May The Force Be With You: Force-Based Relay Attack Detection

Iakovos Gurulian[1], Gerhard P. Hancke[2], Konstantinos Markantonakis[1], and Raja Naeem Akram[1]

[1] Information Security Group Smart Card Centre, Royal Holloway, University of London, Egham, United Kingdom
{Iakovos.Gurulian.2014, k.markantonakis, r.n.akram}@rhul.ac.uk
[2] Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong
gp.hancke@cityu.edu.hk

Abstract. Relay attacks pose a significant threat against communicating devices that are required to operate within a short-distance from each other and a restricted time frame. In the field of smart cards, distance bounding protocols have been proposed as an effective countermeasure, whereas, in the field of smartphones, many proposals suggest the use of (natural) ambient sensing as an effective alternative. However, empirical evaluation of the proposals carried out in existing literature has reported negative results in using natural ambient sensing in distance- and time-restricted scenarios, like EMV contactless payments that require the proximity to be less than 3cm and the transaction duration to be under 500ms. In this paper, we propose a novel approach for Proximity and Relay Attack Detection (PRAD), using bidirectional sensing and comparing button presses and releases behaviour (duration of press and gap between presses and releases), performed by a genuine user during the transaction. We implemented a test-bed environment to collect training and analysis data from a set of users, for both the genuine and attacker-involved transactions. Analysis of the collection-data indicates a high effectiveness of the proposed solution, as it was successful in distinguishing between proximity and relay-attack transactions, using thresholds set after analysis of genuine training transaction data. Furthermore, perfect classification of genuine and relay-attack transactions was achieved by using well-known machine learning classifiers.

Keywords: Mobile Payments, Relay Attacks, Contactless, Experimental Analysis

1 Introduction

Relay attacks [6, 8, 34] are passive man-in-the-middle attacks, aiming to extend the physical distance of devices involved in a transaction beyond their operating environment. Contactless smart cards [8, 13, 14, 16, 18], as well as smartphones [6, 20, 23, 24] are susceptible to relay attacks. Using such attacks, an attacker can gain unauthorised access to services and facilities that a genuine user is eligible for, like payments and access to buildings.

In the field of smart cards, distance-bounding protocols have been proposed as an effective countermeasure [15, 28]. However, distance bounding protocols may not be applicable in the field of smartphones due to unpredictable behaviour related to their multi-tasking architecture and the multitude of hardware components [31].

In recent years, a number of proposals suggest the use of ambient sensing as an alternative Proximity and Relay Attack Detection (PRAD) mechanism against the off-the-shelf attacker [12, 19, 26, 30, 32, 33]. Such proposals rely on the collection of data over a period of time from both transaction devices, using ambient sensors, and subsequently comparing the collected data for similarity.

Specific scenarios susceptible to relay attacks, like EMV contactless payments and transport ticketing, have industry-imposed time limits regarding the completion of a contactless transaction. In the case of EMV, the limit is 500ms [2–4], and in the case of transport ticketing, typically between 300 and 500ms [1]. Recent evaluations of previously proposed PRAD techniques have yielded limited evidence that ambient sensing can effectively counter relay attacks in contactless transactions under 500ms [11, 25]. The generation of an artificial ambient environment (AAE) has been proposed instead, and evaluated using infrared light, with promising results [10]. However, attacks against ambient sensing (in sound, WiFi, Bluetooth, temperature, humidity, gas, and altitude) have been demonstrated in the presence of an attacker with context manipulating capabilities [27]. Even though such attacks might not be able to cause a false positive in the case of using infrared light as an AAE actuator, denial of service attacks might be achieved.

In this paper we propose a novel approach towards PRAD based on sensing button presses on the user's smartphone by both transaction devices (transaction terminal and transaction 'user' smartphone) simultaneously. During the time of the transaction, the user is requested to press four buttons randomly picked by a smartphone application. The input button sequence, as well as timings of button presses and between consequent button presses (referred to as *'releases'*) are captured by both transaction devices and used as features for similarity comparison. Empirical evaluation demonstrated high success rate of the method as a PRAD mechanism. Using threshold-based evaluation, all the attack attempts were detected. Perfect classification was achieved by using the Support Vector Machine classifier. Near-perfect classification (up to 99.8%) was achieved by other well-known machine learning classifiers.

The main contributions of this paper are:

- Force Sensing Relay Attack Detection: A novel approach for PRAD based on force sensing by the transaction devices (Section 4).
- Evaluation Frameworks: The design of two evaluation frameworks, for evaluating the proposed solution as a proximity detection mechanism (Section 6), and as a relay-attack detection mechanism, by subjecting it against a set of volunteers attempting to attack the scheme (Section 7).

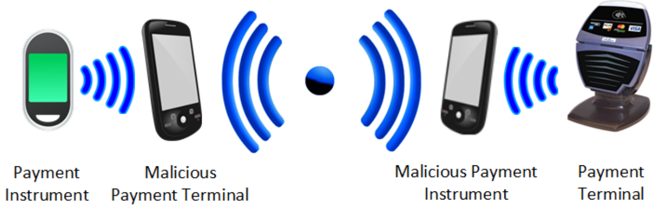


Fig. 1. Overview of a Relay Attack

- Two-Fold Evaluation: Threshold- and machine learning-based evaluation of the proposed system, in the presence of a relay-attacker.

