Artificial Ambient Environments for Proximity Critical Applications

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

ABSTRACT

In the field of smartphones a number of proposals suggest that sensing the ambient environment can act as an effective anti-relay mechanism. However, existing literature is not compliant with industry standards (e.g. EMV and ITSO) that require transactions to complete within a certain time-frame (e.g. 500ms in the case of EMV contactless payments). In previous work the generation of an artificial ambient environment (AAE), and especially the use of infrared light as an AAE actuator was shown to have high success rate in relay attacks detection. In this paper we investigate the application of infrared as a relay attack detection technique in various scenarios, namely, contactless transactions (mobile payments, transportation ticketing, and physical access control), and continuous Two-Factor Authentication. Operating requirements and architectures are proposed for each scenario, while taking into account industry imposed performance requirements, where applicable. Protocols for integrating the solution into the aforementioned scenarios are being proposed, and formally verified. The impact on the performance is assessed through practical implementation. Proposed protocols are verified using Scyther, a formal mechanical verification tool. Finally, additional scenarios, in which this technique can be applied to prevent relay or other types of attacks, are discussed.

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

ACM Reference format:
Iakovos Gurulian, Konstantinos Markantonakis, Raja Naeem Akram, Keith Mayes. 2017. Artificial Ambient Environments for Proximity Critical Applications. In *Proceedings of ARES '17, Reggio Calabria, Italy, August 29-September 01, 2017,* 10 pages.
https://doi.org/10.1145/3098954.3098964

ARES '17, August 29-September 01, 2017, Reggio Calabria, Italy

ACM ISBN 978-1-4503-5257-4/17/08…$15.00
https://doi.org/10.1145/3098954.3098964

1 INTRODUCTION

During a relay attack, an attacker is attempting to extend the physical distance between two communicating devices, using some form of relay equipment. By performing this type of attack, an attacker may gain access to services that a legitimate user is eligible for, like payments, access to buildings, or access to user accounts.

In the domain of smart cards, distance bounding protocols have been proposed in order to counter relay attacks [24, 37], whereas for smartphones, distance bounding protocols are not applicable due to the multitude of hardware components and the multi-process architecture which leads to unpredictable performance behaviour [39]. Relay attacks have been demonstrated in the field of smartphone-based Near-Field Communication (NFC) transactions [19, 20, 42].

Ambient sensing has been proposed as a potential alternative technique for countering relay attacks [23, 26, 34, 38, 40, 41]. This method often requires the devices involved in a transaction to collect data from their surrounding environment (e.g. the room temperature) that can imply their coexistence when compared against each other. However, the effectiveness of such techniques has been found to be insufficient in the case of time restricted contactless transactions [22, 33], like EMV contactless payment transactions, that are required to be completed within 500ms [3, 5–7, 9].

The generation of an artificial ambient environment (AAE), using infrared light, has demonstrated positive results regarding proximity detection/relay attack prevention in short time frames [21] (further details in Section 2.3). A random-bit sequence is emitted through the infrared blaster of the smartphone, and captured by the communicating party over a predefined short period (100ms). Infrared blasters are available on a large portion of modern smartphones [32]. The emitted and captured data are then compared against each other in order to establish proximity assurance.

Empirical evidence indicates that this technique is hard to relay using off-the-shelf equipment, providing a strong proximity detection/anti-relay mechanism against such attackers [21]. However, the applicability of this technique in real-world scenarios has not been investigated. Moreover, the architectural enhancements, limitations and requirements regarding the deployment in a large scale have not been discussed.

The primary contributions of this paper are:

- We describe how infrared light can be employed as a means of proximity detection/relay attack prevention in contactless transactions in mobile payments, physical access control, and transportation ticketing (Section 3).
- We propose a scheme for relay attack resilient continuous Two-Factor Authentication (2FA), for host-based services authentication (Section 4). Further possible scenarios are discussed in Section 5.

- We propose protocols that follow industry standards (where applicable), for the integration of the proximity detection technique. The security of all the protocols has been verified using the Scyther tool[1].
- We evaluate the performance of each of the aforementioned cases by practically implementing them. Our results indicate that the performance cost is low.

2 BACKGROUND

In this section, background and related work are discussed.

2.1 Relay Attacks

A wide range of applications can be affected by relay attacks, like contactless transactions, and access control. For example, in an NFC-based contactless payment using a smartphone, an attacker would have a malicious 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. A transaction between the payment instrument and the payment terminal will be initiated. All data communicated during a transaction should be relayed between the malicious devices, as shown in Figure 1. In this case, a transaction between the payment instrument and the payment terminal will be performed, however these devices can be beyond the operating environment of NFC.

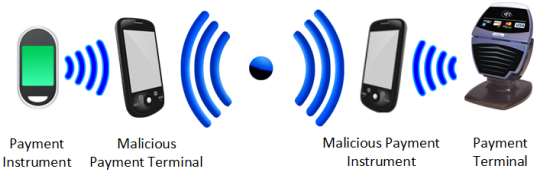

Figure 1: Relay Attack

Relay attack detection and prevention requires information regarding the coexistence of the devices involved in the transaction. As already mentioned, distance bounding protocols that are available for smart cards may not work in the field of smartphones. Several alternative methods have been proposed regarding relay attack prevention. Many of these methods have demonstrated that using the environmental (ambient) sensors that are present in modern smartphones can provide sufficient information (further discussion regarding related work in Section 2.2).

In such scenarios, the devices involved in a transaction record data, using some ambient sensor, for some predefined time. The recorded data from the two devices is then transmitted to an entity that performs a comparison, in order to decide whether the devices are in vicinity. This entity can either be one of the devices involved in the transaction, or a trusted third party (TTP).

However, the effectiveness of using data from the natural ambient environment in scenarios that require a limited time frame ($\leq 500ms$), like EMV contactless payments [9], and transportation related transactions [2, 17], has been questioned [22, 33]. The use of artificial ambience has been proposed instead [21], as an effective countermeasure.

[1]Scyther tool: https://www.cs.ox.ac.uk/people/cas.cremers/scyther/

2.2 Existing Anti-relay Mechanisms

Drimmer et al. [16] and Ma et al. [27] proposed the use of GPS (Global Positioning System) as a means of co-location detection. A time frame of 10 seconds was used by Ma et al. for data collection, and values were recorded every second. High success rate was reported by the authors for proximity detection.

Halevi et al. [23] 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. [41] 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. [40] 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. [30] 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. [38] 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. [35] 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 [22] and [33], 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.

