-based mobile services is expanding, as illustrated by the number of proposals that currently exist. In this paper we extended the discussion to a large set of ambient sensors and included real-world relay attack data – not only analysing their effectiveness as proximity detection mechanism but also as an anti-relay mechanism. Table II shows that we have undertaken a comprehensive evaluation of ambient sensors for proximity detection and anti-relay effectiveness. A point to note is that in previous literature, ambient sensors were not evaluated for their effectiveness as an anti-relay mechanism (Table I). Without a strong ability to detect relay attacks, the proximity detection alone does not warrant their deployment as an effective mechanism in NFC based mobile transactions.

IV. TEST-BED FRAMEWORK: THEORETICAL MODEL

A test-bed environment was designed and developed that captures both legitimate and illegitimate pairs of sensor measurement under real-world conditions. A genuine pair of

measurements is from two devices that are physically in close proximity to each other (<3cm), while an illegitimate pair is from two devices that are not physically in close proximity, which we consider to be 5ft (1.5m) – emulating a reasonable distance for pickpocketing at a busy supermarket. To achieve this, we develop a test-bed of four devices, as per Figure 1, that records sensor measurements on both the legitimate terminal and device, and an emulated victim phone at distance (all three devices measure the ambient sensor values at approximately the same time). To avoid any discrepancies introduced because of the dependence of ambient sensors on location and time, we collected these pairs concurrently.

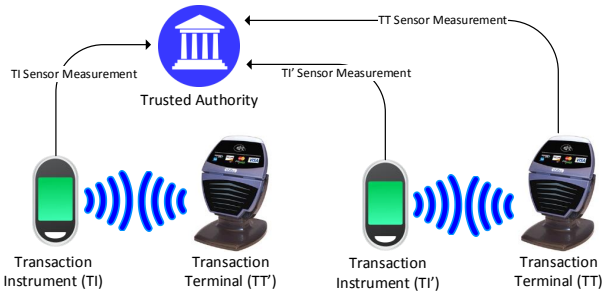

Fig. 3. Overview of Test-bed – Trial Data Collection Platform

Figure 3 shows our data collection setup. The Transaction Terminal (TT) is a static device, and we use this as a reference point for our two pairs. The Transaction Instrument (TI′) is a mobile phone in close proximity to the TT. The ambient sensor measurement pair TT–TI′ is referred to as the genuine pair. Another mobile phone, at a 5ft (1.5m) distance from the TT, is referred to as the Transaction Instrument (TI), and is co-located with the Transaction Terminal (TT′). The ambient sensor measurement pair TT–TI is referred to as the illegitimate pair. The rationale for setting up the test environment in this manner is to collect ambient sensor values from proximate and distant devices almost simultaneously. If, for a transaction 'T_i', the genuine pair is uniquely identified (and accepted) then the ambient sensor is considered effective. However, if the illegitimate pair is accepted (whether uniquely or along with the genuine pair) then the relay attack on that 'T_i' is successful. The reasoning is that if two devices at 5ft (1.5m) apart measure ambient sensors independently, and the illegitimate pair is indistinguishable from a genuine pair, then the attacker can successfully relay messages between these two devices without being detected.

At a point in time a user taps TI′ to TT; at approximately this point, TI is also tapped against TT′ and initiates the ambient sensor measurements. Thus, at approximately the same time, we have three separate ambient sensor measurements for the three devices (TT′ does not record any sensor values). Overall, for an ambient sensor to be effective, the Trusted Authority should be able to distinguish the genuine pair from the illegitimate pair. To evaluate each ambient sensor's effectiveness for proximity

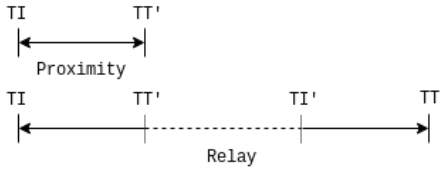

Fig. 4. Test-bed Scenarios

detection and to detect relay attacks, we analysed the collected data using threshold and machine learning based analyses.