2 Relay Attacks

A variety of applications are 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 masquerading (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 will be performed.

Relay attacks against mobile devices have been demonstrated [7,23,24]. 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. This approach has demonstrated positive results when using infrared light as an AAE actuator in transactions with industry imposed time limits of up to 500ms, like EMV contactless payments and transport ticketing.

For 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).

3 Related Work

In this section, we identify and summarise key pieces of related work that have suggested using natural ambient sensing as a PRAD mechanism.

Ma et al. [19] 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. [12] 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. [33] 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. [32] 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. [21] 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. [30] 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. [26] 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.

Jin et al. [17] used the magnetometer in order to pair devices that are in proximity. An average of 4.5s is required for the pairing to succeed. The authors only focus on proximity detection, and do not claim that this method can be used as an effective relay attack detection mechanism.

In [11] and [25], the effectiveness of recording the natural ambient environment in short transactions (up to $500ms$) was empirically evaluated, with different results from the existing literature. Comparison algorithms used in previous works, as well as machine learning techniques, produced very high false negative results.

Gurulian [10] proposed a relay attack detection framework by using artificial ambient environments (AAE). Infrared light was evaluated as an AAE actuator. Relay attacks were successfully detected, while the false rejection rate was approximately 2%.

4 Force-Sensing PRAD

In this section we present the theoretical foundation of the proposed framework, and the threat model.

4.1 PRAD Framework

During the course of a transaction, the user is called to position the Transaction Instrument (TI) on an extension force-sensitive panel of the Transaction Terminal (TT) in order to complete the PRAD procedure. A smartphone application (running on TI) presents to the user an interface with buttons that the user is called to tap. Alternative interface design approaches can be followed. For example, the application might present buttons with numbers and ask the user to input a provided 4 digit sequence. Alternatively, the application can present a button at a time that the user has to press in order for the next button to appear, until all the buttons of the sequence have been pressed. It should be stressed that even though the aforementioned method was used during the evaluation of the proposed framework (Section 5), it is not restrictive, as alternative approaches can be used instead. Figure 2 presents the basic architecture of the framework, when using the later application design method. Also, the use of a Personal Identification Number (PIN) and other forms of user identification codes may pose a threat, as an attacker could potentially capture them if the user attempts to perform a transaction with a malicious terminal.

Each button is assigned an ID based on the location of the button (e.g. the top left button is assigned the ID *'1'*). When the user taps on a button, both transaction devices record the ID of the button that was pressed, and the duration of the button press. The duration between subsequent button presses is also recorded by the two devices. Further features can also potentially be used, like the amount of pressure applied, but were not considered in this work due to limitations inflicted by the architecture of the majority of modern smartphones (further discussion in Section 8).

In order for device TI to recognise the buttons and timings, simple API calls are required. Device TT can recognise the buttons and timings based on the coordinates and duration of the detected pressure on the force-sensitive panel.

The assumption is that the two devices are going to record approximately the same values, and that an attacker has a low probability of accurately replicating

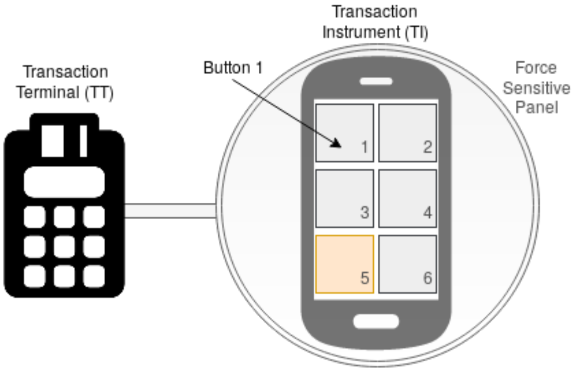


Fig. 2. The Basic Framework Architecture

the movement of a genuine user. The accuracy of the system should be at the millisecond (ms) level. The captured data of the two devices should provide sufficient proximity evidence when compared against each other, while data captured when a relay attack is taking place should be detectable at a high rate, in accordance to the requirements of the deployment scenario.

The proximity verification process can take place during the course of the transaction by one of the communicating devices, or afterwards, by a Trusted Third Party (TTP). The captured data should be communicated between the devices or to the TTP in an encrypted and authenticated form, but discussion regarding the trusted comparison party's (i.e. the TTP or one of the transaction devices) architecture is out of the scope of this work. As the main focus of this work is to examine the effectiveness of the proposed solution as a PRAD mechanism, we limit the discussion and do not investigate the integration with existing applications.

4.2 Threat Model

In this paper, the attacker is of opportunistic nature and requires no prior interaction or knowledge of neither TT nor TI. However, the potential implications if TT or TI are compromised are out of the scope of this paper, since a relay attack may not be necessary to achieve the same goals under these circumstances. We focus primarily on the issue of genuine devices requiring proximity assurance in order to conduct a legitimate transaction.

