Preventing Relay Attacks in Mobile Transactions Using Infrared Light

Iakovos Gurulian, Raja Naeem Akram, Konstantinos Markantonakis, Keith Mayes
Information Security Group, Smart Card Centre
Royal Holloway, University of London
Egham, Surrey, UK, TW20 0EX
{iakovos.gurulian.2014, r.n.akram, k.markantonakis, keith.mayes}@rhul.ac.uk

ABSTRACT

Near Field Technology (NFC) enables a smartphone to emulate a smart card, enabling it to provide services, like banking and transport ticketing. Similar to smart cards, NFC-based transactions are susceptible to relay attacks. Distance bounding protocols have been proposed for smart cards to counter relay attacks. However, this may not be effective in the field of mobile transactions, due to their requirement of high time-delay sensitivity and specialised hardware. A number of proposals are being put forward that show that sensing the natural ambient environment is an effective anti-relay mechanism. Existing literature neither involves a threat actor in their analysis nor they are in compliance with EMV's transaction requirement of $500ms$. In this paper, we look at the anti-relay mechanism from a different point of view. Instead of measuring the natural ambience, we generate and measure a unique artificial ambient environment (AAE) using peripherals of the devices involved in a transaction. To evaluate our proposal and its effectiveness, we selected infrared from the proposed set of off-the-shelf actuator/sensor pairs available on modern smartphones. We designed and deployed six distinct test-beds, each based on a unique method of relay attack, in order to evaluate the effectiveness of our proposal in the context of infrared. From our experimentations, we can empirically state that infrared showed high success rate in relay attack detection – higher than any existing work in academic literature.

CCS Concepts

•**Security and privacy** → **Authorization;** *Domain-specific security and privacy architectures;* Multi-factor authentication; Access control;

Keywords

Mobile Payments; Relay Attacks; Artificial Ambient Environment; Contactless; Infrared; Experimental Analysis

SAC 2017, April 03-07, 2017, Marrakech, Morocco

DOI: http://dx.doi.org/10.1145/3019612.3019794

1. INTRODUCTION

Near Field Communication (NFC) [7] allows smartphones to emulate contactless smart card functionality, enabling them to provide smart card services, like banking and transport ticketing. Such services have already been launched by leading technology firms, like Google and Apple.

An attacker can use a relay attack to gain access to services that a legitimate user is eligible for, such as payments or access to buildings. Both smart card and smartphone based contactless transactions are susceptible to relay attacks. Distance bounding protocols have been proposed as a countermeasure in the case of smart cards [6, 8–10, 14, 22]. Distance bounding protocols however may not be applicable in the case of NFC-enabled smartphones due to the multitude of specialised hardware, and the high variability in time delay during task execution [7, 13]. Sensing the natural ambient environment, using ambient sensors, has been proposed instead as a strong candidate of proximity/relay attack detection [9, 13, 18, 20, 25–27]. However, most of the proposed methods do not adhere to industry requirements in domains like payments and ticketing. Previous work has shown that sensing the ambient environment may not provide adequate information regarding the proximity of two devices, within the operational time frame of $500ms$ [5].

We are proposing the generation and measurement of a unique artificial ambient environment (AAE) as a means of proximity/relay attack detection, using the peripherals of the devices involved in a transaction. A set of potential AAE actuator/sensor pairs is proposed that are widely available on modern smartphones. We selected infrared from that set, for further evaluation. Six test-beds were deployed, using different methods for attacking the proposed system. High success rate was observed during the experimental phase, regarding relay attack detection.

This work focuses on protecting against the off-the-shelf attacker. There are limitations in the amount that can be spent in a contactless transaction, for example in the UK is currently set at £30 per transaction [24]. In the case of more security critical scenarios, like access to governmental buildings, smartphones may not provide adequate amount of security [15].

The primary contributions of this paper include:

- Artificial Ambient Environment: We proposed a generic framework for generation and measurement of the AAE. In addition, this generic framework defines the methodology of how two devices can ascertain whether they are in proximity to each other. Discussion of this framework is in Section 3.

- Infrared-Based AAE: We listed possible AAE actuators, widely available on modern smartphones (Section 3.1.2). Taking the generic framework, we designed and built an evaluation test-bed, using infrared as an AAE generator (Section 4).
- Effectiveness Analysis: We empirically evaluated the effectiveness of using infrared as an AAE generator for proximity/relay attack detection, against six distinct relay attack scenarios (Section 5). The experimental results indicated a high success rate in both proximity, and relay attack detection.

2. CONTACTLESS MOBILE TRANSACTIONS AND RELAY ATTACKS

To initiate an NFC-based mobile transaction, the NFC device is brought within the radio frequency range of a transaction terminal (<3-5cm distance). The contactless transaction might require some form of authentication or authorisation, like a Personal Identification Number (PIN) code or a biometric, although this is not always required [4]. However, authentication and/or authorisation are not effective countermeasures to relay attacks, as in the case of the Mafia fraud attack [8].