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. [25] 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.

2.3 Infrared as a Proximity Detection Mechanism

As mentioned in sections 2.1 and 2.2, the natural ambient environment may not provide sufficient information regarding the coexistence of two devices in time restricted transactions. In [21], a proximity detection technique was proposed, by generating an artificial ambient environment (AAE). Infrared light was used to evaluate the effectiveness of the proposed solution. This method relies on a short transmission of a random-bit sequence, as pulses and pauses, through the infrared transmitter of the smartphone, over a period of $100ms$, upon the initiation of a transaction. The random-bit sequence is captured by the communicating party and compared for similarity against the sequence generated by the

smartphone. Infrared transmitters are available in a variety of Android devices, and mainly used for controlling home equipment (e.g. televisions) [11].

Each random bit in the sequence is represented by a 200μs long pulse or pause, so 500 random bits are transmitted over the course of 100ms. The infrared emitting side would therefore convert the random-bit sequence to 200μs-long pulse/pause sequences. The receiving side would translate the captured pulse and pause timings back into a bit-sequence, using the same principle. For example, the sequence "1101110011" would be converted into pulse and pause timed sequences, as depicted in Figure 2.

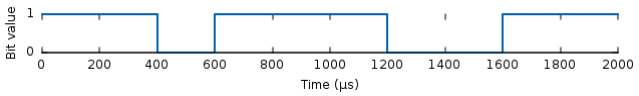

Figure 2: Representation of the Bit-Sequence "1101110011" in pulses–pauses

Theoretically, the bit timings can be reduced to a minimum of 13μs/bit, as the frequency of the consumer infrared is 38.4kHz. However, after experimentation we concluded that our evaluation equipment was capable of reliably transmitting a single bit in no less than 200μs.

Upon initiation of a transaction (e.g. through NFC or WiFi) between a transaction terminal (TT) and a transaction instrument (TI), TI transmits, and TT receives infrared signals. The transaction is initiated by device TT, which upon transmitting the transaction initiation message starts listening for infrared signals for some predefined time. Emission of the random-bit sequence is initiated by device TI upon receiving the transaction initiation message.

The infrared communication acts as a second channel, in order to detect whether the two devices are in proximity (Figure 3). The transmitted and/or received data is transferred between the two devices, or to a TTP, for comparison, in a secure manner through the first channel.

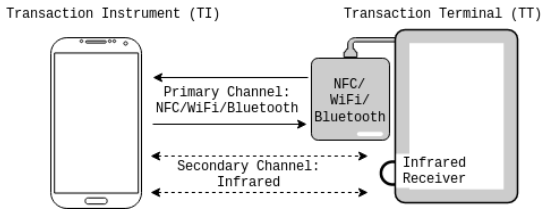

Figure 3: Framework Architecture

Through experimentation, the similarity of the emitted and received bits in a legitimate transaction reached or exceeded 98%, in more than 98% of the experimental runs. The similarity of 98% was therefore set as a threshold. Less than 2% false negative rate was perceived in the working prototype. The smart card industry accepts some amount of failure in transactions [31].

The challenge for an attacker is to successfully relay the bits transmitted through the infrared channel in a timely manner. In case relaying of a bit is delayed, a pulse or pause will last longer, leading to new bits being introduced in the random sequence. In case an attacker uses caching techniques and delays the transmission of the bits, some bits will not be received by the device TT, as they will be emitted after the last has stopped listening for infrared signals. The caching window for an attacker is approximately the 200μs, plus the maximum time that a communication might need to initiate (imposed by delays in the first communication channel – e.g. select application ID APDU through the NFC channel). It was calculated to be 4ms. In both occasions, if the similarity between the transmitted and received bits does not reach the threshold of 98%, the relay attack will be detected, and the transaction rejected.

Our experiments included six distinct relay techniques, using both off-the-shelf, and custom-built equipment. None of the relay techniques were capable of reaching the similarity threshold. The detailed techniques and results are presented in [21]. However, the integration of the technique in real world scenarios, related threat models, potential architectures, and performance cost has not been investigated. In the following sections we will be discussing and evaluating the use of the technique in various scenarios, susceptible to relay attacks.

3 CONTACTLESS TRANSACTIONS

Modern NFC-enabled smartphones are capable of performing smart card-like contactless transactions. Domains that benefit from this type of transactions include contactless payments, physical access control and transport ticketing. This way of performing transactions is convenient for the users, but due to security implications, restrictions are usually imposed by the operators. For example, contactless payments are limited to £30 per transaction in the UK [10].

These limitations apply in contactless transactions in general, including smart cards. However, smartphones are susceptible to more security threats. They usually do not contain secure storage, their operating system is much larger, therefore more prone to security flaws due to larger amount of coding errors [29], and users have control over the software installed on a device.

Among the possible attacks on contactless transactions is the relay attack. As already mentioned, in order to protect against relay attacks, the coexistence of the devices involved in a transaction should be able to be proven.

3.1 Threat Model

In a contactless transaction scenario, an attacker performing a relay attack can use a legitimate user's identity without authorisation. Unauthorised access to services that the legitimate user is entitled to can therefore be granted. For example, an attacker can perform a payment using the legitimate user's account, or get unauthorised access to buildings.

For the attack to take place, a relay pair should be operated by an attacker (Figure 1). A relay transaction instrument TI′, presented as a legitimate transaction instrument, should be tapped to a legitimate transaction terminal (TT) and initiate a transaction. Simultaneously, the attacker should tap the relay device TT′ to a legitimate payment instrument (TI). This can be achieved either by masquerading TT′ and presenting it to the legitimate user as a legitimate transaction terminal, or by tapping it without the user's consent.

For a successful attack, the communication messages should be relayed between the devices TI and TT, through the relay pair. As already mentioned, according to the EMV standards, a transaction should be completed within a time frame of $500ms$. Similarly, in the case of transport related transactions, a time restriction of $300 - 500ms$ is usually imposed [2]. Therefore, the relay equipment should be capable of relaying all the communication data between the legitimate device pair (TT – TI) in a timely manner.

In this paper, the attacker requires no prior interaction or knowledge of neither TT nor TI; the attack is of opportunistic nature and has only control over the devices TT′ and 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. Finally, the attacker only has access to off-the-shelf relay equipment.

3.2 Candidate Integration Architecture

The anti-relay mechanism should be optimised in order to detect relay attacks during the time frame of the transaction, as imposed by industry standards. In order for the proposed solution to be introduced to the existing infrastructure, an infrared receiver should be available on payment terminals. Moreover, payment instruments should be equipped with infrared blasters.