The attacker only has access to off-the-shelf relay equipment. Usually transaction limits apply on mobile transactions, for example a £30 limit on digital payment transactions in the UK [5]. Moreover, an attacker that might be using more advanced techniques, like a robotic arm that replicates the movement of a genuine user with accuracy, is likely going to be detected by the genuine device operators. Therefore, a very powerful attacker might not be a major concern.

5 Test-bed Architecture

We designed and built a test-bed in order to evaluate the proposed solution. Android devices were used to represent TI, and an Arduino-based prototype was developed and used as TT.

An Android application was built and installed on three devices; a light and small Android smartphone, a heavier and larger Android smartphone, and an Android tablet device. As the first, a Samsung Galaxy S5 Mini (SM-G800F) was used. It features a 4.5-inch display, and weighs 120 grams. A Samsung Galaxy S4 (GT-I9500) device was used as the second device. It features a 5-inch display and weighs 130 grams. Finally, a Nexus 9 was used as the tablet. Its display is 8.9-inches and weighs 425 grams[3].

The Android application, running on TI, displayed six equal size zones on the touch screen (as in Figure 2). In this paper, we refer to these zones as 'buttons'.

[3] All device characteristics found at http://www.gsmarena.com/

Initially, a random button on the screen was highlighted using a colour. This indicated to the user to press the randomly highlighted button on the screen. Releasing a button would trigger it to disappear, and the application would randomly pick and highlight a second button. A total of four buttons were randomly highlighted on each transaction. A random shared ID assigned to each transaction, along with the button sequence IDs, and the timings of button presses and interval duration between subsequent button presses were appended to a local CSV file that was later extracted from the device for data comparison.

An Arduino Due was used for the TT prototyping. Four Force Sensitive Resistors (FSRs) were connected to the board's analog inputs (Figure 3a). FSRs can be used to detect pressure or weight. The basic working principle is that the resistive value of the sensor alternates depending on the amount of force that is being applied on the sensor. Silicone buffer pads[4] were manually attached to the centre the sensors, as the force resistive area of the FSRs was not reachable by the surface of the smart devices otherwise. Even though FSRs were used in this prototype, other sensors might potentially also be used alternatively, for example capacitive touch sensors.

Initially, when TI was placed on TT and the process started, a calibration phase had to be conducted, in order to cancel the weight of the device. For one second after the process initiation, TT would capture values from all four sensors and the maximum recorded value of each sensor would be used as a reference point. A LED would indicate the completion of the calibration phase.

After the calibration phase, the user was called to input the sequence on TI. While a button was being pressed, the calibrated TT sensor(s) closest to that button were recording the highest values (Figure 3). The indication used for detecting that a button was being pressed was that some of the sensors were recording values above their calibration point. The time during which higher values than the calibration point were being recorded by some of the FSRs was considered to be the pressing time. Similarly, the time between subsequent button presses was the time during which none of the sensors were recording higher values than the calibration threshold.

After a button was released, TT would estimate the button that was being pressed. To achieve this, each sensor was assigned a value, which was the average of all the values captured by that particular sensor during the button press period. Six virtual buttons had to be detected, corresponding to the buttons presented by TI. For the detection of the pressed button, the values that were assigned to each of the four sensors were used.

Algorithm 1 lists pseudocode for detecting the pressed buttons when the button ID numbers of device TI are per Figure 2, and the sensor ID numbers of device TT as per Figure 3. The input *sensor1* – *sensor4* is the value assigned on each of the sensors. Initially, the side (left or right) of the screen on which the press was on was determined by comparing the sum of the sensor values returned of each side. Once the side was determined, the proportion of the force

[4] Example of buffer pads: `https://www.amazon.co.uk/gp/product/B00P11D4VK/ref=s9u_simh_gw_i2`

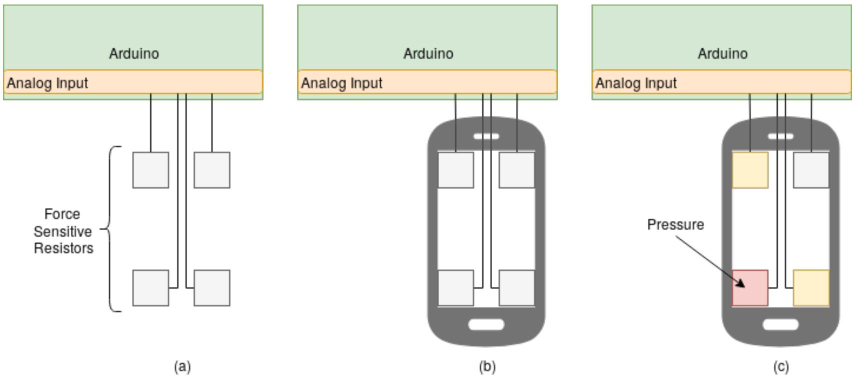


Fig. 3. Detection of Presses by TT

applied on the top sensor was calculated. The proportion was divided in three equal parts. If the proportion was between 0.66 and 1, the top virtual button ID was returned. Similarly, values between 0.33 and 0.66 corresponded to the middle virtual button, and between 0 and 0.33 to the bottom one.