A relay attack [11, 12, 28] is a passive man-in-the-middle attack during which an attacker is extending the communication distance of two legitimate devices by relaying each communication message between them, without the legitimate users' consent. This way the attacker can be facilitated to access legitimate user's services.

For a smartphone-based contactless payment, an attacker would have a malicious (compromised) payment terminal and payment instrument. The malicious payment terminal should be presented to a legitimate user and the malicious payment instrument to a genuine payment terminal. Data communicated during a transaction would be relayed between the attacking devices, as shown in Figure 1.

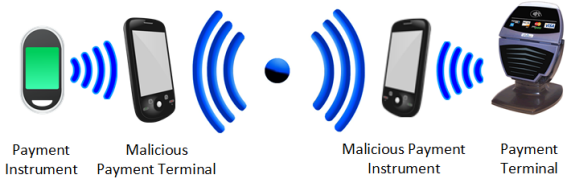

Figure 1: Relay Attack

Relay attack detection and prevention requires information regarding the coexistence of the devices involved in a transaction. As mentioned in Section 1, distance bounding protocols available for smart cards may not work in the field of smartphones. Several methods have been proposed regarding relay attack prevention on NFC-enabled smartphones. Many of these methods attempt to detect device coexistence during NFC transactions using the environmental (ambient) sensors that are present in modern smartphones (further discussion regarding related work in Section 6.1).

3. ARTIFICIAL AMBIENCE FOR PROXIMITY DETECTION (AAPD)

In this section the theoretical foundation of the proposed framework is described and analysed.

3.1 Proposed Framework

Ambient sensors in smartphones measure a particular environmental or physical property of the immediate surrounding of the smartphone, like the light intensity, the ambient sound, or the device's acceleration. Existing literature, discussed in Section 6.1, argues two devices using ambient sensors at the same time, for a short duration of time, can provide proof of proximity.

Majority of the previous work however does not comply with the EMV transaction requirement to complete in $500ms$ [2–4, 16]. According to [5], data captured in up to $500ms$ by any ambient sensor on off-the-shelves smartphones cannot effectively provide proof of proximity.

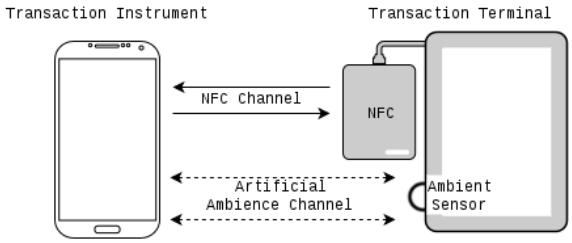

Figure 2: Framework Architecture

We propose the generation of artificial ambient environments (AAEs) by using the smartphone's peripherals, like the speaker or the phone's vibration. The aim is to increase the irreproducibility of ambient environment to thwart the relay attacks and establish strong proximity proof.

During an NFC transaction, one (unidirectional) or both (bidirectional) devices involved in the transaction will generate (be actuators) and/or measure (be sensors) the AAE, forming a second communication channel (Figure 2). This second channel, referred to as Artificial Ambience Channel (AAC), will be operating in parallel to the NFC channel, for some predefined time.

Smartphones available in the market are not the same in performance capabilities, operating system (OS), and/or available ambient sensors. Depending on a device's capabilities, the two devices involved in a transaction can negotiate the tuple $\{\mathcal{A}, \mathcal{S}, \mathcal{T}_{Sent}, \mathcal{M}\}$ that will be used in the transaction. $\mathcal{A}$ denotes the AAE actuator that will be used in the transaction. $\mathcal{S}$ denotes the sensor which will measure the AAE. $\mathcal{T}_{Sent}$ (where $\mathcal{T}_{Sent} \leq 500ms$), denotes the time for which the AAE will be active. Finally, $\mathcal{M}$ denotes whether the AAE generation will be unidirectional or bidirectional.

Upon the completion of the AAE measurement by the devices involved in the transaction, a comparison between the transmitted and measured data will take place either by one or both devices, or by trusted third party. Transmission of the captured measurement will be done in a secure manner, integrated in the transaction protocol running on the NFC channel (for on-device verification), or through a secure network channel (for verification by third party). Such an integration is beyond the scope of this paper.

During the comparison phase, only data exchanged through the AAC while the AAE was active will be regarded ($\mathcal{T}_{Received}$, where $\mathcal{T}_{Received} = \mathcal{T}_{Sent}$). Thus, during a relay attack AAE information will have to be successfully relayed within the $\mathcal{T}_{Sent}$ time frame, or they will be discarded from the comparison phase.

For an AAE generation method to be successful, the result of the comparison should provide adequate proof of proximity. Therefore, the AAE should be difficult to be accurately replicated by an attacker in a relay attack.

In this paper, only the properties of the AAC channel are investigated. This is to empirically establish whether AAE can provide an effective proof of proximity for relay attack countermeasures and two-factor authentication, similar to [17]. To investigate the proposed solution, test-beds were developed using off-the-shelf and custom hardware.