Since infrared light has a range of a few metres, a denial of service attack can be achieved. However, the objective of the adversary, described in Section 3.1, can still not be attained. A protective cover around the infrared sensor can effectively prevent such attacks, by blocking infrared light out of the range required for the payment process from reaching the infrared sensor.

In order for the bit-sequence comparison to take place, three pieces of information are required:

(1) The random-bit sequence generated and emitted by TI.
(2) The time required for TI to emit the information through infrared.
(3) The infrared sequence captured by TT.

The second element is required because a delay (initialisation time) was noticed before device TI could start emitting infrared. After analysis of Android's source code, we concluded that it is induced due to the infrared driver of the device. Since the source code of the driver is not publicly available, we are considering this delay, as we were not able to reduce it (further discussion in Section 3.5). However, since the total emission time ($100ms$) and the total initialisation time are known, we are able to find the time (relative to the initiation of the transaction), when device TI started emitting infrared. Any infrared signals received by device TT prior to the initiation of the emission, or after its completion were discarded by the comparing party.

The average total infrared emission process time by device TI was measured to be $226.66ms$, in 300 runs of the experiment, using a Samsung S5 mini (SM-G800F) device, running Android 5.1.1. In order for the relay attack detection method to be performed during the time of the transaction, the similarity comparison between the transmitted and the captured bit-sequence has to be performed by one of the devices involved in the transaction. The required data should be transferred through a secure channel, in this occasion, the NFC channel that is running in parallel.

In order to accelerate the relay attack detection technique, TI can encrypt and transfer the random-bit sequence to TT, through the NFC channel, in parallel to the infrared emission process. Since the two devices are considered trusted, the security of the system is maintained. Once the emission of the infrared is complete, the time required for the process to complete is also transferred in a secure manner from device TI to device TT. Device TT discards any bits received outside the emission time-frame and proceeds with the comparison of the emitted and received bits. The `wdiff`[2] program is responsible for the comparison process. If the computed similarity is greater or equal to 98%, device TT responds with an approval message to device TI. Contactless transactions tend to use either public, or symmetric key cryptography, hence the integration of the anti-relay mechanism was evaluated against these scenarios.

3.2.1 Public Key Based Protocol. In mobile payments and physical access control, public key cryptography is usually recommended for use in related protocols, according to EMV [18] and NIST [28], respectively. The protocol messages used for the secure exchange of the required data between the two devices are listed in Protocol 1. A mechanical formal verification of the protocol has been performed, using the Scyther tool, against all automatic Scyther claims, except the secrecy of data sent in plain text. The verification script can be found in Appendix A. No attacks were detected through the mechanical formal verification.

Protocol 1 Public-key Based Protocol

1: $TT \rightarrow TI : Cert_{Auth}(TT)||ID_{TT}||n_{TT}$
2: $TI \rightarrow TT : E_{PK_{TT}}\{ID_{TI}||ID_{TT}||n_{TI}||n_{TT}||K||IRseq\}$
$||S_{SK_{TI}}[K||n_{TT}]||Cert_{Auth}(TI)$
3: $TI \rightarrow TT : E_K\{ID_{TI}||ID_{TT}||n_{TI}||n_{TT}||IRtiming\}$
4: $TT \rightarrow TI : E_K\{ID_{TI}||ID_{TT}||n_{TI}||n_{TT}||approval||S_{SK_{TT}}[n_{TI}]\}$

– **Message 1:** Device TT sends to device TI its public key certificate $Cert_{Auth}(TT)$, its ID ID_{TT}, and a random nonce n_{TT}.

– **Message 2:** TI verifies the received certificate. If it is genuine, device TI encrypts, using the public key of device TT, its ID ID_{TI}, ID_{TT}, random nonces n_{TI}, and n_{TT}, a session key K, and the random-bit sequence that is emitted through the infrared channel $IRseq$. The aforementioned encrypted message, along with the RSA public key certificate of device TI, and a signature on K, and n_{TT} – using the private key of device TI – are sent to device TT.

– **Message 3:** Device TI sends to device TT, ID_{TI}, ID_{TT}, n_{TI}, n_{TT}, and the time required until it was capable of emitting infrared signals $IRseq$. The data is encrypted with the session key K. This message is separated from the previous for performance reasons, as $IRtiming$ is available after the infrared sequence is emitted, during which time $IRseq$ can be sent.

– **Message 4:** If the certificate of device TI, as well as the signature from Message 2 are verified, device TT proceeds to compute the emitted and captured infrared bit similarity. Device TT takes a decision and responds with an approval or rejection message. The decision message, along with ID_{TI}, ID_{TT}, n_{TI}, n_{TT}, and a signature

[2] wdiff: https://www.gnu.org/software/wdiff/

on n_{TI} – using the private key of device TT – are sent to device TI. The message is encrypted using key K (as in Message 3).

3.2.2 Symmetric Key Based Protocol. In the field of smart ticketing, protocols are often based on symmetric key cryptography. For example, the MIFARE Plus card [8], and the Calypso ticketing system [4]. In the case of the MIFARE Plus card, in order for the security of the system to be maintained in case the key of a card is compromised, each card uses a different key. The terminal contains a master key, from which the symmetric key of a card/smartphone is derived, through a process called symmetric key diversification [1]. The card provides the Unique Identification number (UID), the Application ID (AID), and the System Identifier (SID) to the terminal. This information is used as input to the diversification process in order for device TT to generate the key stored in device TI.

In a smartphone-based transaction, a dynamic element should also be included in the diversification process, possibly provided by the scheme operator. Wthout a dynamic element, a device with a compromised key would not be able to be used for this type of transaction again. Replacing the device in such scenarios is expensive. However, developing a diversification process that takes as input dynamic elements is out of the scope of this paper.

A symmetric key cryptography-based protocol was designed (Protocol 2) and undergone mechanical formal verification, using Scyther, against all automatic Scyther claims. The verification script can be found in Appendix B. No attacks were detected through mechanical formal verification.