When four button presses were detected by the prototype, it would return the pressed button sequence and the timings to a computer through Arduino's serial port. An application written in Python would request from the experiment operator to manually input the ID that was assigned to the particular transaction by TI. The returned values and the transaction ID would be appended to a CSV file with the same format as the one on TI's side.

Algorithm 1: Detection of Pressed Button

```
Input  : int sensor1, int sensor2, int sensor3, int sensor4
Output : int pressedButtonID
leftSide ← sensor1 + sensor2;
rightSide ← sensor3 + sensor4;
if $leftSide > rightSide$ then
    force ← sensor1 / leftSide;
    if $force \geq 0$ and $force < 0.33$ then
        return 5
    else if $force \geq 0.33$ and $force < 0.66$ then
        return 3
    return 1
end
force ← sensor3 / rightSide;
if $force \geq 0$ and $force < 0.33$ then
    return 6
else if $force \geq 0.33$ and $force < 0.66$ then
    return 4
return 2
```

Two experimental frameworks were developed, for proximity and relay attack evaluation. At the end of each phase, the stored data would be extracted form each of the devices and moved to a computer for the evaluation of the results.

6 Proximity Detection Framework

In order to evaluate the performance of the proposed solution in a proximity detection scenario, 100 transactions were performed with each of the three devices listed in Section 5. A random sequence was generated and presented by TI on each transaction. The FSRs were aligned in a set-up for the smallest device (SM-G800F), and this set-up was maintained throughout the experimental phase, regardless of the device being used. After the completion of the first phase, using each of the three TI devices, the collected data in CSV format was extracted from both transaction devices (the smartphone and the laptop on which the Arduino was connected).

The aim of the framework was to assess whether sufficient information for establishing proximity evidence is collected during the process. The results were used to set acceptable upper and lower bounds for presses and releases.

6.1 Evaluation Methodology

A Python application was developed for the data analysis process, which would compare the input sequence recorded by both devices, and calculate the difference of individual corresponding features (button press and release timings). The minimum and maximum press and release differences were detected for each of the devices, and all 300 measurements combined. The minimum and maximum press and release differences were used as limits for distinguishing genuine from relay attack transactions in the relay attack detection framework.

6.2 Results and Discussion

The results of the proximity detection framework are listed in Table 1. They refer to the time difference in milliseconds between the timings recorded by TT and TI. For example, the difference of the first press for a single transaction is calculated as:

$$Diff_{Press1} = TT_{Press1} - TI_{Press1}$$

The maximum, minimum, and the average calculated values are listed, as well as the span between the maximum and minimum observed values. The analysis has been performed for each of the three devices, as well as for the combination of all the recordings of the three (referred to as 'Total'). Negative timings denote that the measurement of a press or release by TI was longer than that of TT.

Differences among the devices were observed, likely related to their weight. However, the measured time span was approximately the same for all three devices. Therefore, a real-world deployment is recommended to take into account the device model, in order to minimise the attack window approximately by half.

Table 1. Proximity Detection Results – in ms (negative results indicate that TI's measurement durations were larger than TT's)

	Minimum		**Maximum**		**Span**		**Average**	
	Press	**Release**	**Press**	**Release**	**Press**	**Release**	**Press**	**Release**
SGS5 mini	-46	11	-11	48	35	37	-29.09	28.71
SGS 4	-28	-8	9	30	37	38	-5.32	5.26
Nexus 9	-18	-8	12	23	30	31	-2.16	2.83
Total	-46	-8	12	48	58	56	-12.19	12.27

High accuracy was also ascertained in the detection of the input pattern by device TT. Prior to the initiation of the experimental phase, the pattern detection of the smaller devices was very high. Failures occurred in a few occasions where the buttons on device TI were pressed very close to neighbouring buttons. Slightly reducing the size of the buttons effectively restricted this issue.

The detection of button presses on the tablet required longer presses than on the smartphones, which had no special input requirements. Even though perfect detection was recorded when longer button presses were applied, the use of tablets is not recommended without readjustment of the underlying FSRs.

7 Relay Attack Detection Framework

We subsequently evaluated the effectiveness of the solution as an anti-relay mechanism. We gathered 10 volunteers who attempted to attack the system in two phases, explained below. A separate Android application was developed for the purposes of the experiments, based on the application used in the proximity evaluation framework. It was presenting predefined sequences instead of random ones. Device TT's architecture did not alter between the two frameworks.

During the first attack phase, a set of videos of a person inputting a sequence (genuine user) were presented to the volunteers. The sequence timings entered by the genuine user on TI were stored in a CSV file as per Section 6 for later comparison against relay attack data. The videos were played on a large screen, and the movement of the genuine user was evident. The camera used to record the videos was placed on the left of the device, and at a 45 degree angle, in order to make the presentation of the three dimensions more clear. Prior to the initiation of the experiments, the volunteers were asked to choose the rotation and flip of the video that they preferred. A set of 10 different patterns were presented to each volunteer, who was asked to attempt to replicate/mimic the movement of the genuine user with accuracy while the video was playing, or afterwards. The same pattern was presented on both the video and the device provided to the volunteers. Also, the same device (SM-G800F) was used on the video and by the volunteers. Data from both devices was stored in separate CSV files, separately for each volunteer.

