Good Vibrations: Artificial Ambience-Based Relay Attack Detection

Iakovos Gurulian*, Konstantinos Markantonakis*, Eibe Frank†, and Raja Naeem Akram*
*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, k.markantonakis, r.n.akram}@rhul.ac.uk, eibe@waikato.ac.nz

***Abstract*—Relay attacks are passive man in the middle attacks, aiming to extend the physical distance of devices involved in a transaction beyond their operating environment, within the restricted time-frame. In the field of smartphones, proposals have been put forward suggesting sensing the natural ambient environment as an effective Proximity and Relay Attack Detection (PRAD) mechanism. However, these proposals are not in compliance with industry imposed constraints (e.g. EMV and ITSO) mandating that transactions should complete within a certain time-frame (e.g. 500ms for EMV contactless transactions). The generation of an artificial ambient environment (AAE) using peripherals of the transaction devices has shown positive results when using infrared light as an AAE actuator. In this paper we propose the use of vibration as an alternative AAE actuator. We empirically evaluated the effectiveness of the proposed solution as a PRAD mechanism on an experimental test-bed that we deployed. A total of 36,000 genuine and relay attack transaction pairs were analysed using well-known machine learning algorithms. The results of our analysis indicate that the proposed solution is highly effective.**

***Index Terms*—Mobile Payments, Mobile Ticketing, Relay Attacks, Ambient Environment Sensing, Contactless, Experimental Analysis.**

I. Introduction

Smart cards [1]–[5] and smartphones [6]–[9] are susceptible to relay attacks [10]. Using a relay attack, an attacker may get unauthorised access to services and facilities, like payments and access to buildings. Proposals for countering relay attacks in the field of smart cards suggest the use of distance bounding protocols [11], [12]. In the field of smartphones, distance bounding protocols may not be applicable due to unpredictable behaviour related to their multi-process nature and the multitude of hardware components [13]. Sensing of the natural ambient environment has been proposed as a potential alternative against the off-the-shelf attacker [14]–[19]. However, existing work does not take into account industry specific restrictions. For example, EMV contactless transactions have to complete within 500ms [20]–[22], and transport ticketing related transactions typically require between 300 and 500ms [23].

Limited effectiveness of natural ambient sensing as a Proximity and Relay Attack Detection (PRAD) mechanism in transactions of up to 500ms has been demonstrated through empirical evaluation [24], [25]. The generation and measurement of an artificial ambient environment (AAE) based on random bits or sequences by the peripherals of the transaction devices has been proposed as an alternative. Evaluation of infrared light as an AAE actuator has demonstrated positive results [26].

In this paper we investigate the effectiveness of vibration as an AAE actuator. One or both transaction devices vibrate a randomly generated vibration pattern. The impact of the vibration is recorded by both transaction devices through selected ambient sensors. The recorded data from the two devices is subsequently compared in order to establish proximity evidence.

In order to assess the effectiveness of the proposed solution, we deployed an evaluation test-bed. Based on trials, 36,000 genuine and relay transaction pairs, covering a wide range of potential techniques that an attacker might use against the proposed system were used. Analysis with popular machine learning algorithms indicates that the proposed solution is highly effective as a PRAD mechanism, especially in the case of applying a gyroscope as the ambient sensor (0.1% Equal Error Rate).

The main contribution of this paper are:

- Vibration as an AAE actuator (Section V): We proposed the sensing of short (up to 500ms), random vibration sequences, generated by the transaction devices, as a means of PRAD.
- Evaluation of the proposed solution: We deployed an evaluation test-bed (Section VI) in order to assess the effectiveness of the proposed solution. We analysed 36,000 transactions, using different vibration modalities, ambient sensors, and potential attack techniques. The dataset was evaluated using well-known machine learning algorithms (Section VII). The results indicate high effectiveness of the proposed solution, and resilience against relay attacks, when using certain sensors (namely, accelerometer, gyroscope, linear acceleration, magnetic field, and rotation vector).

This paper is organised as follows:

- In Section II, an overview of relay attacks is given.
- Section III lists related works in the field.
- The use of AAEs as PRAD methods is described in Section IV.
- The theoretical framework of using vibration as an AAE actuator is described in Section V.
- In Section VI, the details of the evaluation framework for the proposed solution are described.
- The experimental results are provided in Section VII.

- Discussion of the results is performed in Section VIII.
- Finally, concluding remarks are made in Section IX.

II. Relay Attacks

A variety of applications is affected by relay attacks, like Near Field Communication (NFC) based contactless transactions. During a relay attack, the goal of the attacker is to relay communication messages between two devices that are located beyond their designated operational environment, without being detected.

The relay of the communication messages is performed using some relay equipment that the attacker possesses. For example, for the case of an NFC contactless payment scenario (Figure 1), an attacker can present to a genuine user a masqueraded (malicious) payment terminal. At a distant location, the attacker should present a masqueraded (malicious) payment instrument to a genuine payment terminal. When the user attempts to perform a transaction, the communication messages of the payment transaction will be relayed between the attacker's relay equipment. If the attack is successful, an unauthorised transaction between two genuine parties has been performed.

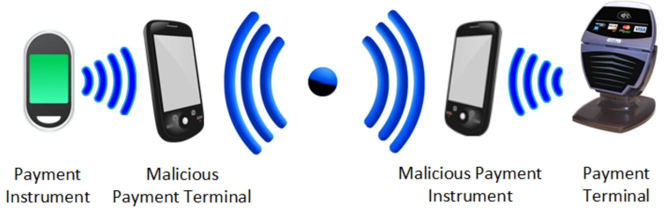


Fig. 1. Overview of a Relay Attack

Relay attacks against mobile devices have been demonstrated [4], [8], [9], [27]. In order to detect the existence of a relay attack, evidence regarding the co-presence of the genuine transaction devices should be established. As already mentioned, in the field of smartphones, establishment of proximity evidence by the assistance of ambient sensing has demonstrated positive results. This technique requires the two transaction devices to capture environmental data, using some ambient sensor, for some predefined period of time.

An alternative approach, by generating an artificial ambient environment (AAE) using the peripherals of the communicating devices has also demonstrated positive results. Using infrared light as an AAE actuator has demonstrated high effectiveness as a PRAD mechanism in transactions with industry imposed time limits of up to 500ms, like EMV contactless payments and transport ticketing.

With both techniques, the data from the two devices is then compared for similarity, based on which a decision is made regarding their co-presence. The comparison process can be performed either by one of the communicating devices, or by a trusted third party (TTP).

III. Related Work

Ma et al. [15] proposed the use of GPS (Global Positioning System) as a means of co-location detection. A time frame of 10 seconds was used for data collection, and values were recorded every second. High success rate was reported by the authors for proximity detection.