3.1.1 Architecture Requirements

In order for a generated AAE to be suitable for relay attack prevention, the following requirements should be met:

- The generation of the AAE should be based on some random streams/sequences.
- The AAE generation and the establishment of proximity evidence should be easy for a genuine pair.
- The generated AAE should be hard to relay/reproduce to a remote location, without detection.

3.1.2 Candidate Architecture

Peripherals widely available on smartphones that can potentially be used to generate an AAE are: 1. Infrared, 2. Camera's flash light, 3. Vibration, 4. Speaker (sound), 5. Bluetooth, 6. Device's display, and 7. Camera

In this paper, we are focusing on Infrared transmitters, that are widely available in different smartphones[1], mainly for allowing users to control home equipment, like televisions [1].

4. INFRARED AS AAPD

Infrared is a form of electromagnetic radiation with a wavelength outside of the visible spectrum for the human eye. Infrared emitters exist in a wide range of modern smartphones, and their main purpose is for controlling home equipment (e.g. televisions). In our evaluations we used Android handsets, as Android provides an API for transmitting infrared signals [1].

In this section, we describe the architecture and deployment of the test-bed for evaluation of infrared AAPD and its effectiveness against six different relay attack techniques.

4.1 Test-bed Architecture – Infrared

Figure 3 depicts the architecture of the test-bed for evaluation of infrared as an AAE. Two scenarios were regarded. First, a transaction between the mobile device TI, and the transaction terminal TT' (referred to as TI-TT'). This scenario aims to evaluate the proximity detection capabilities of the framework, and no relay is involved between the pair. Second, a transaction between the mobile device TI, and the transaction terminal TT (referred to as TI-TT). In this scenario, the pair TT'-TI' is attempting to relay the infrared generated AAE. Figure 4 shows a more clear representation of the devices involved in the two scenarios.

4.1.1 Proximity Scenario

In the first scenario, upon the initiation of the transaction, TI begins emitting a sequence consisting of 500 random pre-generated bits through its infrared emitter. The transaction terminal, TT', captures the transmitted sequence. After the completion of the transaction the transmitted and captured bits are compared for similarity by a trusted third party (during the course of our experiments, a laptop running the comparison software). The comparison step can also potentially be performed by one of the devices involved in the transaction, as mentioned in Section 3, but this is out of the scope of this work.

[1]Devices with infrared emitters: https://goo.gl/4tVORN

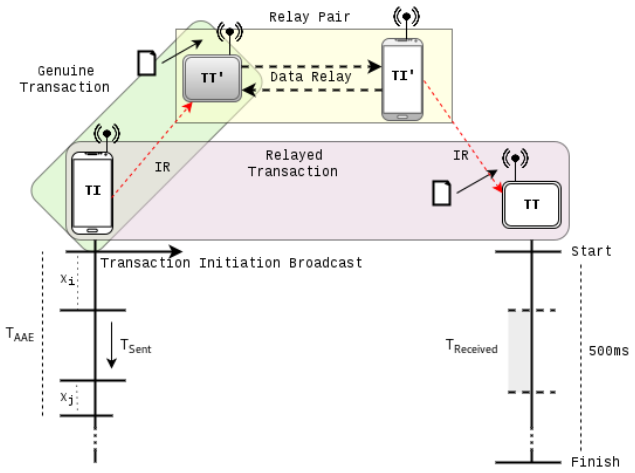

Figure 3: The Evaluation Framework

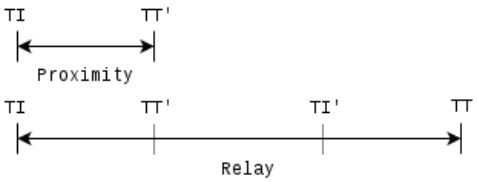

Figure 4: Test-bed Scenarios

The infrared emitter is technically a LED that emits electromagnetic radiation at the infrared spectrum. In order to transmit the 500 bits through it, they first had to be converted to pulses and pauses. After experimentation, we set that one bit is represented by a 200μs pulse (emission) or pause (no emission). Although infrared at 38.4kHz frequency (used during our experiments) can theoretically emit one bit per 13μs, we could effectively emit one bit in no less than 200μs, using the available mobile devices.

Sub-sequences of *ones* and *zeros* from the random 500-bit sequence are encoded into *pulse* and *pause* durations (e.g. a sub-sequence of two consecutive *ones* will be encoded in a 400μs pulse, a sub-sequence of three consecutive *zeros* by a pause of 600μs, and so on). The 500-bit sequence requires 100ms of transmission through the infrared emitter ($\mathcal{T}_{Sent} = 100ms$). Figure 5 represents the conversion of the bit-sequence "1101110011" into pulse and pause timings.