Protocol 2 Symmetric-key Based Protocol

1: $TI \rightarrow TT : \textit{Key Diversification Information}$
2: $TT \rightarrow TI : E_K\{ID_{TT}||n_{TT}\}$
3: $TI \rightarrow TT : E_K\{ID_{TI}||ID_{TT}||n_{TI}||n_{TT}||IRseq\}$
4: $TI \rightarrow TT : E_K\{ID_{TI}||ID_{TT}||n_{TI}||n_{TT}||hash[IRseq] ||IRtiming\}$
5: $TT \rightarrow TI : E_K\{ID_{TI}||ID_{TT}||n_{TI}||n_{TT}||hash[IRseq] ||approval\}$

– **Message 1:** Device TI sends the information used by the diversification algorithm in order for device TT to generate the symmetric key K. A symmetric key compromise in a MIFARE card scenario requires replacement of the card, as the diversification data is static. Replacing a mobile device in such scenarios is very expensive and impractical. Therefore, the diversification information in this scenario includes the Device ID and a dynamic ID that can change in case the original key is compromised.

– **Message 2:** Device TT sends its ID ID_{TT}, and a random nonce n_{TT} to device TI. The message is encrypted using the symmetric key K, which is pre-shared between the two devices.

– **Message 3:** TI responds with its ID ID_{TI}, ID_{TT}, random nonces n_{TI}, and n_{TT}, and the random-bit sequence that is emitted through infrared $IRseq$. The message is encrypted using key K.

– **Message 4:** Upon completion of the emission of the random-bit sequence through infrared, device TI sends the emission initiation time $IRtiming$ to device TT. The message also contains the identities and the random nonces generated by the two devices, as well as the hash of $IRseq$. The last element is provided so that an attacker is not capable of flipping the message order of messages 2 and 3. This message is also encrypted using the key K.

– **Message 5:** Using the provided $IRseq$ and $IRtiming$, device TT returns a rejection or approval message to device TI. The approval, along with ID_{TI}, ID_{TT}, n_{TI}, n_{TT} and a hash on $IRseq$ are sent encrypted with key K to device TI.

3.3 Evaluation Framework

In order to evaluate the performance of the protocols and the proposed architecture, working prototypes were built. In all scenarios, as device TT we used a laptop running Fedora 25, with an i7-2620M CPU at 2.70GHz processor, and 8GB of RAM. Device TI was represented by a Samsung S5 mini (SM-G800F) device, running Android 5.1.1. The device is regarded to be of low/medium specifications. Two applications were built for each device, one for the evaluation of each of the two protocols. Both protocols are relevant in the case of contactless transactions, since public key cryptography is suggested in the cases of contactless payments and physical access control, and symmetric key in the case of transportation ticketing.

An NFC-reader (ACS ACR1281-C1) was attached to device TT. The applications for device TT were developed in Python, using the `pyscard` library[3], in order to control the NFC-reader. Android applications were developed in Java, using Host-based Card Emulation (HCE) [12] in order to control the NFC adapter of the device.

Raw infrared data captured in [21] was used for the emulation of infrared emission/capture. In order to further weigh potential impact on the performance of the system, device TI was emitting infrared signals during the time of the transaction. The performance impact of capturing infrared data is minimal. A Raspberry Pi, with an infrared receiver connected to its General Purpose Input Output (GPIO) pins was used to measure the system load when receiving infrared signals. The `liblirc` library[4] was used, and the performance burden on the device was negligible.

3.3.1 Public Key Based. In the first scenario, which applies in domains that require the use of public key cryptography in contactless transactions, 2048-bit RSA was used. Each device contained a public key pair, used for the transaction.

Upon the establishment of the session key, AES 128-bit encryption, using the CBC mode and PKCS5 padding, was used. All the keys and nonces were randomly generated, using the `SecureRandom` Java class on device TI, and the `Random` module of the `pycrypto` Python library on device TT.

The bit-similarity comparison between the captured and transmitted infrared sequence was performed by device TT, upon receiving the third message. Bits captured outside the time frame during which device TI was transmitting were discarded. The `wdiff` tool was then called through the `check_output` module of the `subprocess` Python library. The similarity percentage was returned to the application running on device TT. The *approval* flag was finally sent as part of the last message to device TI, based on that percentage (`0x01` if the similarity percentage was equal or exceeded the threshold of 98%, otherwise `0x00`).

[3] `pyscard` library: https://pyscard.sourceforge.io/
[4] `liblirc` library: http://www.lirc.org/

3.3.2 Symmetric Key Based. Similar to the public key-based scenario, 128-bit AES, CBC mode, with PKCS5 padding was used. The bit-similarity comparison was performed upon receiving the fourth message, by device TT, in the same way as in the public key-based scenario. The key diversification code, according to the document AN10922 [1], was integrated by modifying the implementation in [13]. Finally, SHA256 was used as the hashing algorithm.

3.4 Results

Each of the protocols was run 100 times in order to estimate the performance of the system, and evaluate the applicability in contactless transaction scenarios. The performance measurements can be found in Table 1. The measurements were taken by device TT and represent the round trip time (i.e. the time between device TT starts generating a message and the response that comes from device TI is processed). Since the clocks of the two devices cannot be perfectly synchronised, practically measuring the timing of each individual message was not feasible. In the first message of each protocol, the timing of the *SELECT* command is also included.

The last column lists the results of the second protocol, with lower number of bits transmitted (300 bits), for better integration with transport related contactless transactions, which require a time restriction of $300 - 500ms$ (further discussion in Section 3.5). The minimum and maximum observed timings in the 100 protocol runs, and the average running time are listed, for each of the protocol messages, the time required for the comparison of the transmitted and captured bit-streams, and the total protocol running time.

Table 1: Results of Contactless Transactions (in *ms*)

	Protocol 1			Protocol 2			Protocol 2 (300b)		
	min	***max***	***avg***	***min***	***max***	***avg***	***min***	***max***	***avg***
Diversification	-	-	-	30.2	42.4	39.0	28.1	50.5	38.8
Round Trip 1	116.1	169.4	152.2	26.6	81.5	32.6	21.9	40.9	27.6
Round Trip 2	95.7	135.5	109.4	146.6	201.5	194.7	95.5	115.2	109.0
Round Trip 3	52.1	133.7	70.1	24.4	32.1	27.2	24.6	38.0	28.6
Bit Similarity	3.8	5.9	4.3	3.8	5.2	4.5	3.5	5.2	4.1
Total	304.9	395.4	331.8	284.3	299.7	293.5	193.8	215.0	204.0

3.5 Discussion

The results indicate that the running time of the first protocol are within the timing bounds that EMV requires. Even though the infrared emission lasts $100ms$, the average total emission time was approximately $227ms$. The aforementioned emission time refers to the time between requesting the infrared sequence emission from the operating system, until the completion of the process. After inspection of the Android Open Source Project's (AOSP) source code[5], we concluded that this delay is imposed either by the proprietary infrared drivers, or by modifications on Android's source code by Samsung. Since access to the source code of these components is not available, due to their proprietary nature, we were not able to investigate the problem further. However, we were capable of emitting infrared signals, using an infrared LED attached to the GPIO pins of a Raspberry Pi, controlled by a program written in the C language, with negligible delay. Therefore, we believe that this delay can be confined. An average extra performance cost of approximately $57ms$ and $82ms$ was measured in the public and symmetric key scenarios respectively, in the case of 500 bits.