Halevi et al. [14] proposed the use of ambient light and sound. Values were captured for 30 and two seconds, respectively. The authors used various comparison algorithms, and high success rate was reported.

Varshavsky et al. [19] compared the WiFi networks, along with the signal strengths, that the devices were able to detect. The main objective of this work was device pairing, and positive results were reported.

Urien et al. [18] combined ambient temperature and an elliptic-curve based RFID and/or NFC authentication protocol. No performance results were presented by the authors, as there was no practical implementation.

Mehrnezhad et al. [28] recorded values using the accelerometer of the devices involved in a payment transaction in order to detect device co-location. A double tap was required in their proposal. According to the authors, the transaction time lasted between 0.6 and 1.5 seconds, and a high success rate was observed.

Truong et al. [17] assessed a variety of sensors for proximity detection. The recording time frame was between 10 and 120 seconds, and positive results were reported.

Shrestha et al. [16] used a Sensordrone and recorded multiple sensors. The precise sample duration is not provided in this work, however the authors state that recordings lasted for a few seconds.

In [24] and [25], the effectiveness of recording the natural ambient environment in short transactions (up to $500ms$) was empirically evaluated, with results different from those in the existing literature. Comparison algorithms used in previous works, as well as machine learning techniques, produced a large number of false negative results.

Further work on using the ambient environment for device co-location has been performed in the field of two-factor authentication (2FA). Karapanos et al. [29] proposed using sound as a means of proximity detection, for 2FA. The aim of the authors was to provide a more usable 2FA method than the existing ones, in order to make 2FA more widely accepted.

In previous work [26], we proposed a PRAD framework by using artificial ambient environments (AAE). Infrared light was evaluated as an AAE actuator. The challenge for the attacker in this case would be to accurately relay a 100ms long random bit sequence in the form of infrared pulses and pauses, generated by the transaction instrument. Failure to timely and accurately relay the sequence would lead to a detection of the attack, as the transaction terminal stops listening for infrared signals 100ms after the transaction initiation. Relay attacks were successfully detected, while the false rejection rate was approximately 2%.

IV. Artificial Ambient Environment-based Relay Attack Detection

In order to increase the irreproducibility and uniqueness of the ambient environment, the transaction devices can generate an artificial environment using a peripheral, measurable by

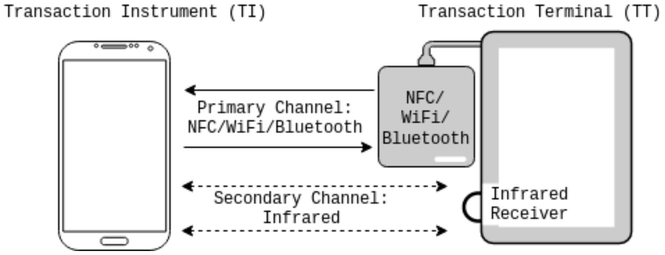


Fig. 2. Framework Architecture

an ambient sensor(s). The artificial environment should be based on randomly generated bits or sequences to act as a second (out-of-band) channel (Figure 2) for assuring proximity between the transaction devices.

Upon initiation of a transaction, one (unidirectional) or both (bidirectional) the devices should be responsible for the generation and/or sensing of the AAE for some predefined time. Upon completion of the sensor measurement, a comparison of the captured data of one device against the captured data of the other device, or the generated data used to construct the AAE, should take place. The comparison can either be performed by one of the communicating parties, or by a trusted third party.

Only data captured while the Artificial Ambient Channel (AAC) is active should be considered during the comparison, and any data captured outside the time-frame should be discarded. This way, an attacker cannot capture the generated sequence and then reproduce it at a remote location, as delayed streams (outside the AAC time frame) are disregarded. For an effective AAE, the attacker should also not be able to accurately relay the information from the second channel, such that the comparing party cannot distinguish between a legitimate and an illegitimate transaction with a high degree of confidence.

So, the basic principles of an AAE should be:

1) The AAE generation should be based on random bits/sequences.
2) The AAE should provide sufficient evidence in order for two genuine devices to establish proximity assurance.
3) The AAE should be hard for an attacker to accurately reproduce at a remote location.

The main goal of the AAEs is to protect against off-the-shelf attackers. By the term *'off-the-shelf attacker'* we refer to an attacker with access to off-the-shelf, state of the art equipment, without capabilities to build customised hardware or channels with the purpose to defeat the proposed scheme. A resourceful attacker with access to state of the art equipment might be capable of effectively and timely reproducing the same conditions at a remote location. However, smartphones suffer from multiple security issues [30], and therefore using such devices for security critical applications should be avoided, so resourceful attackers are not the main concern.

Peripherals widely available on modern smartphones that could potentially act as AAE actuators include: 1) the infrared emitter, 2) the speaker, 3) the flash light, 4) the device's vibration, 5) the device's display, 6) the WiFi, 7) the Bluetooth, and 8) the camera.

V. Vibration as an AAE Actuator

In this section we present the basic principles of using vibration as an AAE actuator.

A. AAE Framework

The genuine user is asked to place the Transaction Instrument (TI) on the Transaction Terminal (TT). Upon initiation of the transaction, one (unidirectional) or both (bidirectional) transaction devices vibrate using some randomly generated pattern. Simultaneously, both devices use some ambient sensor to measure the impact of the vibration over a period of time. In this work, we used a 500ms vibration pattern and recording. In the case that both devices vibrate, a fusion of the two random vibrations is causing the impact. Candidate sensors, available on a wide range of modern smartphones, that can potentially be used to measure the impact of the vibration include: 1) the accelerometer, 2) the geomagnetic rotation vector, 3) the gyroscope, 4) the magnetic field, 5) the rotation vector, 6) the gravity sensor, and 7) the linear acceleration sensor.

The vibration-based AAE channel (referred to as the 'vibration channel') is used as an out-of-band channel[1], along with the main communication channel (e.g. NFC, WiFi), aiming to provide proximity assurances. The initiation of the vibration channel should be subsequent to the initiation of the main communication channel for each of the devices, with as minimal delays as possible, in order to minimise the attack window.

Upon completion of the transaction, data streams captured by the sensors of the two devices are compared for similarity. The similarity comparison can be performed either by one of the transaction devices, or by a Trusted Third Party (TTP). The exact characteristics and architecture of the party responsible for the comparison (i.e. one of the transaction devices or the TTP) are beyond the scope of this paper. Of course, the recoded data streams should be communicated to the comparison party in an encrypted and authenticated form.

B. Threat Model

In this paper, the attacker is of opportunistic nature and requires no prior interaction or knowledge of either TT or TI. Scenarios where TT or TI are compromised will not be covered in this paper because a relay attack is unlikely to be required to achieve the attacker's goals when this is the case. The focus of this paper is on proximity assurance for genuine devices.