All volunteers were university students and staff who had a good understanding of security and relay attacks. Prior to the experiments, their goal and the principle on which the anti-relay mechanism was based were thoroughly explained to them. Moreover, four of the volunteers had background in playing some musical instrument (guitar, piano, or both), ranging from medium to advanced level.

During the second phase, the volunteers were asked to attempt and attack the exact same video 10 times. Meaning, a genuine user entered a sequence that was recorded, then the attacker was given 10 tries to train to replicate the sequence as close to the genuine user as possible. This tested the possibility of whether an attacker who watches the same pattern being entered a number of times, his/her potential of replication would increase. The same set-up as in the previous phase was used. In both phases the attacker was very powerful, as the input sequence was known, and there was a clear view of the genuine user's movements.

7.1 Evaluation Methodology

Measurements captured from the TI that the genuine user was using on the video were compared against measurements from device TT used by the volunteers. In a relay attack, these two devices would be the genuine devices, the other two, the devices operated by the attacker.

In order to evaluate the performance of the volunteers, two methods were used; threshold- and machine learning-based analyses. Initially, we set acceptable minimum and maximum thresholds for presses and releases, based on the results of the proximity evaluation framework (Section 6.2). The recorded presses and releases from the two devices were sequentially compared against each other. If the difference between a press or a release captured by TT and TI was within the bounds, it was considered to be acceptable. Otherwise a relay attack was detected. The point at which the inconsistency appeared was the detection point. For example, if an attacker performed the first press and the first release correctly, but the second press was out of bounds, the second press would be considered as the detection point. This part of the evaluation was conducted in two phases. We first subjected the relay attack data against the thresholds set when by all the three devices, and then against device specific thresholds (SM-G800F).

Weka [9] was used to apply a suite of well-known classifiers. The classifiers were trained on a set of feature vectors with corresponding binary labels (genuine or relayed transaction), which were collected beforehand. The trained model was used to classify subsequent transaction data streams as genuine or relayed. We tested the Random Forest, Naïve Bayes, Logistic Regression, Decision Tree (grown using the C4.5 algorithm), and Support Vector Machine with the RBF[5] kernel (the SVM w/RBF hyper-parameters, C and γ, were established using standard exhaustive grid search) classifiers. The threshold was based on the probability estimate output by the learned classification model, i.e. the estimated probability that a transaction is genuine. As genuine transactions were considered transactions collected during the proximity evaluation phase (Section 6).

[5] RBF: Radial Basis Function.

Table 2. Threshold-Based Relay Attack Detection Results

	Press 1	**Release 1**	**Press 2**	**Release 2**	**Press 3**	**Release 3**	**Press 4**	**False Accept**
				General Threshold – Phase 1				
Detected	73	24	0	2	1	0	0	0
Correct	27	10	37	10	24	5	31	—
				Device Specific Threshold – Phase 1				
Detected	84	16	0	0	0	0	0	0
Correct	16	7	25	9	9	4	23	—
				General Threshold – Phase 2				
Detected	57	37	5	1	0	0	0	0
Correct	43	11	29	24	26	9	20	—
				Device Specific Threshold – Phase 2				
Detected	65	33	2	0	0	0	0	0
Correct	35	6	21	22	20	6	9	—

7.2 Results and Discussion

A total of 200 relay transactions were evaluated. The results of both threshold- and machine learning-based analyses are presented in this section.

Evaluation 1: Threshold-Based. The results of the threshold-based analysis are listed in Table 2. *'General Threshold'* refers to the threshold set by combining the proximity results of all three tested devices, while *'Device Specific Threshold'* to the threshold set by SM-G800F, as it was the device used for the relay attack detection evaluation. *'Detected'* refers to the percentage of relay attempts detected at a particular press or release, because the attackers failed to accurately replicate the movement of the genuine user at that point. *'Correct'* refers to the percentage of times that a single press or release was successfully replicated by the attacker. Finally, the first phase refers to the user trying to attack a different video on each try, and the second phase, the attacker trying to attack the same video 10 times.

None of the volunteers attempted to successfully attack the system, using threshold-based analysis. Moreover, the volunteers with background in playing a musical instrument did not present improvements over the rest of the user. However the sample was limited, so further investigation is required.

The best attempt is presented in Figure 4. Device TI′ in the figure represents the measurement recorder by the transaction instrument that the volunteer was using. It illustrates the corresponding proximity transaction, for which all presses and releases were within bounds. The relay attack was detected on the third press, using the threshold set by the combination of the three devices. Using device specific threshold, the attack was successfully detected on the first release.

The majority of the transactions were detected on the first press and release, using both general, and device specific thresholds, even when the volunteers attempted to attack the same video multiple times (second phase). During the second phase, a small incline was observed in the user performance as the same video was being attacked multiple times, most evident in the performance of releases. Figure 5 depicts that incline, along with the average, maximum, and minimum performance of each round's presses and releases.