A. Test-bed Architecture, Data Collection Implementation Platform, and Specifications

As mentioned in Section IV, four devices were used in the data collection phase, TT, TI′, TT′, and TI. During the experimental phase, the devices TT and TI′ were placed at a distance of 5ft from the devices TT′ and TI. When TI′ was brought in close proximity to TT, an NFC connection between the two devices would be established, initiated by TT, indicating the beginning of a transaction. According to the EMV standard, TT and TI′ should be in proximity, less than 3cm apart [18]. During the analysis process the pair TT–TI′ represented the genuine devices, where no relay attack was involved. The pair TT–TI represented the genuine devices, where a relay attack was active. So, device TI′ had a double role, acting as both a genuine, and a relay device. Figure 4 depicts the experimental setup.

Three Samsung Galaxy S4 (GT-I9500) Android devices running Android 5.0.1 were used in the experimental phase for data collection.This specific device was found to include a large variety of sensors, covering the majority of sensors supported by the Android platform [27], excluding the Geomagnetic Rotation Vector. A Nexus 5 Android device was used as TT′, which was not collecting sensor data. Table II lists the available sensors on the Samsung Galaxy S4 device, the ones used in our experiments and the rational behind excluding some. For the majority of excluded sensors, no values could be captured in the 500ms timeframe in our initial tests. Based on these initial tests, for less than 5% of the 500ms transactions any data is being returned by the Bluetooth, GPS, Network Location and WiFi sensors. Furthermore, during the initial evaluations of Ambient Temperature and Relative Humidity sensors, we discovered insufficient data was returned by the sensors during 500ms for any meaningful proximity analysis. The proximity sensor on Samsung Galaxy S4 returns only a true or false value when the sensor, located in the front of the device, is covered or uncovered, hence it could not be used effectively in the experimental phase. Lastly, the sound sensor, i.e. the microphone, of the specific device was tested; we found that this could not initiate and record values within 500ms. Finally, after our initial analysis we selected seven sensors and discarded the others.

An overview of the measurement recording process across the four devices is presented in Figure 5. Three Android applications were developed. Devices TI′ and TI were run-

TABLE II
SENSOR AVAILABILITY FOR SAMSUNG GALAXY S4

Sensor	Supported	Used	Reason
Accelerometer	✓	✓	-
Gravity	✓	✓	-
Gyroscope	✓	✓	-
Light	✓	✓	-
Linear Acceleration	✓	✓	-
Magnetic Field	✓	✓	-
Rotation Vector	✓	✓	-
Ambient Temperature	✓	✗	Insufficient values in timeframe.
Bluetooth	✓	✗	Insufficient values in timeframe.
GPS	✓	✗	Insufficient values in timeframe.
Relative Humidity	✓	✗	Insufficient values in timeframe.
Network Location	✓	✗	Insufficient values in timeframe.
Pressure	✓	✗	Insufficient for data collection.
Proximity	✓	✗	Insufficient values in timeframe.
Sound	✓	✗	Insufficient values in timeframe.
WiFi	✓	✗	Insufficient values in timeframe.
GRV†	✗	✗	Not present. Used Rotation Vector instead.

†Geomagnetic Rotation Vector

ning the same application, and Host-based Card Emulation (HCE) [28] was used to achieve NFC communication with TT and TT′. The application for device TT included two connection interfaces. For the first interface – used for the NFC connection with the device TI′ – device TT was set to NFC reader mode, allowing it to interact with discovered NFC tags. The second – used for connection with the device TT′ over WiFi – device TT would broadcast a UDP packet in the local network. In both, the message transmitted on the NFC or wireless channel included the sensor to be measured in that transaction, and a random 7-byte transaction ID. The transaction ID was generated by device TT and used in the analysis phase in order to uniquely identify each transaction across the devices. In real-world scenarios, device TI′ would act as the device communicating with TT′. However, since the scope of this paper is to evaluate the effectiveness of the ambient environment as an anti-relay mechanism, device TT was responsible for sending the information across the WiFi channel for greater transaction synchronisation. The final results were not influenced by this.

Lastly, the application running on device TT′ featured a broadcast listener for UDP packets from TT. Devices TT and TT′ were connected to the same wireless hotspot, created for the requirements of the experiment. Upon receiving a packet from device TT, device TT′ would be able to initiate a transaction with device TI, upon tapping the latter to the former. After the initiation of a transaction, devices TT, TI′, and TI would start recording data using some predefined sensor, for 500ms. On the Android operating system, data captured by a sensor is returned to an application in time intervals set by the application. The rate at which data was polled from the sensors was set at the highest available that was the same across all three devices.