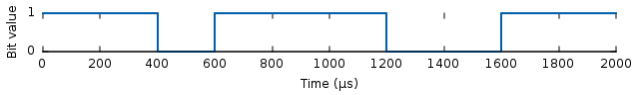

Figure 5: Time Representation of Bit-Sequence "1101110011"

An infrared receiving diode capable of sensing infrared radiation, is attached to the transaction terminal, and accessed through software in order to time the pulse and pause sequences. The terminal listens for infrared signals for the whole period of the transaction. Intercepted pulse and pause timings are then decoded into a bit sequence. In case the data comparison should take place on one of the devices involved in the transaction, the infrared listening period can be reduced, according to the requirements of the desired transaction protocol. However, integrating the proposed framework in a particular protocol is beyond the scope of this paper.

As mentioned in Section 3, some time may be required before an AAE actuator can be initialised. In the case of the Android devices used in our experiments, through experimentation we concluded that time x_i was required before a device could start emitting infrared, and time x_j was required for the process to shut down, after emission had stopped. Since we did not have low level access to the API calls, we were not able to accurately calculate these delays. Modifying the Android operating system (OS) was also not an option, as after inspection of the OS's source code we believe that both delays are caused by the proprietary infrared drivers of the devices. Because of the infrared Android API, we were only able to know the total time of the infrared transmission process (referred to as $\mathcal{T}_{AAE}$). However, after analysis of 150 runs of the infrared transmission process, we were able to calculate x_j to be equal to $28 \pm 2ms$. Therefore, since $\mathcal{T}_{AAE}$ can be measured by an application, and $\mathcal{T}_{Sent}$ and x_j are known, Eq. 1 provides x_i.

$$x_i = \mathcal{T}_{AAE} - \mathcal{T}_{Sent} - x_j \quad (1)$$

During the comparison phase, any data received by the transaction terminal prior to time x_i (compared to the initiation of the transaction) or after time $x_i + \mathcal{T}_{Sent}$ is discarded. This time frame is referred to as $\mathcal{T}_{Received}$. This aids towards the avoidance of relay attacks through replay (an attacker having captured the whole sequence transmitted by the legitimate mobile device and replaying it to a remote transaction terminal). Taking into account the $4ms$ window (equivalent to 10 bits) introduced by x_j:

$$x_i - 2ms \leq \mathcal{T}_{Received} \leq x_i + \mathcal{T}_{Sent} + 2ms \quad (2)$$

The runtime of $\mathcal{T}_{Received}$ is $\mathcal{T}_{Sent} + 4ms$. The $4ms$ window can potentially be eliminated by enhancements in the Android API and the device drivers. As mentioned earlier, this is not possible without access to the driver source code.

4.1.2 Relay Scenario

In the second scenario shown in Figure 4, the same principles apply. The difference is that in this case, the malicious pair TT'-TI' is introduced. The malicious terminal TT' is capturing infrared signals transmitted by the genuine mobile TI, forwards them to TI', which plays them towards TT. The challenge for the attacker is to be able to correctly relay bits in no more than $200\mu s$ (the length of one bit) from the time of receiving them. Delay in transmission of a few bits will lead to introduction of new bits, since pauses or pulses will last longer. Also caching for longer than the $4ms$ window explained above will cast the attack detectable, since not enough bits will be captured within the time frame $\mathcal{T}_{Received}$. Caching however may cause an overhead, since it requires data to be pushed in and popped from a buffer, that may render it inefficient for such short time frame. Moreover, as already mentioned, the $4ms$ time frame may be reduced by enhancing the Android API and device drivers.

4.2 Data Collection

During our experiments, device TT' was acting as both a legitimate and a relay terminal. Thus, at the same time it was saving and relaying to TI' infrared signals. A more thorough comparison was possible this way, since we were able to juxtapose data captured by TT' and TT, when compared against the same bit-sequence transmitted by device TI. On both devices, captured infrared pulse and pause timings (in μs) were saved in different log files for each transaction, along with the time (in μs) at which the first bit was received, compared to the initiation of the transaction. Device TI was appending the random bit stream, along with time $\mathcal{T}_{AAE}$, in a `.csv` file. Calculation of x_i (as Eq. 1), and consequently of the bounds of $\mathcal{T}_{Received}$ were thus possible. These files were used during the evaluation phase, described in Section 5. After the completion of a transaction, device TI would generate a new random-bit sequence, to be used in the next transaction.

All the devices involved in the experiments were connected to a single, dedicated Wi-Fi network. The indication of the transaction initiation was performed through a network broadcast, transmitted by TI. Assessment of the NFC channel is out of the scope of this paper, and an NFC initiated relay would require tapping on both TT' and TT simultaneously in order to ensure the maximum efficiency of the attacker. Such task may be hard to accomplish, even in a controlled environment. Initiating the transaction through a network broadcast satisfies this requirement. However, in a realistic scenario, the transaction initiation would be accomplished through the NFC channel, and an attacker would have to overcome the additional barrier of synchronising the tapping of TI' with that of TI.

In the next section we discuss different test-bed variant deployments that were used to evaluate our proposal.

4.3 Deployed Test-beds with Relay Attack Variants

In order to assess the effectiveness of the framework against relay attacks, six different variants of the test-bed were deployed, based on potential relay attack configurations.