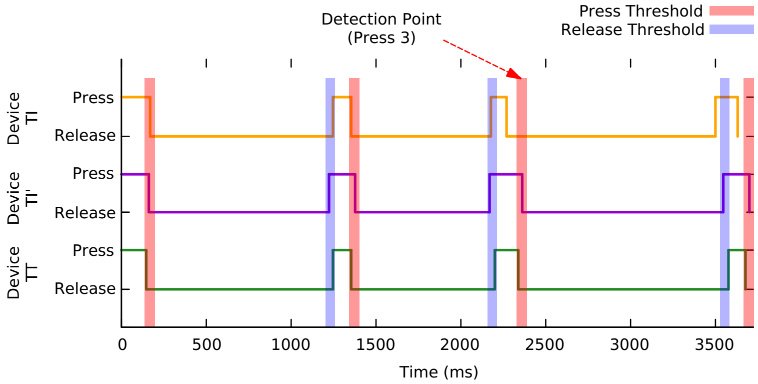


Fig. 4. Graphical Representation of the Best Attack Attempt (TT – TI) Versus the Corresponding Genuine Transaction (TT – TI′)

Evaluation 2: Machine Learning-Based. The results of the machine learning-based analysis, obtained by repeating stratified 10-fold cross-validation, are presented in Table 3. A training set of 300 genuine and 200 relay attack transactions was used. The default settings of each algorithm were used on Weka. The metrics listed are the classification accuracy *(Accuracy)*, the Area Under the Receiver Operating Characteristic (ROC) Curve *(AUC)*, the *F1-score*, and the Equal Error Rate *(EER)*.

Perfect classification was achieved by using the Support Vector Machine classifier. Near perfect classification (> 98%) was achieved by the Random Forest, Naïve Bayes, and Decision Tree classifiers, with best performance observed by the first three (> 99.5% accuracy). The Random Forest algorithm failed to accurately classify two relay transactions, the Naïve Bayes one, and the Decision Tree six.

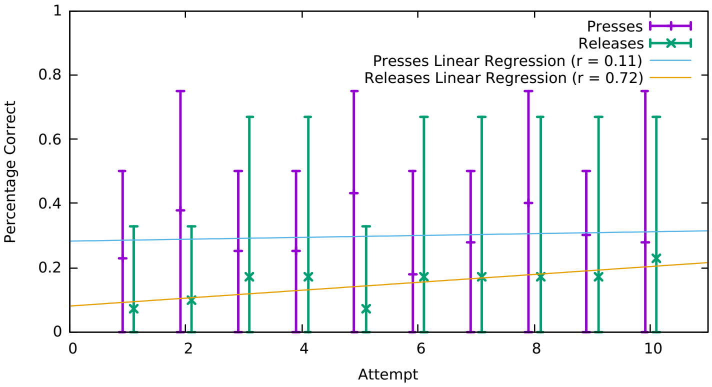


Fig. 5. Performance Variation of Each of the 10 Attempts in the Second Phase

Table 3. Machine Learning Classification Results Obtained by Repeating 10-Fold Cross-Validation 10 Times

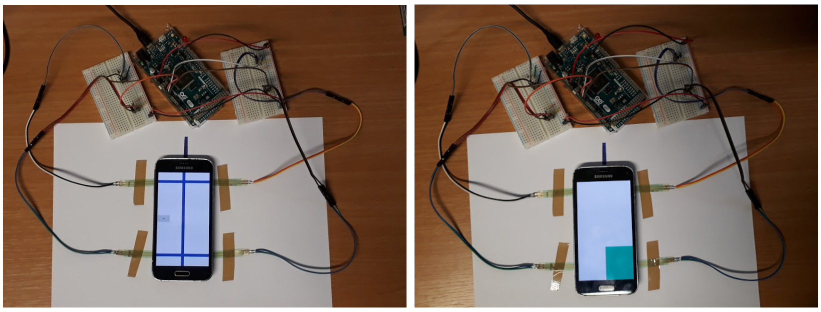
	Random Forest	Naïve Bayes	Logistic Regression	Decision Tree	Support Vector Machine
Accuracy (%)	99.62	99.80	86.58	98.78	100.00
AUC	0.9999	0.9996	0.8163	0.9873	1.0
F1-score	0.9969	0.9984	0.90	0.9901	1.0
EER	0.0022	0.0047	0.1993	0.104	0.0

8 Discussion and Outcome

The analysis of the experimental data indicated that the effectiveness of the proposed solution as a PRAD mechanism was high. However, some usability concerns might arise, especially for visually impaired users, and users with motor difficulties. Compared to the majority of previous works, additional steps are required by the user, including the correct placement of device TI on TT.

Since the positioning of device TI on TT is important for the later to accurately detect the ID of the pressed button, device-specific positioning lines were presented on the TI's display prior to the process initiation. This assisted in more equally dividing the mass across the four FSRs. The model of the device could be determined through API calls, which was used to display the correct positioning. It should be stressed that the positioning precision had to be performed with only some degree of accuracy. The detection was found to be very accurate even without perfect placement of the devices. Figure 6 depicts the guidelines and the pattern input interface on a SM-G800F device.