We assume that the attacker only has access to off-the-shelf relay equipment. Usually transaction limits apply on transactions using smartphones, for example a £30 limit on digital payment transactions in the UK [31]. Therefore, due to the limited finanical gain involved, a powerful attacker with advanced and expensive relay equipment is not our major concern.

[1] Out-of-band channel: Second channel

For the same reason, we assume that the attacker can only try to guess the pattern vibrated by the genuine devices. More advanced attacks, like acoustic attacks, during which an attacker captures the vibration pattern by performing acoustic analysis and replays it at a distant location, are not considered. Even though acoustic attacks have been demonstrated in the context of decoding a vibration pattern [32], combining an acoustic and a relay attack by relaying the decoded pattern at a remote location in real time is a challenging task for off-the-shelf attackers. Delays associated with audio recording latency, signal processing, data communication between the two relay devices, and replaying at a distant location, which would be required for such an attack, would significantly increase its complexity. Note that delay in relaying the pattern at a distant location would lead to a shift of the relayed vibration pattern on that side (e.g. Figure 3).

VI. Evaluation Framework

In order to evaluate the proposed solution, a test-bed framework was deployed. Four Android devices were used to emulate a genuine (proximity) and a relay transaction. The two transactions (genuine and relay) were being captured at the same time when required, or generated based on the captured data otherwise. Three scenarios were evaluated, 1) both transaction devices vibrating, 2) only the transaction terminal vibrating, and 3) only the transaction instrument vibrating. Each of these scenarios was evaluated with each of the six sensors 1) accelerometer, 2) gravity sensor, 3) gyroscope, 4) linear acceleration sensor, 5) magnetic field, and 6) rotation vector.

The communicating devices in the genuine transaction scenario were a Transaction Terminal (TT), and a Transaction Instrument (TI′). In the case of the relay transaction, the communicating devices were the same transaction terminal TT, and a distant Transaction Instrument (TI), located 5ft (1.5m) away. Devices TI′ and a transaction terminal co-located with TI (referred to as TT′) were used as relay devices. Figure 4 depicts the two transaction scenarios.

Device TI′ had a double role in this set-up. It acted as both a genuine and a relay device. This way, data from both scenarios could be collected simultaneously, when possible. The two devices of each pair were placed one on top of the other, such that the transaction terminal was at the bottom, facing downwards, and the transaction instrument on top, facing upwards. The two pairs (genuine and relay) were placed on separate tables, such that vibration of one pair would not affect the measurement of the other.

Five different attack techniques were performed against each sensor and each vibration scenario. Four scenarios were obtained by assuming that the attacker was capable of vibrating a pattern from one, both, or none of the transaction devices. A further scenario was obtained by assuming that the same pattern could be vibrated on both relay devices during a single transaction.

The notation that we use to describe the vibration mode of the transaction and relay devices in this and subsequent sections has the form of four characters, each character representing either ‘Vibration’ or ‘No Vibration’ using ‘V’ and ‘N’ respectively. The sequence of four characters represents the four devices of the test-bed, as ‘TT TI′ TT′ TI’. For example, *VNVN* means that TT and TT′ vibrate, while TI′ and TI do not.

TABLE I
Vibration Modes

	TT	TI′	TT′	TI	Experimentally Recorded	Generated By	
						Genuine Pair	Relay Pair
Both Vibrate	V	V	V	V	✓	—	—
	V	V	V*	V	✓	—	—
	V	N	V	V	✗	**VV**VV	**V**NVN + VVV**V**
	V	V	N	V	✗	**VV**VV	**V**VVV + NVN**V**
	V	N	N	V	✗	**VV**VV	**V**NVN + NVN**V**
TT Vibrates	V	N	V	N	✓	—	—
	V	V	V	N	✗	**VN**VN	**V**VVV + VNV**N**
	V	V	V*	N	✗	**VN**VN	**V**VV***N** (Other)
	V	V	N	N	✗	**VN**VN	**V**VVV + NNN**N**
	V	N	N	N	✗	**VN**VN	**V**NVN + NNN**N**
TI Vibrates	N	V	N	V	✓	—	—
	N	V	V	V	✗	**NV**NV	**N**VNV + VVV**V**
	N	V	V*	V	✓	—	—
	N	N	V	V	✗	**NV**NV	**N**NNN + VVV**V**
	N	N	N	V	✗	**NV**NV	**N**NNN + NVN**V**
Other	V	V	V*	N	✓	—	—
	N	N	N	N	✓	—	—

*TT′ vibrates the same pattern as TI′

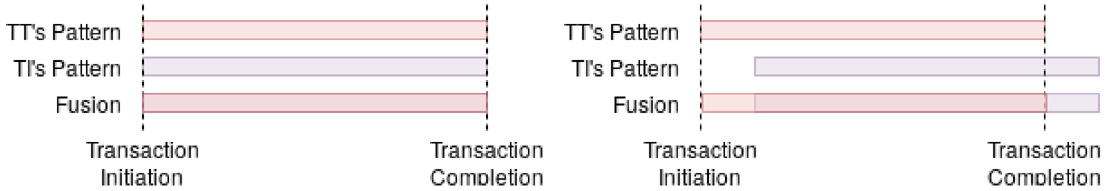
In total, 17 sets were part of the empirical analysis. Out of these 17, 5 sets were collected from field trials, and 10 sets were generated based on the raw data from the field trials (we refer to these sets as synthetic data). Two extra sets were collected from the field experiments, and they were used to assist the generation of synthetic datasets.

In the case of synthetic datasets, transactions collected by the pair TT–TI′ through field trials were regarded as genuine transactions. Relay transactions were based on combining genuine transaction measurements performed by TT and TI from different, experimentally collected sets. A point to note is that synthetic data was based on raw field-data, and was not generated randomly or based on artificial-replication of specific features.

Figure 5 depicts the generation process of the synthetic dataset *VNNV* as an example case. In this case, a genuine transaction (no relay involved) required both transaction devices to be vibrating. We used three sets of raw field-data for the construction of this case. As for genuine transactions, we considered transactions between TT and TI′ from the set *VVVV*. For relay transactions, we used raw field-data captured by device TT from the set *VNVN* (where TI′ did not vibrate and TT did) and data captured by device TI from the set *NVNV* (where TI vibrated and TT′ did not). Table I lists all the evaluated vibration scenarios, and experimentally recorded pairs that were used for the generation of synthetic pairs. Under the *‘Generated By’* heading of the table, bold characters in vibration mode sequences identify the raw field-data that constitute the synthetic data for the respective mode. The two extra sets (discussed before) are marked as *‘Other’* in the table.