It should be stressed that the measured performance is an indicator of the performance cost of NFC contactless transactions, using the proposed solution. An additional barrier was the multitude of EMV and physical access control implementations, as well as the limited available information regarding major transport ticketing protocols. However, industry standards, like the EMV, were used as guidelines for developing and delivering a robust solution.

In a real-world scenario, more information is exchanged between the devices as part of the transactions, that is not covered by our protocols. Real-world applications, after the establishment of a secure channel between the devices involved in a transaction, would require to transfer the random-bit sequence, and the total emission time, in order for the similarity comparison to be performed.

Since some transportation ticketing transactions require to be completed within $300 - 500ms$, as mentioned earlier, the performance of the second protocol might not be optimal for some slower devices. Therefore, we repeated the experiment with a reduced number of random bits. A total of 300 randomly generated bits were generated in the second experimental round, constituting an emission time of $60ms$ The probability of an attacker guessing the random sequence, assuming its true random nature, is $\frac{1}{2^{300}}$.

The average performance of the protocol run was improved, with potential to successfully be integrated into such protocols. The average total emission time, caused due to the abovementioned delay in the emission was $137ms$. An average extra performance cost of approximately $32ms$ was introduced due to the issue.

4 CONTINUOUS TWO-FACTOR AUTHENTICATION

In order to add an extra layer of security, access to websites, services, or systems, may require an extra verification step in addition to the login credentials. This second layer of security is often referred to as *Two-Factor Authentication* (2FA). A widely adopted 2FA technique is the use of a one time password (OTP), which can be sent to the user via SMS, generated by an application on the user's device, or other means. In the case of host-based 2FA, hardware tokens, like smart cards or OTP generators, are often used. Upon the successful completion of the login credentials, users are typically requested to proceed by providing the 2FA information in order to successfully complete the login process. Access to a service that provides 2FA functionality is only granted upon successful completion of both authentication steps.

Using an OTP as a second factor usually requires input of the OTP once per session. Depending on the system settings, a session may last for several months, without requesting an OTP again. Continuous authentication is not achieved in this case. Moreover, being able to provide the OTP to the service does not imply the user's physical presence close to the device. Shoulder surfing attacks can be used for example by attackers in order to compromise the OTP and login in a remote location [14], as the expiration time of these passwords can be several seconds long.

[5] AOSP: https://source.android.com/

In the case of smart cards, an extra cost is associated, and distance bounding protocols may not be applicable in order to counter relay attacks [39]. The multi-process nature of the host (e.g. a server or a personal computer), the operating system and its configuration, and the hardware variability do not assist towards this direction.

We are hence proposing a 2FA model, based on the principle of infrared as a proximity detection method, capable of providing continuous authentication, while eliminating the risk of attacks, like shoulder surfing and relay attacks. In this scenario, the secure primary channel is established over WiFi or Bluetooth between devices TT and TI (a personal computer and a smartphone, respectively). The second channel is the infrared channel, as described in the previous sections. Further discussion on the architecture can be found in Section 4.2.

4.1 Threat Model

Since relay attack detection is based on the comparison of data captured by both devices TT and TI, both devices should be trusted. In case device TT is not trusted, captured data can be manipulated or delayed, which can lead to a false proximity result. In the case of a website, which can be accessed from multiple devices, device TT cannot be considered trusted. By requiring some trusted execution environment, like Intel SGX [15], trust in device TT can be obtained. However, even though a relay attack might not be applicable in this scenario, an attacker in the vicinity to device TI can successfully login using a computer (e.g. a Raspberry Pi), which is controlled locally or remotely. Therefore, this technique is only applicable in scenarios where logging in to a service can be achieved through a single, trusted device, like user login to an operating system and other host-based services.

Similar to contactless transactions, in order for an attacker to perform a relay attack, a relay pair TT′ – TI′ is required. The goal of the attacker is to be able to successfully relay infrared signals in a timely manner. Since the same design principles described in Section 2.3 are used, an attacker has a very limited window for caching infrared signals and replaying them at a remote location.

4.2 Candidate Integration Architecture

The generic architecture of the proposed system is depicted in Figure 4. When a user logs in to a system that supports the proposed solution, a communication initiates with an underlying service, responsible for the 2FA process. In order to provide continuous authentication, the service is called periodically (e.g. every 15 seconds). Whenever the service is called, it communicates with device TI through a secure WiFi or Bluetooth channel. Device TI must be running an application capable of communicating with the service. The application running on device TI should be linked to the service that device TT is attempting to login to (e.g. by prior login of the user to the mobile application).

Depending on the settings of the service, the underlying service can be called periodically and request from device TI to provide an authentication token, in the form of random infrared bits. If the authentication is unsuccessful, an action is taken. Actions can vary, according to the requirements of the service. For example, the active session on device TT can be discontinued.

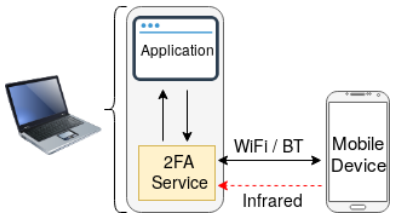

Figure 4: Architecture of Two-Factor Authentication

Since the success of the authentication transaction depends on the orientation of the two devices (i.e. whether the infrared blaster is aligned with the infrared receiver), less drastic measures might be taken instead. For example, permission escalation can be used to restrict access to certain critical actions and directories.

Either of the two protocols described in Section 3.2 can be used for the establishment of the secure channel, depending on the requirements and the implementation. For example, a symmetric key can be shared during the setup process, by asking the user to scan a barcode. The key can be periodically updated, but this is out of the scope of this paper. In a multi-user environments, such keys can be generated through symmetric key diversification. In the case of multi-user environments, public key cryptography might be a better option for logging user access.

4.3 Evaluation Framework

The same devices described in Section 3.3 were used in this scenario as well as devices TI and TT. Instead of an NFC adapter, two alternate primary communication channels were examined; WiFi and Bluetooth.