Following the recording time period, devices TI′ and TI would send a response message, containing the transaction ID and sensor used to TT and TT′, respectively. Device TT would then validate the received data. In case of inconsistencies, data would not be stored for the specific transaction. The three devices would store the recorded data in a local SQLite database, along with the transaction ID, sequence number,

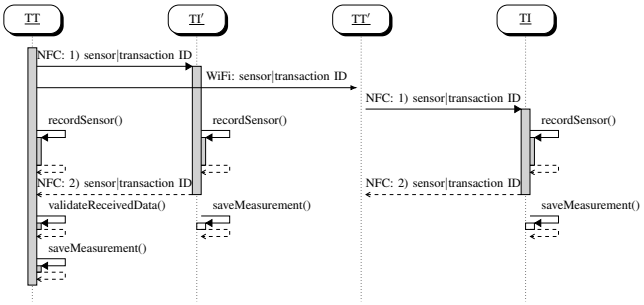

Fig. 5. Measurement Recording Overview

timestamp and the pre-defined location in which the recording took place. Separate database tables were used for each sensor. The recorded data was stored in XML format. During the data analysis phase, only transactions that existed in all three databases, based on their transaction ID, were considered. A total of 400 transactions were collected for each sensor, distributed in 2 distinct locations on the university campus.

V. TRANSACTION DATA ANALYSIS

In line with the framework discussed previously, we conducted a two-fold evaluation using the sensor data collected during field trials.

A. Evaluation 1: Threshold-based Analysis

We employed similarity metrics from related work to compute the similarity of the legitimate and illegitimate transaction pairs. This analysis – used prevalently in past work and other binary classification tasks, like biometrics – assumes some threshold, t, is able to separate both illegitimate and legitimate attempts. The computed similarities provide a range of thresholds to test for each metric, before calculating the Equal Error Rate (EER) for each sensor – defined as the intercept of the False Acceptance Rate (FAR) and False Rejection Rate (FRR). This threshold would subsequently be used by decision authorities, e.g. the Trusted Authority in Figure 3, to determine whether the transaction devices are proximate. This way, the success/failure rate of a relay attack being conducted successfully can be determined in the presence of ambient sensing.

B. Evaluation 2: Machine Learning

Simple distance functions give equal weight to each measurement obtained from the sensors. To address this, we consider machine learning to perform the discrimination between legitimate and illegitimate transactions more effectively through modelling non-linear interactions between measurements. The experiments we conduct apply the same evaluation strategy as the first method, but the threshold is based on the probability estimate output by the learned classification model, i.e. the estimated probability that a transaction is legitimate. Moreover, to avoid optimistic bias in the error estimate when applying machine learning, it is necessary to perform a train-test experiment. More specifically, the full set of transaction pairs

is split into a training set and a test set. The machine learning algorithm is applied to the training set to build a classification model that can output class probability estimates. Once the model has been been built, it is applied to obtain probability estimates for the test set. When we split the data into training and test sets, we ensure that two transactions with the same ID (i.e., a legitimate and a fraudulent transaction that were recorded simultaneously) are either both in the training set or both in the test set, to avoid potential bias. Moreover, instead of using a single train-test split, we use 10-fold cross-validation repeated 10 times, a standard estimation technique from machine learning that generates 100 different train-test splits based on shuffled versions of the data. The learning algorithm is run 100 times on the 100 training sets, to build 100 models, and these 100 models are evaluated on the corresponding test sets. Performance estimates from the 100 test sets are averaged to obtain a final performance estimate.

C. Data Analysis Workflow

In this section, we describe the details of both evaluations in further detail, including any pre-processing, the evaluation metrics, and the results.

1) Sensor Measurement Similarity: We employed two similarity metrics used in related work to determine whether the measurements of TI'_i and TI_i were in proximity with TT_i, i.e. between (TI'_i,TT_i) and (TI_i,TT_i). Specifically, we used the Mean Absolute Error (MAE) and Pearson's Correlation Coefficient, as used in [23] to evaluate proximity detection mechanism proposed for NFC based mobile transactions, which are given in Eqs. 1 and 2 respectively.

$$MAE(A_i, B_i) = \frac{1}{N}\sum_{j=1}^{N}|A_{i,j} - B_{i,j}| \tag{1}$$

Where N refers to the number of datapoints in a sensor measurement, and $A_{i,j}$ refers to the j^{th} datapoint of the i^{th} sensor measurement on device A.