The rationale for the synthetic data generation is due to one of the following two reasons - depending upon the vibration mode sequence:

1) Both a genuine and relay transaction could not be captured



(a) No Relay Attack Involved (b) Relay Attack Involved

Fig. 3. Vibration Fusion Without and With Relay Attack

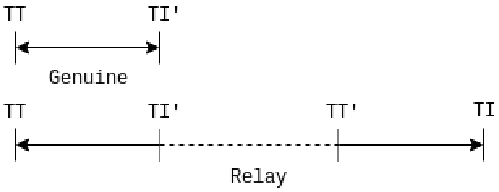


Fig. 4. Test-bed Scenarios

simultaneously (i.e. when TI′'s vibration mode was not the same as TI).

2) Recorded data could be used for the generation of a pair (e.g. in the case of VVNV, whose genuine pair can be taken from VVVV and the relay pair by using the same pair's TT recording with NVNV's TI).

A total of 400 transactions per captured pair were recorded for each sensor (totalling 16,800 transaction pairs), based on which 24,000 synthetic relay transactions were generated (400 for each synthetic set). Each synthetic set consisted of 400 genuine transactions, recorded through our test-bed, and 400 synthetic ones, based on the combination of recorded transactions.

Four Android applications were developed, one for each device. The devices used in the genuine pair were two Samsung Galaxy S4 (GT-I9500) devices. For the relay pair, a Samsung Galaxy S4 and a Nexus 5 were used. Each device was capable of communicating with the next device and the previous device in the chain (see Figure 4) through WiFi, using UDP packets. For example, device TI′ could communicate with device TT as part of the genuine transaction, and TT′ as part of the relay transaction.

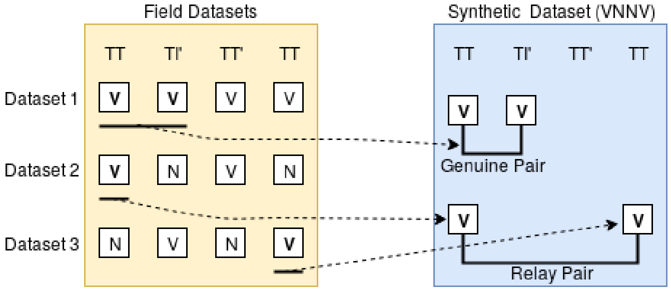


Fig. 5. Synthetic Data Generation (VNNV)

Figure 6 depicts the transaction process between the four devices. A transaction was initiated through a UDP packet from device TT to TI′, containing a transaction ID, the sensor to be used in the transaction, and which device(s) should vibrate (vibration mode). The same information, along with the random pattern that TI′ would vibrate, if the vibration mode required it, were forwarded from TI′ to TT′. Upon initiation of a transaction, a 500ms long random pattern was vibrated by one or both transaction devices, depending on the current transaction's vibration mode. At the same time the devices were recording the impact of the vibration using some sensor. Upon completion of the transaction, the measurements were saved on a local SQLite database, for each of the relevant devices for later extraction and analysis. Subsequent transactions were separated by a 5 second gap in order for the devices to rest.

VII. Transaction Data Analysis

For the analysis of the collected data, the Weka [33] machine learning software was used to determine whether the measurements of TI'_i and TI_i were in proximity with TT_i, i.e. whether it was possible to uniquely distinguish between $(\text{TI}'_i,\text{TT}_i)$ and $(\text{TI}_i,\text{TT}_i)$. On Android, all the examined sensors produce a vector of values consisting of x, y and z components. The vector magnitude (Eq. 1) was used as a general-purpose method for producing a single, combined value prior to generating the training sets for machine learning.

$$M = \sqrt{x^2 + y^2 + z^2} \qquad (1)$$

We calculated the 'optimal' threshold that separates illegitimate and legitimate attempts: the threshold that minimizes equal error rate. The threshold was based on the probability estimate output by the learned classification model, i.e. the estimated probability that a transaction is legitimate. To avoid optimistic bias in the error estimate when applying machine learning, it was necessary to perform a train-test experiment. More specifically, the full set of transaction pairs was split into a training set and a test set. The machine learning algorithm was applied to the training set to build a classification model that can output class probability estimates. Once the model had been built, it was applied to obtain probability estimates for the test set. When we split the data into training and test sets, we ensured that two transactions with the same ID (i.e., a legitimate and a fraudulent transaction that were recorded simultaneously) were either both in the training set or both in the test set, to avoid potential bias. Moreover, instead of using a

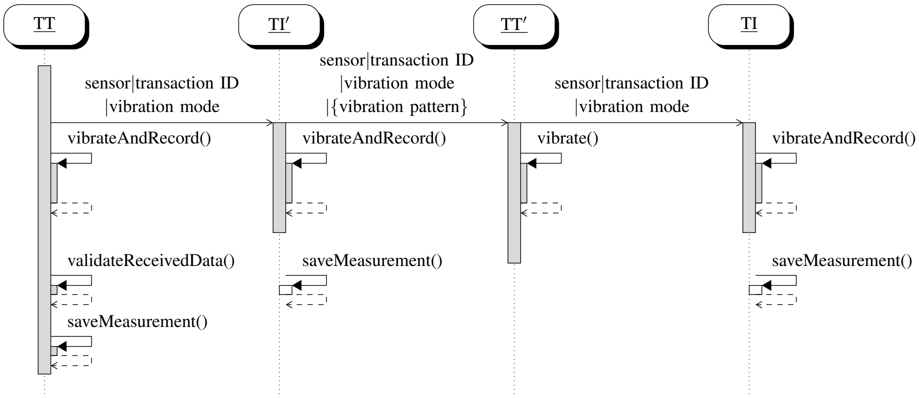


Fig. 6. Measurement Recording Overview

single train-test split, we used 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 was run 100 times on the 100 training sets, to build 100 models, and these 100 models were evaluated on the corresponding test sets. Performance estimates from the 100 test sets were averaged to obtain a final performance estimate.

We computed Equal Error Rates (EERs) by determining thresholds where the rate of false acceptances (FAR) was equal to the rate of false rejections (FRR) for each tested sensor. The FAR and FRR were measured at each threshold using Eqs. 2 and 3. To establish the relevant counts in these equations, the estimated probability of being legitimate, as output by the learned classifier, was compared to the threshold under consideration. 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. This would yield an equal error rate that is zero.

$$TAR = \frac{TA}{TA + FR} \qquad TRR = \frac{TR}{TR + FA} \tag{2}$$

$$FAR = 1 - TRR \qquad FRR = 1 - TAR \tag{3}$$

When training and testing each machine learning model, we used the individual differences $|A_{i,j} - B_{i,j}|$ (where $A_{i,j}$ refers to the j^{th} datapoint of the i^{th} sensor measurement on device A) 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 had 100 numeric features (corresponding to 500ms sampled at 5ms intervals). An example was labelled as positive if it corresponded to a legitimate transaction and as negative otherwise.