In all experiments, a Samsung Galaxy S5 mini (SM-G800F, running Android 5.1.1) was used as device TI. Raspberry Pi 3 computers (4x1.2GHz CPU and 1GB RAM), running Raspbian Jessie were used as devices TT' and TT. The Raspberry Pi (RPi) was a good candidate for acting as a terminal device, as it is equipped with 40 General Purpose Input Output (GPIO) pins, making it possible to easily build prototypes. In the case of the device TT', except from listening for infrared signals, it was also relaying them towards device TI' (except in Test-beds 5 and 6 described below, where TT' and TT were running the same application).

4.3.1 Test-bed 1: Mobile Based

In the first scenario, a mobile device (Samsung Galaxy S4 – GT-I9505, running Android 5.0.1) was used as device TI'. Timings of detected infrared pulses and pauses were transmitted from TT' to TI' through the Wi-Fi channel. In

the case of a pulse, TI' would use the `ConsumerIrManager` class [1] (Android infrared API). In the case of a pause, TI' would wait for the indicated time, before playing the next available pulse.

4.3.2 Test-bed 2: RPi Based (no caching)

Using a mobile device for infrared emission was proven to be inefficient, due to severe delays at transmitting any data via the infrared emitter (further discussion in Section 5). A custom built RPi-based relay device was used as device TI' in this scenario instead. An infrared emitting LED was connected to the GPIO pins of the RPi, and controlled by a program written in the C language (for performance reasons) using the GPIO access C library *WiringPi*[2]. Device TT' was used for relaying infrared data to TI' through the wireless network. Unlike in the previous scenario, whenever infrared state alteration was perceived (switch from pulse to pause, or the opposite), TT' would send a UDP packet to TI'. The application built for TI' would listen for UDP packets from TT', and upon packet arrival it would switch the state of the attached infrared LED, attempting to replicate the sequence played by TI.

The same libraries and programming tools as in this scenario were also used in test-beds 3 and 4.

4.3.3 Test-bed 3: RPi Based (caching – TI' side)

Because there are inconsistencies in the packet delivery time through the network, using the method described in Test-bed 2, extra bits might be introduced or subtracted from the infrared sequence played by TI', depending on the delay of subsequent packets. In order to reduce the number of packets that have to be transmitted, therefore reducing the number of potential added/removed bits, an attacker can cache data and relay in chunks. As mentioned in Section 4.1 , due to x_j, an attacker has a window of approximately $4ms$ during which data can be cached.

In this scenario, TT' would transmit pulse and pause timings, upon alteration of the perceived infrared state. Upon receiving by TI', the pulse-pause timings would be pushed in a stack (array), and $4ms$ after the arrival of the first packet, TI' would initiate playing them. Using a timer, TI' then plays the received sequence, altering the state of the infrared emitter as per the timings received by TT'.

Delays on the network may cause inconsistencies in the delivery time between two packets. In order to minimise the amount of inconsistencies in the bit stream, delays were handled. In case the next packet arrives while the data received by the previous packet is still being emitted, its data is cached by TI' and played after the completion of the current instructions. In case the next packet is delayed, the state of the infrared will not change, until new instructions have arrived.

4.3.4 Test-bed 4: RPi Based (caching – TT' side)

In this scenario, TT' caches infrared state alterations for $4ms$ before forwarding to TI'. Although this scenario is similar to the previous, the amount of generated network traffic, which was concluded to be a bottleneck to the relay process, is reduced, since data packets are relayed in $4ms$ buckets instead of per infrared state alteration. The cached information forwarded to TI' indicate the pulse-pause sequence timings that should be played by TI', in μs. The same method as in the previous scenario was used for playing the infrared sequences, caching data that arrive early and altering the state of the infrared LED when the stack is empty.

4.3.5 Test-bed 5: Infrared Extender Based

Wireless remote control extenders are available in the market. These are devices that come in pairs and convert infrared signals to radio-waves (receiver side) and back to infrared (transmitter side). The typical use of such devices is to control infrared remote controlled home equipment from a distance (e.g. from another room). This is achieved by repeating or relaying infrared sequences transmitted by the remote control.

Marmitek Powermid XL, a relaying wireless remote control extender, was used during the evaluation phase. According to the manufacturer, the range of the product is up to 100m (free field) or up to 25m (through walls). After inspection of the product's circuit, the receiving device appears to be applying current to an antenna through an infrared receiving diode. Therefore, the circuit remains open when no infrared is detected, and it closes, making it transmit at 433.39MHz frequency, when pulses at the infrared spectrum are detected. On the other side, when the receiving antenna captures a signal at 433.39MHz, a circuit is closed, powering three infrared emitters. The product was successfully tested for controlling home appliances from a distance.

4.3.6 Test-bed 6: Random Generation Based

In the last test-bed scenario, a different random sequence was generated and played by a second device. The same device as in Test-bed 1 was used as an TI', which was listening for broadcasts regarding transaction initiation as well. Upon receiving a broadcast, it would initiate emission of a pre-generated infrared sequence, in a similar way as TI.