$$corr(A_i, B_i) = \frac{cov(A_i, B_i)}{\sigma_{A_i} \cdot \sigma_{B_i}} \tag{2}$$

Where σ_{A_i} denotes the standard deviation of the sensor measurements of A_i, and cov represents the covariance, given below in Eq. 3, with μ_{A_i} denoting the mean of A_i.

$$cov(A_i, B_i) = \frac{1}{N}\sum_{j=1}^{N}(A_{i,j} - \mu_{A_i})(B_{i,j} - \mu_{B_i}) \tag{3}$$

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

2) Pre-processing: All sensors except those for light produce a vector of values consisting of x, y and z components. For these sensors, 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. In the event that the sensor values exceeded the maximum permitted transaction time (500ms), such values were discarded prior to computing the similarity. Moreover, any sensor values on device A that were recorded after the maximum time recorded by device B were also discarded. This was to prevent undefined results when computing MAE; calculating $|A_{i,j} - B_{i,j}|$ is undefined when $A_{i,j}$ has no corresponding datapoint on device B, i.e. $B_{i,j}$.

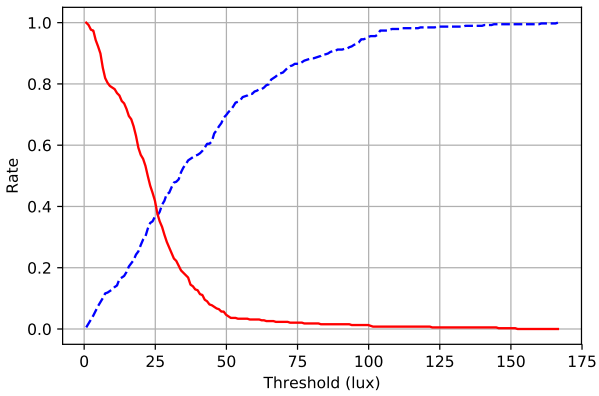

Fig. 6. Light sensor FAR-FRR curves with MAE.

D. Calculating FAR, FRR and EER

We computed Equal Error Rates (EERs) to determine optimal similarities/thresholds where the rate of false acceptances (FAR) is equal to the rate of false rejections (FRR) for each tested sensor. The following notions of true accepts (TAs) and rejects (TRs), false accepts (FAs) and rejects (FRs) are defined for each evaluation:

Evaluation 1. TA: the legitimate pair, transaction instrument-terminal (TI'_i,TT_i) is accepted correctly. TR: the distant instrument-terminal pair, (TI_i,TT_i), is rejected correctly. FA: the distant-terminal pair is wrongly accepted as a legitimate transaction. FR: the legitimate transaction instrument-terminal pair is rejected incorrectly.

Evaluation 2. For the machine learning experiments, we use the same definitions as in Evaluation 1. As already indicated above, the estimated probability of being legitimate is used instead of a similarity score when the threshold for EERs is determined. When training and testing each machine learning model, we use the individual differences $|A_{i,j} - B_{i,j}|$ as attributes (also called features or independent variables) that describe each pair of transactions. Each example for training and testing the machine learning model thus has 49 numeric features (corresponding to 490ms sampled at 10ms intervals). An example is labelled as positive if it corresponds to a legitimate transaction and as negative otherwise.

To determine an optimal similarity threshold that broadly balances the security and usability of each sensor, 250 thresholds – between the minimum and maximum observed distances for MAE and $[-1, 1]$ for correlation – were tested for each similarity metric using EERs. The FPR and FNR were measured at each threshold using Eqs. 5 and 6. Here, a threshold is the maximum permitted difference in similarity

before a transaction is rejected; conversely, a transaction is accepted if the similarity between sensor measurements is within this threshold. Ideally, a chosen threshold should reject all illegitimate transactions, namely those between the distant instrument and the terminal, while accepting all legitimate transactions between the transaction instrument and terminal. The EERs and associated thresholds for each sensor are calculated for both of the similarity metrics described in Section V-C. Figure 6 shows the FAR/FRR graph produced for 100 different thresholds using the Gyroscope with Pearson's Correlation Coefficient. By inspection, the point at which FRR and FAR intersect (FAR=FRR) has a corresponding EER of approximately 42% for this sensor and metric.

$$TAR = \frac{TA}{TA + FR} \qquad TRR = \frac{TR}{TR + FA} \tag{5}$$

$$FAR = 1 - TRR \qquad FRR = 1 - TAR \tag{6}$$

E. Individual Sensor Results

The rate of accepted transactions using the distant instrument – analogous to the rate of successful relay attacks – is estimated by inspecting the EER. The FAR is equivalent to the proportion of successful distant transactions (FAs) to the total of FAs and correctly denied distant transactions (TRs). The rate of potentially successful relay attacks can be estimated from the EERs, which we present in Table III. We estimate, for example, that using the accelerometer with MAE will result in 49.4% relayed transactions being accepted using a threshold of $0.913 ms^{-2}$.