Weka's default settings were used for each of the first four machine learning algorithms in Table II. The first classifier, a random forest [34], is based on inducing a collection of decision trees, in a semi-random manner, from bootstrap samples of the original training set. By default, Weka generates a random forest consisting of 100 decision trees. The second classifier is the naive Bayes classifier. It assumes independence of the attributes given the classification. We used a Gaussian distribution with a diagonal covariance matrix, which is Weka's default. The third classifier, logistic regression, assumes that the log-odds of the class probabilities are linearly related to the attributes. The fourth classifier consists of decision trees obtained using the C4.5 algorithm [35]. For the fifth classifier, support vector machines (SVMs) optimised with the SMO algorithm [36], the complexity parameter C and the width of the RBF kernel γ were tuned using a grid search by performing internal cross-validation on the training data. AUROC was the criterion being optimized in the grid search.

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

Vibration Mode	**Classifier**				
	Random Forest	**Naive Bayes**	**Decision Tree**	**Logistic Regression**	**Support Vector Machine**
		Accelerometer			
Both Vibrate	0.069	0.094	0.104	0.172	**0.067**
TT Vibrates	0.096	0.185	0.143	0.247	**0.084**
TI Vibrates	0.083	0.110	0.123	0.186	**0.078**
		Gravity			
Both Vibrate	0.500	0.493	0.500	**0.488**	0.492
TT Vibrates	0.499	**0.484**	0.500	0.486	0.490
TI Vibrates	0.503	**0.494**	0.500	0.500	0.497
		Gyroscope			
Both Vibrate	0.003	**0.001**	0.009	0.014	0.003
TT Vibrates	**0.001**	**0.001**	0.003	0.003	**0.001**
TI Vibrates	**0.001**	0.003	0.006	0.006	0.002
		Linear Acceleration			
Both Vibrate	0.015	0.017	0.042	0.047	**0.014**
TT Vibrates	0.016	**0.015**	0.038	0.044	0.016
TI Vibrates	**0.002**	**0.002**	0.020	0.009	0.003
		Magnetic Field			
Both Vibrate	0.059	0.315	0.187	0.477	**0.031**
TT Vibrates	0.065	0.202	0.132	0.291	**0.035**
TI Vibrates	0.146	0.236	0.149	0.409	**0.138**
		Rotation Vector			
Both Vibrate	0.008	0.008	0.031	0.113	**0.007**
TT Vibrates	**0.002**	0.010	0.014	0.106	**0.002**
TI Vibrates	0.016	**0.005**	0.024	0.039	0.013

Table II shows the results obtained from the different machine learning algorithms. These results were obtained by averaging estimated equal error rates across all the relevant attack scenarios from Table I. The best result for each sensor is shown in bold. The results for individual attack scenarios can be found in Appendices A (both devices vibrating scenario), B (terminal only vibrating scenario), and C (instrument only vibrating scenario). For each dataset/sensor and learning algorithm, these tables shows the mean and standard deviation of the 100 equal error rate estimates obtained using 10-fold cross-validation repeated 10 times.

Table II shows that SVMs produce the lowest equal error rate for most sensors/vibration modes. Random forests and naive Bayes also demonstrated low equal error rates for many sensors/vibration modes. 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 SVM classifiers yields the following ranking (best to worst): gyroscope, rotation vector, linear acceleration, magnetic field, accelerometer, and gravity. The same ranking holds for random forests. Based on naive Bayes, the ranking is: gyroscope, rotation vector, linear acceleration, accelerometer, magnetic field, and gravity. Finally, none of the vibration modes seems to be performing substantially better than the others, except in the cases of magnetic field, where the TI only vibrating performed substantially worse than the rest, and the case of accelerometer, where TT only vibrating performed substantially worse.

VIII. Discussion and Outcome

The analysis of the experimental data indicates that the effectiveness of the proposed solution as a PRAD mechanism is high. Using most sensors, the empirically observed relay attack detection rate is higher than that of previously proposed solutions [14], [16], [37], [38]. No particular attack against the proposed solution performs better than the others. However, some attacks perform better than others depending on classifiers and sensors involved. The performance of the solution is not significantly affected by using any of the evaluated attacks.

Even though a different smartphone was used as device TT′, no significant impact was observed in the results. Observing the results of TI only vibrating, where TT′'s impact in the transaction was negligible, except in the case of using the magnetic field sensor, no significant degradation in the effectiveness of the proposed solution was observed. However, further investigation might be required before deployment in the real world. More aspects should also be examined, like the impact of protective smartphone cases on the proposed solution.

Moreover, even though no significant performance differences were observed between different vibration modes, in order to prevent acoustic attacks by a more resourceful attacker, both devices vibrating might be preferred. Observing the vibrated pattern has been demonstrated with relatively high success rate in [32], so it might have some effectiveness against the proposed technique. Fusing vibrations from both devices can be a significant barrier against off-the-shelf attackers, as explained in Section V-B.

Finally, no additional or non-standard hardware on the TI side is likely to be required, like in many of the previously proposed solutions, as in [26], and [16]. The examined sensors are available in a large variety of smartphones, and most modern smartphones will be equipped with at least one of them. Many of the previously proposed solutions might also be vulnerable in the presence of an attacker with context manipulating capabilities [39]. Since this solution is not dependant upon the surrounding environment, such attacks do not apply, unless the attacker physically tampers with the devices.

IX. Conclusion and Future Work

Communicating devices are vulnerable to relay attacks. Traditional distance bounding protocols that aim to counter such attacks might not be applicable in the field of smartphones. Alternative approaches against off-the-shelf attackers have been proposed, mostly based on sensing of the ambient environment. However, these might not be suitable or secure under certain scenarios, like in the case of transactions with industry imposed time restrictions of up to 500ms (e.g. EMV and transportation related transactions). The generation of a random bit/stream-based Artificial Ambient Environment (AAE) by peripherals of the transaction devices has demonstrated promising results as a Proximity and Relay Attack Detection (PRAD) mechanism when using infrared light as an AAE actuator.

In this work we have investigated the use of vibration as an AAE actuator. During the transaction, one (unidirectional) or both (bidirectional) transaction devices vibrate some random pattern, which is measured by a selected ambient sensor by both transaction devices. The two measurements are then compared for similarity, in order to establish proximity evidence.

A test-bed was designed and built for the evaluation of the proposed solution as a PRAD mechanism. We evaluated three different vibration modes (only one, or both the transaction devices vibrating a random pattern), against five different attack scenarios. Six ambient sensors were evaluated. A total of 36,000 genuine-relay transaction pairs, based on field trials, were collected, or synthetically generated. Analysis using five machine learning classifiers was performed, indicating high classification accuracy between genuine and relay transactions on all vibration modes, and using most of the sensors.