Although we assume that Android's random number generator is secure, this scenario was assessed mainly for reference reasons.

5. RESULTS AND EVALUATION

The evaluation methodology, the results, and the discussion of the results are presented in this section.

5.1 Data Analysis Methodology

The evaluation was conducted in two phases. During the first phase of evaluation, 50 transactions were performed in the lab for each of the test-beds. After the completion of each experimental phase, data from the RPis and the mobile device used as TI was transferred to a laptop (2x2.7GHz Intel i7 CPU and 8GB of RAM, running Fedora 24 OS) for analysis. The transfer was accomplished through SSH and USB cable, respectively. A program, written in Java, would then convert the captured pulse-pause sequence timings, recorded by TT' and TT, of each transaction into streams of bits. Only bits captured within time $\mathcal{T}_{Received}$ were regarded. Translation of pulse-pause timings into bits was accomplished by inverting the methodology described in Section 4.1. For this, time $\mathcal{T}_{AAE}$ of each transaction was also given as an argument to the program. The `dwdiff`[3] program was used for the comparison of the bit streams. This tool uses the *GNU diff* utility in order to detect the longest common sub-sequences between two lines (in our scenario

[2]WiringPi website: http://wiringpi.com/

[3]dwdiff website: http://os.ghalkes.nl/dwdiff.html

Table 1: Similarity Percentage Between Sent and Received Bits

	Test-bed 1	Test-bed 2	Test-bed 3	Test-bed 4	Test-bed 5	Test-bed 6
Genuine Pair	98.86%	98.78%	98.34%	98.20%	98.28%	98.32%
Relay Pair	3.70%	44.72%	22.30%	30.76%	81.94%	69.56%

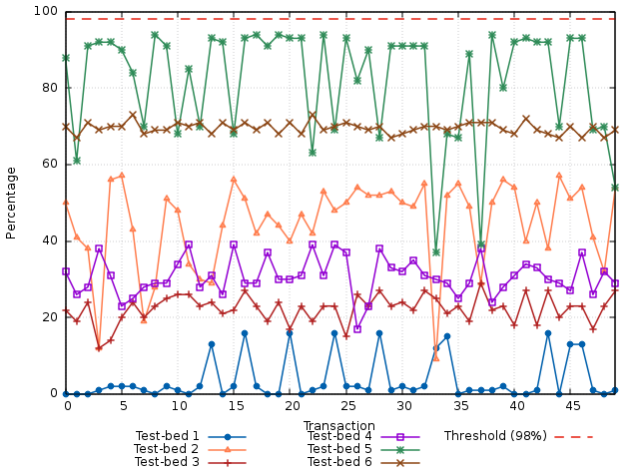

Figure 6: Test-bed Performance

each bit stream corresponded to one line), and based on that calculates the similarity percentage.

In the second evaluation phase, an additional 450 transactions were performed, using the best performing test-bed (Test-bed 5). The same evaluation procedure was followed.

5.2 Results Evaluation

Regarding the genuine transaction pair (TI-TT'), the similarity between the transmitted and the received bit streams was in all scenarios, except in nine cases, either 98% (in 17% of all transactions) or 99% (in 81% of all transactions). In these nine cases, a portion of the data was captured by TT' in the $\mathcal{T}_{Received}$ time frame. The reason for the delay of TI in transmitting the data is not clear, but may be related to the extensive use of the device during the experiments. However, some amount of failure is acceptable by the smart card industry [21].

For the evaluation of the proposed solution, we set a threshold of 98% similarity between the captured and transmitted streams, based on the evaluation results of the genuine pair. According to our results, the false negative rate would therefore be less than 2%.

The average similarity percentages for the genuine pair (TI-TT') and the relay pair (TI-TT) for 50 runs for each test-bed, are presented in Table 1. Test-bed 5 is an exception, as the average occurs from analysis of 500 runs. Figure 6 presents the similarity percentage of fifty individual transactions, for all test-beds (randomly chosen in the case of Test-bed 5).

The true negative rate, maintaining the threshold at 98%, was 100%. None of the assessed relay methodologies were capable of exceeding this threshold during our experiments.

In Test-bed 1, the mobile device was incapable of relaying a significant amount of data in the given time frame. In many occasions no data was relayed. Similar to the device TI, the relay mobile device required time x_i before it could initiate the infrared transmission.

In Test-bed 2, due to inconsistencies in the network packet transmission time, extra bits were frequently introduced or deducted from the bit stream. For example, for two subsequent infrared state alterations, in case of a delay in the transmission of the second network packet, comparatively to the arrival of the first packet, extra bits would be introduced until the arrival of that packet. Similarly, if the transmission delay of a third packet was shorter than that of the second, bits would be deduced in the time frame between the second and third packets.