TABLE III
THRESHOLDS AND EERS FOR THRESHOLD-BASED ANALYSIS

Sensor (units)	t_{MAE}	EER_{MAE}	t_{corr}	EER_{corr}
Accelerometer (ms^{-2})	0.913	0.494	0.526	0.480
Gyroscope ($rads^{-1}$)	0.329	0.521	0.336	0.455
Magnetic Field (μT)	153.9	0.444	0.399	0.473
Rotation Vector (N/A)	0.493	0.330	0.470	0.472
Gravity (ms^{-2})	0.290	0.521	0.586	0.490
Light (lux)	26.54	0.367	0.714	0.488
Linear Accel. (ms^{-2})	0.569	0.482	0.064	0.536

F. Machine Learning Analysis

Table IV shows the results obtained using five different machine learning algorithm that induce different types of classification models from data. For each dataset/sensor and learning algorithm, the table shows the mean and standard deviation of the 100 equal error rate estimates obtained using 10-fold cross-validation repeated 10 times. The best result for each sensor is shown in bold.

A random forest [29] is an ensemble classifier comprising a large number of decision trees induced in a semi-random manner from bootstrap samples of the original dataset. In our experiments, we used the default parameters for the *RandomForest* classifier in the Weka software [30]. It generates an ensemble of 100 decision trees from the training data. We also evaluated (a) a simple naive Bayes classifier, which assumes conditional independence of the attributes given the classification, and models the data for each class using a Gaussian distribution with a diagonal covariance matrix, (b) logistic regression, which assumes that the log-odds of the class probabilities are linearly related to the attributes, (c) decision trees grown using the C4.5 decision tree learning algorithm [31], and (d) support vector machines optimised with the SMO algorithm [32]. For the support vector machines, the complexity parameter C and the width of the RBF kernel γ were tuned using a grid search by optimising AUROC as estimated using internal cross-validation on the training data.

Table IV shows that the random forest method produces the lowest equal error rate for most sensors but classification is far from perfect. The lowest equal error rate across sensors, 17.9%, is achieved for the gyroscope sensor. On the data from the light sensor, the nominal result for a single decision tree is better than that for a random forest but the estimated equal error rate for the decision tree learner exhibits high standard deviation. When performing a corrected resampled paired t-test, the observed difference is not statistically significant at the 5% significance level. On the other hand, random forests perform statistically significantly better than all other learning algorithms on the accelerometer data, the gyroscope data, and the linear acceleration data.

Although accuracy is far from perfect, the results from the machine learning experiments indicate that all sensors apart from the gravity sensor provide useful information for the discrimination between legitimate and distant transactions. Ranking the sensors based on discriminative power (i.e., equal error rate) when evaluated in conjunction with random forest classifiers yields the following ranking (best to worst): gyroscope, accelerometer, rotation vector, linear acceleration, light, magnetic field, and gravity. Interestingly, very few measurements from the gyroscope sensor are required to achieve the observed classification performance for random forests. Using just the first 10 measurements (obtained at 10ms, 20ms, ..., 100ms), the random forest classifier achieves an error rate of 17.8% already. The best observed equal error rate is obtained when using all measurements in the range 10ms-200ms: 16.6%. However, this improvement on the result obtained on the full set of measurements is not statistically significant. Note that, excluding the gravity sensor, which never produces useful results, the gyroscope is the only sensor for which the number of measurements can be reduced without affecting accuracy. All other sensors yield an immediate drop in accuracy when the number of measurements is reduced.