As part of our ongoing investigation, we are planning to expand the study by using more devices. We are also planning to examine whether using multiple sensors simultaneously (sensor fusion) will increase the accuracy of the system. Finally, we are planning to conduct a user study in order to explore the effectiveness of the proposed solution in this case, along with potential usability concerns.

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 Computer Society, 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 Systems,” 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 Relay Attack on Contactless Transactions by Using NFC Mobile Phones.” *IACR Cryptology Archive*, vol. 2011, p. 618, 2011.

[5] G. P. Hancke, “Distance-bounding for RFID: Effectiveness of ‘terrorist fraud’ in the presence of bit errors,” in *2012 IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, Nov 2012, pp. 91–96.

[6] 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.

[7] 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.

[8] 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.

[9] ——, *Relay Attacks on Secure Element-Enabled Mobile Devices*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 1–12. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30436-1_1

[10] 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.

[11] 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 Computer Society, 2005, pp. 67–73.

[12] 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.

[13] 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, 2015 First Conference on*. IEEE, 2015, pp. 1–6.

[14] 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.

[15] D. Ma, N. Saxena, T. Xiang, and Y. Zhu, “Location-aware and safer cards: Enhancing rfid security and privacy via location sensing,” *IEEE TDSC*, vol. 10, no. 2, pp. 57–69, 2013.

[16] 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.

[17] 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, 2014 IEEE International Conference on*. IEEE, 2014, pp. 163–171.

[18] 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.

[19] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, “Amigo: Proximity-Based Authentication of Mobile Devices,” in *UbiComp 2007*, ser. LNCS, J. Krumm, G. Abowd, A. Seneviratne, and T. Strang, Eds. Springer, 2007, pp. 253–270.

[20] “How to Optimize the Consumer Contactless Experience? The Perfect Tap,” MasterCard, Tech. Rep., 2014.

[21] “Transactions Acceptance Device Guide (TADG),” VISA, Specification Version 3.1, November 2016.

[22] “EMV Contactless Specifications for Payment Systems: Book A - Architecture and General Requirements,” EMVCo, LLC, Spec V2.6, April 2016.

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

[24] I. Gurulian, C. Shepherd, E. Frank, K. Markantonakis, R. Akram, and K. Mayes, “On the Effectiveness of Ambient Sensing for NFC-based Proximity Detection by Applying Relay Attack Data,” in *The 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, ser. TrustCom ’17. IEEE, August 2017.

[25] 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.

[26] I. Gurulian, R. N. Akram, K. Markantonakis, and K. Mayes, “Preventing Relay Attacks in Mobile Transactions Using Infrared Light,” in *Proceedings of the Symposium on Applied Computing*, ser. SAC ’17. New York, NY, USA: ACM, 2017, pp. 1724–1731.

[27] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, “Practical NFC Peer-to-Peer Relay Attack Using Mobile Phones,” in *Radio Frequency Identification: Security and Privacy Issues*, ser. LNCS. Springer, 2010, vol. 6370, pp. 35–49.

[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,” Newcastle University, Tech. Rep. CS-TR-1428, July 2014.

[29] 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.

[30] M. L. Polla, F. Martinelli, and D. Sgandurra, “A Survey on Security for Mobile Devices,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 446–471, First 2013.

[31] “Digital Payments Solutions Industry Considerations,” The UK Cards Association, Online Report, June 2017. [Online]. Available: http://www.theukcardsassociation.org.uk/wm_documents/Digital%20Wallets%20-%20Industry%20Considerations%20Outline.pdf

[32] T. Halevi and N. Saxena, “Acoustic Eavesdropping Attacks on Constrained Wireless Device Pairing,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 563–577, March 2013.

[33] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench. Online Appendix for “Data Mining: Practical Machine Learning Tools and Techniques”*, 4th ed. Burlington, MA: Morgan Kaufmann, 2016.

[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,” in *2nd International Conference on Research in Security Standardisation (SSR’15)*, October 2014.

[38] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi, “Using contextual co-presence to strengthen Zero-Interaction Authentication: Design, integration and usability,” *Pervasive and Mobile Computing*, vol. 16, Part B, pp. 187 – 204, 2015, Selected Papers from the Twelfth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2014). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119214001771

[39] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan, “Contextual proximity detection in the face of context-manipulating adversaries,” *CoRR*, vol. abs/1511.00905, 2015. [Online]. Available: http://arxiv.org/abs/1511.00905

APPENDIX A
RESULTS: BOTH DEVICES VIBRATE

TABLE III
ESTIMATED EER FOR MACHINE LEARNING ALGORITHMS, OBTAINED BY REPEATING 10-FOLD CROSS-VALIDATION 10 TIMES (CASE: BOTH DEVICES VIBRATE)