Higher result consistency was noticed in Test-beds 3 and 4, compared to Test-bed 2, although the overall performance was lower. In these scenarios, caching was causing an overhead. Moreover, in contrast to Test-bed 2, in these scenarios pause to pulse switching (and the opposite) were based on relayed signal timings, instead of a single network packet. The overhead imposed due to the previous, caused detectable delays in relaying bits in the magnitude of microseconds. Finally, the $4ms$ caching window proved to be insufficient for caching, relaying and replaying signals. Test-bed 4 performed better than Test-bed 3, but although less than half the network packets were required in comparison to Test-bed 3, Test-bed 4 had to cache data on both sides of the relay, which led to suboptimal increase in the overall performance.

The highest performance was observed in Test-bed 5. Almost 52% of the captured relayed bit streams exceeded 90% similarity upon comparison with the transmitted bit stream. Almost 10% of which reached the maximum observed similarity of 94%. However, white noise interference captured by the antenna of the infrared extender was causing significant bit irregularities in certain relayed streams, leading to inconsistencies in the reproducibility of the results. Even though this technique produced a relatively high similarity, all the relayed transactions were detected, since the threshold of 98% was not reached in any case.

Finally, Test-bed 6 was the second best performing test-bed, even though in this test-bed, instead of signal relaying, a different random sequence than the one produced by TI was transmitted by TI'. In the design of a random number generator, the major concern is the unpredictability of pseudo-random sequences, and not the percentage similarity. In any pseudo-random sequence there is a chance that two random numbers have high similarity, but not the same sequence.

Other methods of attacking this system might exist, like relay via fibre optic cable or utilising the infrared range (for few metres), but due to their difficulty in concealment, and distance limitations, as well as the high expense associated with them, they were not investigated any further.

6. RELATED WORK AND FUTURE DIRECTIONS

In this section, the related work and future directions of this work are discussed.

6.1 Related Work

Drimer et al. [9] and Ma et al. [18] used GPS (Global Positioning System) to detect coexistence of NFC-enabled mobile devices. The time frame used by Ma et al. was 10 seconds, and data collection was occurring on every second. Upon completion of the data collection, data recorded by the two parties would be compared. High success rate for proximity detection was reported by the authors.

Ambient sound and light were used by Halevi et al. [13] as a proximity detection method. The authors reported high success rate, after measuring sound and light for 30 and 2 seconds respectively, and using multiple comparison algorithms. Finally, their experiments were conducted in a variety of physical locations.

Varshavsky et al. [27] suggested comparing the discovered Wi-Fi networks, along with their strengths, as a method of proximity detection. The authors reported positive results, and although the work is mainly focussed on secure device pairing, it is potentially applicable in the domain of mobile transactions as well.

Urien et al. [26] proposed ambient temperature, in combination with an elliptic curve-based RFID and/or NFC authentication protocol in order to detect device co-location. Their proposal however was not implemented, therefore no experimental data is available.

Mehrnezhad et al. [20] use the accelerometer to verify device proximity during a mobile payment transaction. The user is called to tap twice on the terminal for the transaction to complete. Upon completion of the transaction, sensor streams, measured by the two devices, are compared for similarity. The recording time is stated to be between 0.6 and 1.5 seconds, and a high success rate is reported.

Truong et al. [25] assessed a variety of ambient sensors for proximity detection, with positive results. However the recording time frame of 10-120 seconds makes this method inconclusive for many NFC-based mobile transactions.

Finally, Shrestha et al. [23] successfully used a Sensordrone (specialised hardware, featuring numerous ambient sensors) for proximity detection. No specific sample duration is reported in this work, however the authors state that data from each sensor was recorded for a few seconds.

Table 2 summarises the related work, by providing the sensors used in each of these works, the sample duration (if provided by the authors), and the suitability for using these methods in contactless mobile transactions.

Table 2: Related Work in Sensors as Anti-Relay Mechanism

Paper	Sensor(s) Used	Sample Duration	Contactless Suitability
Ma et al. [19]	GPS	10 seconds	Unlikely
Halevi et al. [13]	Audio	30 seconds	Unlikely
	Light	2 seconds	More Likely
Varshavsky et al. [27]	Wi-Fi (Radio Waves)	1 second	More Likely
Urien et al. [26]	Temperature	N/A	-
Mehrnezhad et al. [20]	Accelerometer	0.6 to 1.5 Seconds	More Likely
Truong et al. [25]	GPS Raw Data	120 seconds	Unlikely
	Wi-Fi	30 seconds	Unlikely
	Ambient Audio	10 seconds	Unlikely
	Bluetooth	12 seconds	Unlikely
Shrestha et al. [23]	Temperature (T)	Few seconds	Unlikely
	Precision Gas (G)	Few seconds	Unlikely
	Humidity (H)	Few seconds	Unlikely
	Altitude (A)	Few seconds	Unlikely
	HA	Few seconds	Unlikely
	HGA	Few seconds	Unlikely
	THGA	Few seconds	Unlikely

6.2 Future Directions

As part of our ongoing investigation, we are planning to extend this work towards several directions. We are aiming to explore and evaluate other AAE actuators, mentioned in Section 3.1.2, and assess their performance in comparison to this work. In addition, investigate the integration of AAC with traditional NFC protocols for effective proximity detection. Finally, the usability, user acceptance, and performance outside a lab environment are going to be assessed through a field study.