In order to evaluate the applicability and performance of the proposed architecture, a working prototype was built. An application was developed in Java, represented the application that requires 2FA as part of the authentication process. In order for a user to remain logged in, the successful completion of the 2FA had to be achieved. Otherwise, a failure message was displayed and access to certain components of the application was restricted. The 2FA service was called every 15 seconds, in order to provide continuous authentication assurance.

Four Android and four Python applications were built for TI and TT, respectively, in order to assess all the possible combinations. Namely, both protocols described in Section 3.2 were implemented for each of the two primary channel options (WiFi and Bluetooth) applicable to this scenario. In the case of WiFi, the two devices were connected to the same wireless network, while in the case of Bluetooth, a pairing between the devices had to be established before the initiation of the 2FA process. The pairing was performed through the Python applications, using the `PyBluez` library[6].

Upon initialisation of the application requiring 2FA, a call towards the underlying service was triggered. The service, written in Python, would then establish a connection to device TI, in the case of Bluetooth. In the case of WiFi, no session was required, since the communication was performed through UDP packets. Device TT would then request from device TI to initiate a transaction,

[6] `PyBluez` library: https://pypi.python.org/pypi/PyBluez

through the established Bluetooth RFCOMM channel, or a UDP packet. Upon transmission of the message, device TT would begin listening for infrared signals through an infrared receiver for 100*ms*. Upon receiving the message, device TI would start emitting infrared signals through its infrared blaster. Infrared data captured in [21] was used in this case as well. Communication would proceed in parallel through both channels. Using the `wdiff` tool, device TT would compute the similarity between the emitted and captured infrared sequences. If the similarity threshold was reached, upon completion of the protocol, the Python service would allow access to certain elements of the application.

4.4 Results and Discussion

The results of 100 protocol runs over WiFi and 100 over Bluetooth are presented in tables 2 and 3, respectively. For each protocol, the minimum, maximum, and average running time are listed.

Table 2: Results of 2FA – WiFi (in *ms*)

	Protocol 1			Protocol 2		
	min	***max***	***avg***	***min***	***max***	***avg***
Diversification	-	-	-	2.6	5.2	2.7
Round Trip 1	21.7	98.0	30.3	6.2	67.9	8.6
Round Trip 2	131.5	252.6	201.2	162.1	277.3	224.2
Round Trip 3	16.8	62.1	22.6	6.3	18.2	7.9
Bit Similarity	4.0	6.9	4.7	3.9	10.6	4.6
Total	246.3	305.2	254.2	238.4	297.3	243.3

Table 3: Results of 2FA – Bluetooth (in *ms*)

	Protocol 1			Protocol 2		
	min	***max***	***avg***	***min***	***max***	***avg***
Diversification	-	-	-	2.6	3.5	2.7
Round Trip 1	74.3	182.2	116.9	52.1	153.9	76.9
Round Trip 2	123.2	200.6	175.2	129.0	262.3	209.9
Round Trip 3	54.5	144.7	70.7	35.4	154.3	57.2
Bit Similarity	3.9	5.7	4.6	3.8	13.5	4.6
Total	323.6	440.1	362.8	306.2	426.2	346.8

The results indicate the effectiveness of the proposed solution. The overhead of the process was minimal, and the effectiveness high. From a usability perspective, the setup process needs to be completed once, and thereafter the user only has to place device TI to be facing towards device TT. In order to enhance the security of the system, the user can be required to unlock the device in order for the process to initiate.

A usability concern may rise when the user needs to use device TI in order to make or receive a call, while using device TT. A temporal suspension of the continuous authentication process can be applied in this occasion, in case an initial login using both factors has already been completed.

An attacker with access to device TT can further use techniques in order to compromise the system. For example, modify the security parameters by externally plugging the hard drive to a computer. Disk encryption can assist towards the prevention of such attacks.

Finally, since no time restrictions apply in this scenario, the natural ambient environment can be used instead of generating artificial ambience. However, manipulation or prediction of the natural ambient environment might be possible. For example, if the room temperature is used to detect proximity, a heater can be used in order to alter the values to the attacker's benefit. In case of ambient sound, an attacker can compromise the system by following the habits of a legitimate user. When a user is watching a TV show, the attacker can produce the same ambient sound, and successfully authenticate. Attacks on using the sound as a proximity detection mechanism have been described and demonstrated in previous works [25, 36]. Finally, in the case of light, an attacker can cover the light sensor of device TT when device TI is in the pocket of the legitimate user, and gain access to the system.

5 OTHER SCENARIOS

Other scenarios, that are either not in compliance with the context of this paper (Section 5.1), or are covered by the aforementioned scenarios (Section 5.2), are described in this section.

5.1 2FA for Logging In to Web Services

An attacker who has the ability to place a device in the vicinity of TI, may be able to successfully login, using the standard procedure. Although this is not a relay attack, as no data is being relayed between devices TI and TT, access to the legitimate user's account can be achieved. In scenarios where a relay attack is not of concern, infrared emission can operate as a 2FA provider, without requiring device TT to be trusted. For example if an attacker has compromised the credentials of a user, but does not have access to the vicinity of device TI. Since the main objective of this paper is to provide anti-relay capabilities in real-world scenarios, and the aforementioned case does not fit this context, it is listed as a future work direction.

As a note, we would like to stress that in such scenario the authentication must not be performed by device TT, but by the service provider, or device TI. Since device TT is not trusted, the overall architecture of the system in this occasion should be modified. The resulting architecture should be similar to the architecture of Sound-Proof, as described by Karapanos et al. in [25]. In their proposal the use of ambient sound is employed in order to provide device proximity proof and use it as a means of zero-effort 2FA. The work is mostly focused on usability, and potential security problems have been discussed by both the authors and by Shrestha et al. in [36]. We believe that most of the discussed security threats can be eliminated by replacing sound with a randomly generated infrared sequence, however some usability might be diminished.

5.2 Internet of Things Co-Presence Verification

In many occasions, in order for Internet of Things (IoT) devices to achieve their intended operation, proximity should be guaranteed. For example, in a security system, in order to avoid blind spots, in many cases devices should be located within visibility range. An attacker can move some components of the security system and apply a relay attack in order to deceit the system.

By applying the proposed solution, proximity evidence can be provided among such devices. A similar approach to the ones described in this paper can be applied. When one IoT device requests

proximity evidence from another, the same procedure is executed, with the requesting device acting as device TT and the proving device as device TI. In this scenario, the IoT devices of the system should be considered as trusted as well.