Vibration Mode	Classifier				
	Random Forest	**Naive Bayes**	**Decision Tree**	**Logistic Regression**	**Support Vector Machine**
Accelerometer					
VVVV	0.073 ± 0.030	0.115 ± 0.041	0.120 ± 0.047	0.200 ± 0.042	0.058 ± 0.026
VVV*V	0.075 ± 0.032	0.111 ± 0.035	0.122 ± 0.041	0.215 ± 0.046	0.067 ± 0.028
VNVV	0.051 ± 0.027	0.066 ± 0.026	0.079 ± 0.036	0.148 ± 0.039	0.049 ± 0.022
VVNV	0.068 ± 0.030	0.100 ± 0.039	0.099 ± 0.041	0.147 ± 0.042	0.084 ± 0.032
VNNV	0.076 ± 0.033	0.078 ± 0.038	0.099 ± 0.047	0.148 ± 0.036	0.079 ± 0.036
Gravity					
VVVV	0.497 ± 0.048	0.498 ± 0.042	0.500 ± 0.000	0.459 ± 0.041	0.500 ± 0.000
VVV*V	0.508 ± 0.053	0.505 ± 0.041	0.500 ± 0.000	0.540 ± 0.038	0.500 ± 0.000
VNVV	0.509 ± 0.053	0.487 ± 0.056	0.500 ± 0.000	0.453 ± 0.050	0.458 ± 0.050
VVNV	0.493 ± 0.054	0.492 ± 0.041	0.500 ± 0.000	0.510 ± 0.041	0.500 ± 0.000
VNNV	0.494 ± 0.053	0.483 ± 0.055	0.500 ± 0.000	0.478 ± 0.056	0.502 ± 0.062
Gyroscope					
VVVV	0.004 ± 0.009	0.004 ± 0.011	0.009 ± 0.014	0.021 ± 0.016	0.004 ± 0.009
VVV*V	0.002 ± 0.008	0.000 ± 0.000	0.008 ± 0.014	0.009 ± 0.013	0.003 ± 0.009
VNVV	0.004 ± 0.010	0.001 ± 0.005	0.010 ± 0.015	0.008 ± 0.014	0.003 ± 0.009
VVNV	0.005 ± 0.011	0.002 ± 0.008	0.010 ± 0.016	0.019 ± 0.017	0.003 ± 0.009
VNNV	0.002 ± 0.010	0.000 ± 0.003	0.009 ± 0.015	0.013 ± 0.016	0.002 ± 0.007
Linear Acceleration					
VVVV	0.000 ± 0.000	0.000 ± 0.000	0.021 ± 0.025	0.000 ± 0.000	0.000 ± 0.000
VVV*V	0.000 ± 0.000	0.000 ± 0.000	0.023 ± 0.027	0.000 ± 0.000	0.000 ± 0.000
VNVV	0.016 ± 0.019	0.015 ± 0.019	0.045 ± 0.032	0.057 ± 0.032	0.016 ± 0.018
VVNV	0.027 ± 0.021	0.028 ± 0.022	0.058 ± 0.034	0.090 ± 0.042	0.026 ± 0.022
VNNV	0.032 ± 0.025	0.042 ± 0.028	0.065 ± 0.042	0.088 ± 0.036	0.026 ± 0.023
Magnetic Field					
VVVV	0.082 ± 0.030	0.316 ± 0.060	0.218 ± 0.073	0.488 ± 0.062	0.042 ± 0.022
VVV*V	0.100 ± 0.029	0.315 ± 0.053	0.220 ± 0.068	0.530 ± 0.052	0.045 ± 0.026
VNVV	0.008 ± 0.012	0.312 ± 0.059	0.140 ± 0.058	0.441 ± 0.064	0.009 ± 0.013
VVNV	0.097 ± 0.033	0.311 ± 0.061	0.219 ± 0.100	0.484 ± 0.060	0.048 ± 0.023
VNNV	0.009 ± 0.014	0.322 ± 0.046	0.136 ± 0.052	0.444 ± 0.060	0.009 ± 0.014
Rotation Vector					
VVVV	0.007 ± 0.012	0.005 ± 0.017	0.031 ± 0.027	0.109 ± 0.047	0.008 ± 0.013
VVV*V	0.007 ± 0.012	0.005 ± 0.016	0.035 ± 0.027	0.114 ± 0.054	0.005 ± 0.010
VNVV	0.009 ± 0.012	0.010 ± 0.024	0.031 ± 0.026	0.109 ± 0.045	0.007 ± 0.012
VVNV	0.009 ± 0.013	0.009 ± 0.024	0.030 ± 0.026	0.116 ± 0.051	0.007 ± 0.014
VNNV	0.008 ± 0.012	0.009 ± 0.024	0.030 ± 0.025	0.116 ± 0.051	0.007 ± 0.014

*TT′ vibrates the same pattern as TI′

APPENDIX B
RESULTS: TRANSACTION TERMINAL VIBRATES

TABLE IV
ESTIMATED EER FOR MACHINE LEARNING ALGORITHMS, OBTAINED BY REPEATING 10-FOLD CROSS-VALIDATION 10 TIMES (CASE: TRANSACTION TERMINAL VIBRATES)

Vibration Mode	Classifier				
	Random Forest	**Naive Bayes**	**Decision Tree**	**Logistic Regression**	**Support Vector Machine**
Accelerometer					
VNVN	0.055 ± 0.025	0.111 ± 0.039	0.095 ± 0.041	0.168 ± 0.038	0.033 ± 0.019
VVVN	0.037 ± 0.026	0.082 ± 0.039	0.068 ± 0.033	0.148 ± 0.049	0.015 ± 0.016
VVV*N	0.121 ± 0.036	0.465 ± 0.271	0.216 ± 0.054	0.474 ± 0.049	0.092 ± 0.048
VVNN	0.138 ± 0.040	0.141 ± 0.049	0.174 ± 0.050	0.248 ± 0.055	0.146 ± 0.040
VNNN	0.131 ± 0.036	0.124 ± 0.041	0.164 ± 0.056	0.195 ± 0.043	0.133 ± 0.037
Gravity					
VNVN	0.503 ± 0.053	0.492 ± 0.041	0.500 ± 0.000	0.481 ± 0.042	0.500 ± 0.000
VVVN	0.496 ± 0.058	0.473 ± 0.054	0.500 ± 0.000	0.484 ± 0.056	0.483 ± 0.083
VVV*N	0.489 ± 0.052	0.447 ± 0.056	0.500 ± 0.000	0.471 ± 0.056	0.457 ± 0.055
VVNN	0.503 ± 0.059	0.481 ± 0.051	0.500 ± 0.000	0.484 ± 0.052	0.509 ± 0.081
VNNN	0.502 ± 0.053	0.527 ± 0.049	0.500 ± 0.000	0.508 ± 0.041	0.500 ± 0.000
Gyroscope					
VNVN	0.002 ± 0.006	0.000 ± 0.000	0.003 ± 0.008	0.001 ± 0.006	0.000 ± 0.003
VVVN	0.000 ± 0.000	0.001 ± 0.007	0.003 ± 0.008	0.003 ± 0.011	0.001 ± 0.005
VVV*N	0.000 ± 0.000	0.005 ± 0.011	0.003 ± 0.010	0.001 ± 0.004	0.000 ± 0.000
VVNN	0.000 ± 0.000	0.001 ± 0.004	0.003 ± 0.009	0.005 ± 0.011	0.001 ± 0.004
VNNN	0.002 ± 0.006	0.000 ± 0.000	0.002 ± 0.007	0.005 ± 0.011	0.003 ± 0.008
Linear Acceleration					
VNVN	0.000 ± 0.000	0.000 ± 0.000	0.012 ± 0.015	0.000 ± 0.000	0.001 ± 0.004
VVVN	0.007 ± 0.012	0.004 ± 0.009	0.031 ± 0.021	0.037 ± 0.022	0.003 ± 0.008
VVV*N	0.000 ± 0.000	0.000 ± 0.000	0.013 ± 0.020	0.000 ± 0.000	0.000 ± 0.000
VVNN	0.034 ± 0.025	0.034 ± 0.026	0.058 ± 0.033	0.087 ± 0.032	0.034 ± 0.023
VNNN	0.038 ± 0.022	0.035 ± 0.025	0.077 ± 0.037	0.095 ± 0.036	0.042 ± 0.025
Magnetic Field					
VNVN	0.020 ± 0.019	0.328 ± 0.051	0.146 ± 0.053	0.443 ± 0.064	0.014 ± 0.018
VVVN	0.117 ± 0.042	0.321 ± 0.058	0.203 ± 0.114	0.480 ± 0.053	0.058 ± 0.030
VVV*N	0.082 ± 0.035	0.125 ± 0.040	0.123 ± 0.048	0.183 ± 0.045	0.045 ± 0.025
VVNN	0.080 ± 0.028	0.122 ± 0.036	0.119 ± 0.041	0.176 ± 0.039	0.044 ± 0.022
VNNN	0.024 ± 0.020	0.116 ± 0.039	0.071 ± 0.041	0.171 ± 0.041	0.014 ± 0.018
Rotation Vector					
VNVN	0.006 ± 0.013	0.005 ± 0.016	0.022 ± 0.023	0.119 ± 0.046	0.006 ± 0.012
VVVN	0.005 ± 0.011	0.004 ± 0.013	0.020 ± 0.022	0.118 ± 0.048	0.005 ± 0.010
VVV*N	0.000 ± 0.002	0.014 ± 0.019	0.011 ± 0.015	0.096 ± 0.038	0.000 ± 0.000
VVNN	0.000 ± 0.000	0.014 ± 0.019	0.008 ± 0.015	0.100 ± 0.045	0.000 ± 0.000
VNNN	0.000 ± 0.000	0.014 ± 0.019	0.008 ± 0.012	0.099 ± 0.046	0.000 ± 0.000