7. CONCLUSION

Previous works has claimed that sensing the ambient environment and comparing the measurements can be used as a proximity detection method between two devices. However, the effectiveness of this technique in certain types of transactions that require to be completed in short time frames ($\leq 500ms$), like banking and transport related transactions, has been argued.

In this paper we have proposed the generation of artificial ambient environments (AAEs) as a means of proximity/relay attack detection in smartphone contactless transactions with short timing restrictions. Out of a list of AAEs that we have provided, we have evaluated the proposed solution by using infrared as an AAE actuator. Six distinct ways of attacking the system were designed, built, and evaluated. Our experimental results showed that infrared has a high success rate for relay attack detection, as well as for proximity detection. None of the attack methods were capable of evading our framework.

8. REFERENCES

[1] ConsumerIrManager. https://goo.gl/SDZSEJ.

[2] MasterCard Contactless Performance Requirement. Online, MasterCard, March 2014.

[3] Transactions Acceptance Device Guide (TADG). Specification Version 3.0, VISA, May 2015.

[4] EMV Contactless Specifications for Payment Systems: Book D - EMV Contactless Communication Protocol Specification. Spec V2.6, EMVCo, LLC, March 2016.

[5] R. N. Akram, I. Gurulian, C. Shepherd, K. Markantonakis, and K. Mayes. Empirical Evaluation of Ambient Sensors as Proximity Detection Mechanism for Mobile Payments, 2016.

[6] I. Boureanu, A. Mitrokotsa, and S. Vaudenay. Towards Secure Distance Bounding. In *Fast Software Encryption*, pages 55–67. Springer, 2014.

[7] V. Coskun, B. Ozdenizci, and K. Ok. A Survey on Near Field Communication (NFC) Technology. *Wireless Personal Communications*, 71(3):2259–2294, 2013.

[8] C. Cremers, K. Rasmussen, B. Schmidt, and S. Capkun. Distance Hijacking Attacks on Distance Bounding Protocols. In *Security and Privacy, 2012 IEEE Symposium on*, pages 113–127, May 2012.

[9] S. Drimer and S. J. Murdoch. Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks. In N. Provos, editor, *USENIX Security*. USENIX Association, 2007.

[10] A. Francillon, B. Danev, and S. Capkun. Relay Attacks on Passive Keyless Entry and Start Systems

in Modern Cars. In *Network & Distributed System Security*, NDSS. The Internet Society, Feb. 2011.

[11] 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*, RFIDSec'10, pages 35–49, Berlin, Heidelberg, 2010. Springer-Verlag.

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

[13] T. Halevi, D. Ma, N. Saxena, and T. Xiang. Secure Proximity Detection for NFC Devices Based on Ambient Sensor Data. In S. Foresti, M. Yung, and F. Martinelli, editors, *Computer Security – ESORICS 2012*, LNCS, pages 379–396. Springer, 2012.

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

[15] D. He, S. Chan, and M. Guizani. Mobile application security: malware threats and defenses. *IEEE Wireless Communications*, 22(1):138–144, February 2015.

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

[17] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun. Sound-Proof: Usable Two-Factor Authentication Based on Ambient Sound. In *24th USENIX Security Symposium*, pages 483–498, Washington, D.C., Aug. 2015. USENIX Association.

[18] D. Ma, N. Saxena, T. Xiang, and Y. Zhu. Location-aware and safer cards: Enhancing rfid security and privacy via location sensing. *IEEE TDSC*, 10(2):57–69, 2013.

[19] D. Ma, N. Saxena, T. Xiang, and Y. Zhu. Location-Aware and Safer Cards: Enhancing RFID Security and Privacy via Location Sensing. *IEEE TDSC*, 10(2):57–69, March 2013.

[20] 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.

[21] W. Rankl and W. Effing. *Smart Card Handbook*. Wiley Publishing, 4th edition, 2010.

[22] K. B. Rasmussen and S. Capkun. Realization of RF Distance Bounding. In *USENIX Security Symposium*, pages 389–402, 2010.

[23] 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*, pages 349–364. Springer, 2014.

[24] The UK Cards Association. Contactless. https://goo.gl/hSh7Ij. [Online; accessed 2016-08-30].

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

[26] P. Urien and S. Piramuthu. Elliptic curve-based RFID/NFC authentication with temperature sensor input for relay attacks. *Decision Support Systems*, 59:28 – 36, 2014.

[27] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara. Amigo: Proximity-Based Authentication of Mobile Devices. In J. Krumm, G. Abowd, A. Seneviratne, and T. Strang, editors, *UbiComp 2007*, LNCS, pages 253–270. Springer, 2007.

[28] R. Verdult and F. Kooman. Practical Attacks on NFC Enabled Cell Phones. In *Proceedings of the 2011 Third International Workshop on Near Field Communication*, NFC '11, pages 77–82, Washington, DC, USA, 2011. IEEE Computer Society.