6 CONCLUSION AND FUTURE DIRECTIONS

In this paper we investigated the integration of infrared as an artificial ambient environment actuator for relay attack prevention in different real-world scenarios. Existing industry standards were taken into consideration, where applicable. Contactless transactions (EMV payments, physical access control, and transport ticketing), as well as continuous host-based two-factor authentication were investigated. Online two-factor authentication, and IoT co-presence verification were also discussed.

Integration architectures were described, and working prototypes were built and evaluated. Proposed protocols were undergone mechanical formal evaluation. Our results indicated that integration of the proposed solution with existing industry standards can be used to counter relay attacks, and is in compliance with current operational timing requirements, where applicable.

As part of our ongoing investigation, we are planning to extend this work in various directions. We are planning to examine the use of infrared light for continuous two-factor authentication for online logins. Also, to perform a user study in order to assess the usability of the proposed countermeasure. Finally, to investigate other AAE actuators, such as vibration and sound.

REFERENCES

[1] 2010. *AN10922: Symmetric key diversifications.* Technical Report. NXP.

[2] 2011. *Transit and Contactless Open Payments: An Emerging Approach for Fare Collection.* White Paper. Smart Card Alliance Transportation Council.

[3] 2013. *The Future of Ticketing: Paying for Public Transport Journeys Using Visa Cards in the 21st Century.* Whitepaper. VISA.

[4] 2014. *CALYPSO FUNCTIONAL SPECIFICATION, Card Application.* Technical Report. Calypso Networks Association.

[5] 2014. *How to Optimize the Consumer Contactless Experience? The Perfect Tap.* Technical Report. MasterCard.

[6] 2014. *MasterCard Contactless Performance Requirement.* Online. MasterCard.

[7] 2016. *EMV Contactless Specifications for Payment Systems: Book D - EMV Contactless Communication Protocol Specification.* Spec V2.6. EMVCo, LLC.

[8] 2016. *MF1P(H)x1y1, MIFARE Plus EV1, Rev. 2.* Preliminary short data sheet. NXP.

[9] 2016. *Transactions Acceptance Device Guide (TADG).* Specification Version 3.1. VISA.

[10] 2016. *UK Card Payments Summary 2016.* Technical Report. The UK Cards Association.

[11] Android API. 2017. ConsumerIrManager. https://developer.android.com/reference/android/hardware/ConsumerIrManager.html. (2017).

[12] Android API. 2017. Host-based Card Emulation. https://developer.android.com/guide/topics/connectivity/nfc/hce.html. (2017).

[13] Mariano Benedettini. 2011. Gist: key_generation.py. https://gist.github.com/mbenedettini/1409585. (2011).

[14] Fred Cheng. 2010. A Secure Mobile OTP Token. In *Mobile Wireless Middleware, Operating Systems, and Applications: Third International Conference, Mobilware 2010, Chicago, IL, USA, June 30 - July 2, 2010. Revised Selected Papers*, Ying Cai, Thomas Magedanz, Minglu Li, Jinchun Xia, and Carlo Giannelli (Eds.). Springer Berlin Heidelberg, 3–16.

[15] Victor Costan and Srinivas Devadas. 2016. Intel SGX Explained. *IACR Cryptology ePrint Archive* 2016 (2016).

[16] Saar Drimer and Steven J. Murdoch. 2007. Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks.. In *USENIX Security*, Niels Provos (Ed.). USENIX Association.

[17] Martin Emms, Budi Arief, Leo Freitas, Joseph Hannon, and Aad van Moorsel. 2014. Harvesting High Value Foreign Currency Transactions from EMV Contactless Credit Cards without the PIN. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 716–726.

[18] EMVCo. 2011. *EMV Integrated Circuit Card, Specifications for Payment Systems, Book 2, Security and Key Management, Version 4.3.*

[19] Lishoy Francis, Gerhard Hancke, Keith Mayes, and Konstantinos Markantonakis. 2010. 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)*. Springer-Verlag, Berlin, Heidelberg, 35–49.

[20] Lishoy Francis, Gerhard P. Hancke, Keith Mayes, and Konstantinos Markantonakis. 2011. Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones. *IACR Cryptology Archive* 2011 (2011), 618.

[21] Iakovos Gurulian, Raja Naeem Akram, Konstantinos Markantonakis, and Keith Mayes. 2017. Preventing Relay Attacks in Mobile Transactions Using Infrared Light. In *Proceedings of the Symposium on Applied Computing (SAC '17)*. ACM, New York, NY, USA, 1724–1731. https://doi.org/10.1145/3019612.3019794

[22] Iakovos Gurulian, Carlton Shepherd, Eibe Frank, Konstantinos Markantonakis, Raja Akram, and Keith Mayes. 2017. 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.

[23] Tzipora Halevi, Di Ma, Nitesh Saxena, and Tuo Xiang. 2012. Secure Proximity Detection for NFC Devices Based on Ambient Sensor Data. In *Computer Security – ESORICS 2012*, Sara Foresti, Moti Yung, and Fabio Martinelli (Eds.). Springer.

[24] Gerhard P. Hancke and Markus G. Kuhn. 2005. An RFID Distance Bounding Protocol. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM '05)*. IEEE Computer Society, Washington, DC, USA, 67–73.

[25] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. 2015. Sound-Proof: Usable Two-Factor Authentication Based on Ambient Sound. In *24th USENIX Security Symposium*. USENIX Association, Washington, D.C.

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

[27] Di Ma, Navrati Saxena, Tuo Xiang, and Yan Zhu. 2013. Location-aware and safer cards: Enhancing rfid security and privacy via location sensing. *IEEE TDSC* 10, 2 (2013), 57–69.

[28] William I. MacGregor, Ketan L. Mehta, David A. Cooper, and Karen A. Scarfone. 2008. *A Recommendation for the Use of PIV Credentials in Physical Access Control Systems (PACS).* Special Publication (NIST SP) – 800–116. NIST.

[29] Steve McConnell. 2004. *Code Complete* (2 ed.). Microsoft Press.

[30] Maryam Mehrnezhad, Feng Hao, and Siamak F. Shahandashti. 2014. *Tap-Tap and Pay (TTP): Preventing Man-in-the-Middle Attacks in NFC Payment Using Mobile Sensors.* Technical Report CS-TR-1428. Newcastle University.

[31] Wolfgang Rankli and Wolfgang Effing. 2010. *Smart Card Handbook* (4 ed.). Wiley.