G. Analysis Equipment

The first analysis was conducted on a Fedora machine with a quad-core Intel i5 4690k (3.7 GHz) and 16GB of RAM. The analysis application was developed in Python, using the Pandas [33] and NumPy [34] libraries for data loading and numerical computation. The Weka machine learning software, implemented in Java, was used to run the machine learning experiments on a cluster of 10 Ubuntu Linux computers with Intel Core i7-2600 CPUs and 16 GB of RAM. Multi-threading

TABLE IV
ESTIMATED EER FOR MACHINE LEARNING ALGORITHMS, OBTAINED BY REPEATING 10-FOLD CROSS-VALIDATION 10 TIMES

Sensor	Random Forest	Naive Bayes	Decision Tree	Logistic Regression	Support Vector Machine
Accelerometer	**0.277**±0.052	0.474±0.047	0.358±0.059	0.483±0.050	0.454±0.126
Gyroscope	**0.179**±0.041	0.354±0.059	0.228±0.049	0.356±0.055	0.288±0.045
Magnetic Field	**0.361**±0.055	0.400±0.053	0.389±0.063	0.421±0.061	0.385±0.053
Rotation Vector	**0.285**±0.052	0.327±0.055	0.317±0.073	0.353±0.050	0.325±0.050
Gravity	0.499±0.046	0.488±0.043	0.494±0.057	**0.484**±0.043	0.486±0.156
Light	0.361±0.059	0.369±0.058	**0.293**±0.149	0.407±0.054	0.351±0.054
Linear Acceleration	**0.307**±0.050	0.484±0.048	0.392±0.057	0.502±0.049	0.397±0.058

was used for training random forests and performing parameter optimisation for support vector machines. Generating all performance estimates using 10-times 10-fold cross-validation took less than two hours. The vast majority of this time was used to obtain performance estimates for the support vector machines.

VI. OUTCOME AND FUTURE DIRECTIONS

The presented results provide a basis for quantifying the effectiveness of various mobile sensors as an anti-relay mechanism in NFC-based mobile transactions. In both evaluations, investigated the success rate of a potential relay attack by considering readings between the terminal–distant instrument and the terminal–transaction instrument. The results of the first evaluation are shown in Table III. The Rotation Vector sensor performs relatively better with an EER of 0.330, while the Accelerometer has a 49.4% possibility of a success – one of the highest in our analysis. Not only does the EER imply the potential success of the attack, but also the potential of a legitimate transaction being denied. The Rotation Vector EER indicates that 33% of relay attacks would be accepted *and* 33% legitimate transactions would be denied. Denying 1-in-3 legitimate transactions would invariably cause annoyance issues for users in practice. From this analysis, it is difficult to recommend any of the sensors for a single-sensor deployment for high security applications, such as banking. Such sensors 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.

The next evaluation applied machine learning classifiers (see Table IV). The Gyroscope sensor with Random Forest performs significantly better, with an estimated EER of 0.179; in this analysis, the gyroscope was the best sensor, but all sensors apart from the gravity sensor provided some discriminative power. This illustrates that ambient sensors may have potential as an anti-relay mechanism, but the detection accuracy is not high enough to provide sufficient security in a real-world deployment.

One reason that past work achieved different results could be due to the significantly larger sampling durations (see Section III). The sampling duration imposed in our experiments was in line with the performance requirements of an EMV application, i.e. 500 milliseconds. Transportation is another major application for contactless smart cards; in this domain, the recommended transaction duration is far lower, between 300-500 milliseconds. The 500 millisecond limit in our experiment was thus an upper bound of the recommendations of two significant application areas where contactless mobile phones may be utilised.

VII. CONCLUSION

This investigation aimed to analyse and evaluate a range of sensors present in modern off-the-shelf mobile devices, and to determine which sensors, if any, would be suitable as a anti-relay mechanism for NFC-based smartphone transactions. We shortlisted 17 sensors accessible through the Google Android platform, before limiting the investigation to those which are widely-available and displayed promise in our initial trails. In existing literature, only 12 sensors have been suggested as an effective proximity detection mechanisms, as listed in Table I. Some sensors are only available in specialised ambient sensor hardware and, in almost all instances, no relay attack data was collected to determine their effectiveness against such attacks. In this study, we implemented and evaluated 7 sensors and collected data representing a genuine transaction and a malicious transaction (from actual relay attacks). The objective of collecting these two separate sets of data at the same time was to empirical evaluate the implemented sensors with a high degree of objectivity. As part of our ongoing research, we are experimenting with simultaneously measuring multiple sensors within the transaction time duration. This is to evaluate whether this would reduce the risk associated with these sensors individually.

ACKNOWLEDGEMENT

Carlton Shepherd is supported by the EPSRC and the British government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/K035584/1). The authors would also like to thank anonymous reviewers for their valuable comments.