Continues in the next column…

Continuation from previous column…

*TT′ vibrates the same pattern as TI′

*TT′ vibrates the same pattern as TI′

APPENDIX C
RESULTS: TRANSACTION INSTRUMENT VIBRATES

TABLE V
ESTIMATED EER FOR MACHINE LEARNING ALGORITHMS, OBTAINED BY REPEATING 10-FOLD CROSS-VALIDATION 10 TIMES (CASE: TRANSACTION INSTRUMENT VIBRATES)

Vibration Mode	Classifier				
	Random Forest	**Naive Bayes**	**Decision Tree**	**Logistic Regression**	**Support Vector Machine**
Accelerometer					
NVNV	0.101 ± 0.038	0.111 ± 0.043	0.125 ± 0.052	0.180 ± 0.040	0.104 ± 0.039
NVVV	0.026 ± 0.021	0.037 ± 0.028	0.069 ± 0.033	0.090 ± 0.031	0.012 ± 0.014
NVV*V	0.071 ± 0.030	0.115 ± 0.042	0.110 ± 0.041	0.188 ± 0.042	0.046 ± 0.026
NNVV	0.100 ± 0.036	0.128 ± 0.039	0.148 ± 0.054	0.245 ± 0.054	0.082 ± 0.030
NNNV	0.118 ± 0.040	0.158 ± 0.059	0.164 ± 0.049	0.228 ± 0.041	0.144 ± 0.037
Gravity					
NVNV	0.485 ± 0.056	0.487 ± 0.047	0.500 ± 0.000	0.502 ± 0.040	0.500 ± 0.000
NVVV	0.498 ± 0.056	0.506 ± 0.041	0.500 ± 0.000	0.497 ± 0.048	0.500 ± 0.000
NVV*V	0.501 ± 0.049	0.488 ± 0.042	0.500 ± 0.000	0.477 ± 0.039	0.500 ± 0.000
NNVV	0.514 ± 0.058	0.493 ± 0.051	0.500 ± 0.000	0.506 ± 0.049	0.482 ± 0.091
NNNV	0.516 ± 0.054	0.497 ± 0.051	0.500 ± 0.000	0.517 ± 0.054	0.501 ± 0.123
Gyroscope					
NVNV	0.000 ± 0.003	0.000 ± 0.000	0.005 ± 0.011	0.008 ± 0.014	0.000 ± 0.002
NVVV	0.002 ± 0.006	0.011 ± 0.015	0.004 ± 0.009	0.011 ± 0.014	0.006 ± 0.012
NVV*V	0.002 ± 0.008	0.003 ± 0.009	0.007 ± 0.013	0.003 ± 0.008	0.002 ± 0.007
NNVV	0.002 ± 0.006	0.000 ± 0.000	0.006 ± 0.012	0.004 ± 0.009	0.000 ± 0.004
NNNV	0.000 ± 0.003	0.000 ± 0.000	0.006 ± 0.012	0.005 ± 0.010	0.000 ± 0.003
Linear Acceleration					
NVNV	0.001 ± 0.004	0.000 ± 0.003	0.024 ± 0.023	0.007 ± 0.012	0.002 ± 0.006
NVVV	0.000 ± 0.000	0.000 ± 0.000	0.022 ± 0.021	0.000 ± 0.000	0.000 ± 0.000
NVV*V	0.000 ± 0.000	0.000 ± 0.000	0.017 ± 0.020	0.000 ± 0.000	0.000 ± 0.000
NNVV	0.002 ± 0.007	0.002 ± 0.007	0.017 ± 0.021	0.014 ± 0.015	0.004 ± 0.009
NNNV	0.009 ± 0.012	0.008 ± 0.014	0.020 ± 0.022	0.025 ± 0.019	0.010 ± 0.014
Magnetic Field					
NVNV	0.267 ± 0.051	0.273 ± 0.052	0.124 ± 0.109	0.435 ± 0.053	0.276 ± 0.048
NVVV	0.116 ± 0.039	0.119 ± 0.049	0.110 ± 0.062	0.197 ± 0.047	0.126 ± 0.045
NVV*V	0.254 ± 0.056	0.262 ± 0.060	0.198 ± 0.129	0.444 ± 0.056	0.268 ± 0.050
NNVV	0.045 ± 0.025	0.253 ± 0.056	0.171 ± 0.063	0.483 ± 0.066	0.010 ± 0.013
NNNV	0.046 ± 0.026	0.271 ± 0.060	0.143 ± 0.087	0.488 ± 0.064	0.011 ± 0.013
Rotation Vector					
NVNV	0.019 ± 0.022	0.003 ± 0.012	0.030 ± 0.026	0.041 ± 0.025	0.016 ± 0.020
NVVV	0.009 ± 0.015	0.010 ± 0.016	0.007 ± 0.012	0.003 ± 0.008	0.000 ± 0.000
NVV*V	0.017 ± 0.018	0.004 ± 0.012	0.024 ± 0.024	0.060 ± 0.024	0.017 ± 0.019
NNVV	0.019 ± 0.019	0.004 ± 0.015	0.030 ± 0.024	0.050 ± 0.029	0.015 ± 0.019
NNNV	0.017 ± 0.019	0.004 ± 0.015	0.030 ± 0.024	0.041 ± 0.025	0.015 ± 0.018

*TT′ vibrates the same pattern as TI′