[32] Brittany A. Roston. 2016. Use a phone as a remote control: today's phones with IR blasters. https://www.slashgear.com/use-a-phone-as-a-remote-control-todays-phones-with-ir-blasters-03457654/. (Oct. 2016).

[33] Carlton Shepherd, Iakovos Gurulian, Eibe Frank, Konstantinos Markantonakis, Raja Akram, Keith Mayes, and Emmanouil Panaousis. 2017. The Applicability of Ambient Sensors as Proximity Evidence for NFC Transactions. In *Mobile Security Technologies, IEEE Security and Privacy Workshops (MoST '17)*. IEEE.

[34] Babins Shrestha, Nitesh Saxena, Hien Thi Thu Truong, and N Asokan. 2014. Drone to the Rescue: Relay-Resilient Authentication using Ambient Multi-sensing. In *Financial Cryptography and Data Security*. Springer, 349–364.

[35] Babins Shrestha, Nitesh Saxena, Hien Thi Thu Truong, and N Asokan. 2014. Drone to the Rescue: Relay-Resilient Authentication using Ambient Multi-sensing. In *Financial Cryptography and Data Security*. Springer, 349–364.

[36] Babins Shrestha, Maliheh Shirvanian, Prakash Shrestha, and Nitesh Saxena. 2016. The Sounds of the Phones: Dangers of Zero-Effort Second Factor Login Based on Ambient Audio. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. ACM, New York, NY, USA, 908–919.

[37] Rolando Trujillo-Rasua, Benjamin Martin, and Gildas Avoine. 2010. The Poulidor distance-bounding protocol. In *Radio Frequency Identification: Security and Privacy Issues*. Springer, 239–257.

[38] Hien Thi Thu Truong, Xiang Gao, Biva Shrestha, Navrati Saxena, N Asokan, and Petteri Nurmi. 2014. 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, 163–171.

[39] Assad Umar, Keith Mayes, and Konstantinos Markantonakis. 2015. Performance Variation in Host-Based Card Emulation Compared to a Hardware Security Element. In *Mobile and Secure Services, 2015 First Conference on*. IEEE, 1–6.

[40] Pascal Urien and Selwyn Piramuthu. 2014. Elliptic curve-based RFID/NFC authentication with temperature sensor input for relay attacks. *Decision Support Systems* 59 (2014), 28 – 36.

[41] Alex Varshavsky, Adin Scannell, Anthony LaMarca, and Eyal de Lara. 2007. Amigo: Proximity-Based Authentication of Mobile Devices. In *UbiComp 2007*, John Krumm, GregoryD. Abowd, Aruna Seneviratne, and Thomas Strang (Eds.). Springer, 253–270.

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

A PROTOCOL 1 – SCYTHER SCRIPT

```
usertype IRsequence;
hashfunction h;
usertype SessionKey;
secret Cert: Function;

protocol protPK(TI,TT) {
  role TI {
    fresh nti: Nonce;
    var ntt: Nonce;
    fresh IRSeq: IRsequence;
    fresh IRTiming: Ticket;
    var Y: Ticket;
    fresh K: SessionKey;
    recv_1(TT,TI, TT,ntt,Cert(TT));
    send_2(TI,TT, {TI,TT,nti,ntt,K,IRSeq}pk(TT),{h
        (K,ntt)}sk(TI),Cert(TI));
    send_3(TI,TT, {TI,TT,nti,ntt,IRTiming}K);
    recv_4(TT,TI, {TI,TT,nti,ntt,Y,{h(nti)}sk(TT)}
        K);

    claim(TI, Alive);
    claim(TI, Secret, K);
    claim(TI, Secret, nti);
    claim(TI, Niagree);
    claim(TI, Nisynch);
    claim(TI, Secret, IRSeq);
    claim(TI, Secret, IRTiming);
    claim(TI, Secret, Y);
  }

  role TT {
    var nti: Nonce;
    fresh ntt: Nonce;
    var X: Ticket;
    var Y: Ticket;
    fresh apprv: Ticket;
    var K: SessionKey;
    send_1(TT,TI, TT,ntt,Cert(TT));
    recv_2(TI,TT, {TI,TT,nti,ntt,K,X}pk(TT),{h(K,
        ntt)}sk(TI),Cert(TI));
    recv_3(TI,TT, {TI,TT,nti,ntt,Y}K);
    send_4(TT,TI, {TI,TT,nti,ntt,apprv,{h(nti)}sk(
        TT)}K);

    claim(TT, Alive);
    claim(TT, Secret, K);
    claim(TT, Secret, nti);
    claim(TT, Niagree);
    claim(TT, Nisynch);
    claim(TT, Secret, X);
    claim(TT, Secret, Y);
    claim(TT, Secret, apprv);
  }
}
```

B PROTOCOL 2 – SCYTHER SCRIPT

```
usertype IRsequence;
hashfunction h;

macro K = k(TI,TT);

symmetric-role protocol protSK(TI,TT) {
  role TI {
    fresh nti: Nonce;
    var ntt: Nonce;
    fresh IRSeq: IRsequence;
    fresh IRTiming: IRsequence;
    var Y: Ticket;
    recv_1(TT,TI, {TT,ntt}K);
    send_2(TI,TT, {TI,TT,nti,ntt,IRSeq}K);
    send_3(TI,TT, {TI,TT,nti,ntt,h(IRSeq),IRTiming
        }K);
    recv_4(TT,TI, {TI,TT,nti,ntt,Y,h(IRSeq)}K);

    claim(TI, Alive);
    claim(TI, Niagree);
    claim(TI, Nisynch);
    claim(TI, SKR, ntt);
    claim(TI, SKR, nti);
    claim(TI, Secret, IRSeq);
    claim(TI, Secret, IRTiming);
    claim(TI, Secret, Y);
  }

  role TT {
    var nti: Nonce;
    fresh ntt: Nonce;
    var X: Ticket;
    var Y: Ticket;
    fresh apprv: Ticket;
    send_1(TT,TI, {TT,ntt}K);
    recv_2(TI,TT, {TI,TT,nti,ntt,X}K);
    recv_3(TI,TT, {TI,TT,nti,ntt,h(X),Y}K);
    send_4(TT,TI, {TI,TT,nti,ntt,apprv,h(X)}K);

    claim(TT, Alive);
    claim(TT, Niagree);
    claim(TT, Nisynch);
    claim(TT, SKR, ntt);
    claim(TT, SKR, nti);
    claim(TT, Secret, X);
    claim(TT, Secret, Y);
    claim(TT, Secret, apprv);
  }
}
```