Failure to click on the correct button (mistapping) will also lead to inconsistencies between the captured data of the two devices, so the process will have to be restarted. Moreover, a vibration of the phone during the process will cause inconsistencies, so they should temporarily be disabled, until the completion of the process. Finally, the performance of the solution might also degrade if phone



(a) Device Calibration (b) Pattern Entry

Fig. 6. The Evaluation Test-bed

cases are used on TI, or the device's camera is located above the point where an FSR should touch. However, significant advantages over previous works exist, so depending on the deployment scenario this technique might be preferable.

The relay attack detection rate was empirically found to be higher than previously proposed solutions [12, 22, 26, 29]. The detection rate can potentially improve even further by considering more features, like the amount of pressure detected by the two devices. Even though this is possible on TT, at the moment the majority of smartphones are not capable of accurately measuring the amount of pressure applied on their screen. Some Android devices estimate the amount of pressure through the number of pixels being covered on the screen by the user's finger. This technique is not very accurate, and it is also not available on all devices. The feature is not supported by the GT-I9500. On the other two devices, the accuracy was highly dependant on the finger orientation rather than the amount of pressure being applied on the screen. Moreover, inconsistencies were observed among the values recorded by the two devices. We concluded that the technology was not mature enough to provide quality data.

Moreover, no additional or non-standard hardware on the TI side is required, like in many of the previously proposed solutions, as in [10], and [26]. Many of the previously proposed solutions might also be vulnerable in the presence of an attacker with context manipulating capabilities [27]. Since this solution is not dependant upon the surrounding environment, such attacks do not apply, unless the attacker physically tampers with the devices.

9 Conclusion and Future Directions

Communicating devices that require to operate within proximity 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 the off-the-shelf attacker 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 presence of an attacker with context manipulating capabilities. In this work we presented a novel approach for Proximity and Relay Attack Detection (PRAD) by using bidirectional sensing and comparing of subsequent button presses and releases by the transaction devices.

A test-bed was designed and built for the evaluation of the proposed solution as a PRAD mechanism. Initially, the effectiveness in proximity detection was examined. Afterwards, the test-bed was subjected against a set of volunteers who attempted to attack the system. All the attack attempts were successfully detected through threshold-based analysis. Moreover, perfect classification was achieved by using the Support Vector Machine classifier. Classification accuracy of up to 99.8% was achieved by other well-known machine learning classifiers.

As part of our ongoing investigation, we are planning to conduct a more extensive user study. We are also planning to examine the use of additional features that would further minimise the potential of an attacker to perform a relay attack, like pressure intensity.

References

1. Transit and Contactless Open Payments: An Emerging Approach for Fare Collection. White paper, Smart Card Alliance Transportation Council (November 2011)
2. How to Optimize the Consumer Contactless Experience? The Perfect Tap. Tech. rep., MasterCard (2014)
3. EMV Contactless Specifications for Payment Systems: Book A - Architecture and General Requirements. Spec V2.6, EMVCo, LLC (April 2016)
4. Transactions Acceptance Device Guide (TADG). Specification Version 3.1, VISA (November 2016)
5. Digital Payments Solutions Industry Considerations. Online report, The UK Cards Association (June 2017), `http://www.theukcardsassociation.org.uk/wm_documents/Digital%20Wallets%20-%20Industry%20Considerations%20Outline.pdf`
6. Francis, L., Hancke, G., Mayes, K., Markantonakis, K.: 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. pp. 35–49. RFIDSec'10, Springer-Verlag, Berlin, Heidelberg (2010)
7. Francis, L., Hancke, G., Mayes, K., Markantonakis, K.: Practical NFC Peer-to-Peer Relay Attack Using Mobile Phones. In: Radio Frequency Identification: Security and Privacy Issues, LNCS, vol. 6370, pp. 35–49. Springer (2010)
8. Francis, L., Hancke, G.P., Mayes, K., Markantonakis, K.: Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones. IACR Cryptology Archive 2011, 618 (2011)
9. Frank, E., Hall, M.A., Witten, I.H.: The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". Morgan Kaufmann, Burlington, MA, 4 edn. (2016)
10. Gurulian, I., Akram, R.N., Markantonakis, K., Mayes, K.: Preventing Relay Attacks in Mobile Transactions Using Infrared Light. In: Proceedings of the Symposium on Applied Computing. pp. 1724–1731. SAC '17, ACM, New York, NY, USA (2017)
11. Gurulian, I., Shepherd, C., Frank, E., Markantonakis, K., Akram, R., Mayes, K.: 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. TrustCom '17, IEEE (August 2017)
12. Halevi, T., Ma, D., Saxena, N., Xiang, T.: Secure Proximity Detection for NFC Devices Based on Ambient Sensor Data. In: Foresti, S., Yung, M., Martinelli, F. (eds.) Computer Security – ESORICS 2012. LNCS, Springer (2012)
13. Hancke, G.P.: 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). pp. 91–96 (Nov 2012)
14. Hancke, G.P.: Practical Attacks on Proximity Identification Systems (Short Paper). In: IEEE Symposium on Security and Privacy. pp. 328–333. IEEE Computer Society (2006), `http://dblp.uni-trier.de/db/conf/sp/sp2006.html#Hancke06`
15. Hancke, G.P., Kuhn, M.G.: An RFID Distance Bounding Protocol. In: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks. pp. 67–73. SECURECOMM '05, IEEE Computer Society, Washington, DC, USA (2005)
16. Hancke, G., Mayes, K., Markantonakis, K.: Confidence in Smart Token Proximity: Relay Attacks Revisited. Computers & Security 28(7), 615 – 627 (2009), `http://www.sciencedirect.com/science/article/pii/S0167404809000595`