REFERENCES

[1] V. Coskun, B. Ozdenizci, and K. Ok, "A Survey on Near Field Communication (NFC) Technology," *Wireless Personal Communications*, vol. 71, no. 3, pp. 2259–2294, 2013.

[2] A. Umar, K. Mayes, and K. Markantonakis, "Performance Variation in Host-Based Card Emulation Compared to a Hardware Security Element," in *Mobile and Secure Services (MOBISECSERV), 2015 First Conference on*. IEEE, 2015, pp. 1–6.

[3] "Contactless Mobile Payments (finally) Gain Momentum," Deloitte, Online Report, 2015. [Online]. Available: http://goo.gl/YxyzTG

[4] "A Cashless Future on the Horizon," VeriFone, White Paper, 2010. [Online]. Available: http://goo.gl/1jPYQ5

[5] 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.

[6] A. Francillon, B. Danev, and S. Capkun, "Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars," in *Network & Distributed System Security*, ser. NDSS. The Internet Society, Feb. 2011.

[7] 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.

[8] Z. Kfir and A. Wool, "Picking Virtual Pockets using Relay Attacks on Contactless Smartcard Systems," in *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*. IEEE, 2005, pp. 47–58.

[9] 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-Verlag, 2010, pp. 35–49.

[10] L. Francis, G. P. Hancke, K. Mayes, and K. Markantonakis, "Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones." *IACR Cryptology ePrint Archive*, vol. 2011, p. 618, 2011.

[11] 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.

[12] 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. Lecture Notes in Computer Science, S. Foresti, M. Yung, and F. Martinelli, Eds. Springer Berlin Heidelberg, 2012, vol. 7459, pp. 379–396. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33167-1_22

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, "Amigo: Proximity-Based Authentication of Mobile Devices," in *UbiComp 2007: Ubiquitous Computing*, ser. Lecture Notes in Computer Science, J. Krumm, G. Abowd, A. Seneviratne, and T. Strang, Eds. Springer Berlin Heidelberg, 2007, vol. 4717, pp. 253–270.

[18] "EMV Contactless Specifications for Payment Systems: Book D - EMV Contactless Communication Protocol Specification," EMVCo, LLC, Specification Version 2.6, March 2016.

[19] 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.

[20] K. B. Rasmussen and S. Capkun, "Realization of RF Distance Bounding." in *USENIX Security Symposium*, 2010, pp. 389–402.

[21] G. P. Hancke and M. G. Kuhn, "Attacks on Time-of-flight Distance Bounding Channels," in *Proceedings of the First ACM Conference on Wireless Network Security*, ser. WiSec '08. New York, NY, USA: ACM, 2008, pp. 194–202.

[22] I. Boureanu, A. Mitrokotsa, and S. Vaudenay, "Towards Secure Distance Bounding," in *Fast Software Encryption*. Springer, 2014, pp. 55–67.

[23] 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.

[24] C. Shepherd, I. Gurulian, E. Frank, K. Markantonakis, R. Akram, K. Mayes, and E. Panaousis, "The Applicability of Ambient Sensors as Proximity Evidence for NFC Transactions," in *Mobile Security Technologies, IEEE Security and Privacy Workshops*, ser. MoST '17. IEEE, May 2017.

[25] G. Haken, K. Markantonakis, I. Gurulian, C. Shepherd, and R. N. Akram, "Evaluation of Apple iDevice Sensors As a Potential Relay Attack Countermeasure for Apple Pay," in *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*, ser. CPSS '17. New York, NY, USA: ACM, 2017, pp. 21–32.

[26] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, "Sound-Proof: Usable Two-Factor Authentication Based on Ambient Sound," in *24th USENIX Security Symposium*. Washington, D.C.: USENIX Association, Aug. 2015.

[27] "Android API Reference Documentation: Sensors Overview," http://developer.android.com/guide/topics/sensors/sensors_overview.html, [accessed 27-April-2016].

[28] "Android API Reference Documentation: Host-based Card Emulation," http://developer.android.com/guide/topics/sensors/sensors_overview.html, [accessed 27-April-2016].

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

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

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

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

[33] W. McKinney, " Data Structures for Statistical Computing in Python ," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 51 – 56.

[34] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," *Computing in Science Engineering*, vol. 13, no. 2, pp. 22–30, March 2011.