17. Jin, R., Shi, L., Zeng, K., Pande, A., Mohapatra, P.: MagPairing: Pairing Smartphones in Close Proximity Using Magnetometers. IEEE Transactions on Information Forensics and Security 11(6), 1306–1320 (June 2016)
18. Kfir, Z., Wool, A.: 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. pp. 47–58. IEEE (2005)
19. Ma, D., Saxena, N., Xiang, T., Zhu, Y.: Location-aware and safer cards: Enhancing rfid security and privacy via location sensing. IEEE TDSC 10(2), 57–69 (2013)
20. Madlmayr, G., Langer, J., Kantner, C., Scharinger, J.: NFC devices: Security and privacy. In: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on. pp. 642–647. IEEE (2008)
21. Mehrnezhad, M., Hao, F., Shahandashti, S.F.: Tap-Tap and Pay (TTP): Preventing Man-in-the-Middle Attacks in NFC Payment Using Mobile Sensors. Tech. Rep. CS-TR-1428, Newcastle University (July 2014)
22. Mehrnezhad, M., Hao, F., Shahandashti, S.F.: 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)
23. Roland, M., Langer, J., Scharinger, J.: Relay Attacks on Secure Element-Enabled Mobile Devices, pp. 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg (2012), `http://dx.doi.org/10.1007/978-3-642-30436-1_1`
24. Roland, M., Langer, J., Scharinger, J.: Applying relay attacks to Google Wallet. In: Near Field Communication (NFC), 2013 5th International Workshop on. pp. 1–6 (Feb 2013)
25. Shepherd, C., Gurulian, I., Frank, E., Markantonakis, K., Akram, R., Mayes, K., Panaousis, E.: The Applicability of Ambient Sensors as Proximity Evidence for NFC Transactions. In: Mobile Security Technologies, IEEE Security and Privacy Workshops. MoST '17, IEEE (May 2017)
26. Shrestha, B., Saxena, N., Truong, H.T.T., Asokan, N.: Drone to the Rescue: Relay-Resilient Authentication using Ambient Multi-sensing. In: Financial Cryptography and Data Security, pp. 349–364. Springer (2014)
27. Shrestha, B., Saxena, N., Truong, H.T.T., Asokan, N.: Contextual proximity detection in the face of context-manipulating adversaries. CoRR abs/1511.00905 (2015), `http://arxiv.org/abs/1511.00905`
28. Trujillo-Rasua, R., Martin, B., Avoine, G.: The Poulidor distance-bounding protocol. In: Radio Frequency Identification: Security and Privacy Issues, pp. 239–257. Springer (2010)
29. Truong, H.T.T., Gao, X., Shrestha, B., Saxena, N., Asokan, N., Nurmi, P.: Using contextual co-presence to strengthen Zero-Interaction Authentication: Design, integration and usability. Pervasive and Mobile Computing 16, Part B, 187 – 204 (2015), `http://www.sciencedirect.com/science/article/pii/S1574119214001771`, Selected Papers from the Twelfth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2014)
30. Truong, H.T.T., Gao, X., Shrestha, B., Saxena, N., Asokan, N., Nurmi, P.: Comparing and Fusing Different Sensor Modalities for Relay Attack Resistance in Zero-Interaction Authentication. In: Pervasive Computing and Communications, 2014 IEEE International Conference on. pp. 163–171. IEEE (2014)
31. Umar, A., Mayes, K., Markantonakis, K.: Performance Variation in Host-Based Card Emulation Compared to a Hardware Security Element. In: Mobile and Secure Services, 2015 First Conference on. pp. 1–6. IEEE (2015)

32. Urien, P., Piramuthu, S.: Elliptic curve-based RFID/NFC authentication with temperature sensor input for relay attacks. Decision Support Systems 59, 28 – 36 (2014)
33. Varshavsky, A., Scannell, A., LaMarca, A., de Lara, E.: Amigo: Proximity-Based Authentication of Mobile Devices. In: Krumm, J., Abowd, G., Seneviratne, A., Strang, T. (eds.) UbiComp 2007, pp. 253–270. LNCS, Springer (2007)
34. Verdult, R., Kooman, F.: Practical Attacks on NFC Enabled Cell Phones. In: Near Field Communication (NFC), 2011 3rd International Workshop on. pp. 77–82 (Feb 2011)