HB⁺DB: Distance bounding meets human based authentication

Elena Pagnin^a, Anjia Yang^{b,*}, Qiao Hu^b, Gerhard Hancke^b, Aikaterini Mitrokotsa^a

^a Chalmers University of Technology, Sweden

^b City University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:

Received 1 September 2015

Received in revised form

24 May 2016

Accepted 25 May 2016

Available online 11 June 2016

Keywords:

Distance bounding

HB-protocol

Physical layer security

ABSTRACT

Authentication for resource-constrained devices is seen as one of the major challenges in current wireless communication networks. The HB⁺ protocol by Juels and Weis provides device authentication based on the *learning parity with noise* (LPN) problem and is appropriate for resource-constrained devices, but it has been shown to be vulnerable to a simple man-in-the-middle attack. Subsequent work has focused on modifying the cryptographic properties of the original protocol to mitigate this problem. We propose that this attack could be mitigated using physical layer measures from distance-bounding protocols and simple modifications to devices' radio receivers. We take the HB⁺ as a reference protocol and combine it with distance-bounding techniques. This hybrid solution, the HB⁺DB protocol is shown to provide resistance against the man-in-the-middle attacks on HB⁺ as a result of the additional physical-layer mechanisms. We analyze the security of the proposed HB⁺DB protocol against active man-in-the-middle attacks and present experiments showing how it is practically possible to limit the success of a practical man-in-the-middle attack. We also briefly discuss the possibility that HB⁺DB could provide some resistance to basic threats scenarios meant to be mitigated by distance-bounding protocols. We make a practical implementation to verify that our proposed method is feasible. Finally, we discuss a proof-of-concept channel for our scheme implemented on a platform equivalent in resources to a contactless smart card/NFC device.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Human-executable authentication was initially proposed by Hopper and Blum with the HB protocol [1]. The objective was to achieve secure authentication and identification relying on the limited abilities of humans to *remember* and *compute*. Human-executable protocols can for instance be employed to allow a human being to log into an untrusted terminal without using a scratch paper or a computational device while someone is spying on him.

A few years later, Juels and Weis [2] noticed that human beings' computational and storing limitations are very similar to the ones of low-cost Radio Frequency Identification (RFID) tags and other resource-constrained pervasive devices. Indeed, in both cases, people and devices have limited abilities in remembering long passwords and performing long calculations in their memory. As one kind of user-centric devices, RFID tags have been widely deployed in our daily life, such as smart cards, payment systems

and e-passports. Meanwhile, we also need pay attention to the security issues when utilizing these low-cost devices. Juels and Weis proposed an enhanced variant of the HB protocol, called HB⁺ [2], which is suitable for resource-constrained devices. The security of the HB⁺ protocols against passive attacks relies on the hardness of the *learning parity with noise* (LPN) problem [3], a well-known $\mathcal{NP}$ -hard problem. In the same year HB⁺ was proposed, Gilbert et al. presented an active attack against it, known as the GRS attack [4]. The core idea is that if an active adversary can modify the communication between the prover (e.g., RFID tag) and the verifier (e.g., RFID reader), then she can learn one bit of the secret(s) at each new run of the protocol. Subsequently, multiple variations of the HB⁺ have been proposed (including HB⁺⁺ [5], HB* [6], HB-MP [7]). Most of these protocols, however, have been shown to be vulnerable against man-in-the-middle (MiM) attacks [8]. Gilbert et al. designed two solutions (RANDOM-HB[#] and HB[#] [8]). Although both protocols are proven resistant against the GRS attack (among other active attacks), they require a higher computational overhead compared to the one of HB⁺-based protocols.

Like HB-based protocols, distance-bounding (DB) protocols [9] is also a large research area within the broader RFID security

* Corresponding author.

E-mail address: ayang3-c@my.cityu.edu.hk (A. Yang).

community. DB protocols are meant to combat relay attacks (MiM attack where data is not modified) by enabling a verifier – measuring the time-of-flight of the messages exchanged between the verifier and an untrusted prover – to calculate an upper bound on the distance from the prover. DB protocols were initially proposed by Brands and Chaum [10] to combat relay attacks. Subsequently many DB protocols have been proposed, e.g., [11–18], along with frameworks for further security analysis, e.g., [19,18], location privacy concerns e.g., [20], optimal selection of parameters [21,22] and protocol vulnerabilities, e.g., [23–28].

Although some issues has been raised about the feasibility of implementing HB-based protocols for ultra-constrained devices [29] it is still considered a useful ‘lightweight’ protocol approach suitable for more constrained environments. One of the main challenges in designing secure authentication protocols for resource-constrained devices that rely on the HB⁺ protocol remains the resistance against MiM attacks. Meanwhile, distance-bounding protocols are considered to be one of the main countermeasures against simple MiM (relay) attacks. To overcome this challenge, we propose that principles from *distance-bounding* (DB) protocols should be integrated into HB⁺-based protocols to build a novel distance-bounding/authentication protocol called HB⁺DB.

Contributions: There is a growing body of work on ‘physical-layer security’ but previous proposals for improving HB⁺-based protocols have solely focused on cryptographic solutions. The high-level aim of the paper is to demonstrate that simple non-cryptographic measures could be effective in improving cryptographic protocols, i.e., measuring the response time could be used to mitigate a protocol vulnerability instead of modifying the cryptographic response function.

There is a strong research interest in both HB⁺-based protocols and distance-bounding protocols. However, the potential interface between these two types of protocols has not been explored despite some similarities, e.g., multiple exchanges during protocol execution and single-bit responses. This paper combines principles of distance-bounding (detection of physical-layer communication delay) and HB⁺-based protocols (LPN-based response function) to construct a novel hybrid HB⁺DB protocol. This protocol provides resistance against the MiM attack affecting HB⁺-based protocols, and while not intended as such could also potentially serve as a lightweight distance-bounding protocol against basic threat scenarios. It therefore contributes new approaches to the research areas of both these protocols by starting a discussion on using physical-layer mechanisms to improve cryptographic protocols, specifically here HB⁺, and the effectiveness, or lack thereof, of using LPN as the basis of a distance-bounding protocol. We provide a theoretical security analysis for our proposed protocol, considering security mainly from the perspective of using it as an HB⁺ authentication protocol. For the sake of completeness, we also add an initial security discussion from the perspective of using HB⁺DB as a distance-bounding protocol.

We also consider the security of the protocol and effectiveness of a MiM attack in a more practical setting. We show through Matlab simulations that a communication channel with simple modifications to the receiver architecture could limit the success of a MiM attacker, and verify that this modification is feasible in practice by a practical implementation. We also demonstrate that the distance-bounding phase of our protocol could be effectively implemented using a novel distance-bounding channel design based on ISO 14443 compliant physical layer communication symbols and is therefore feasible in contactless smart card and NFC systems.

This article is a full version of our conference paper in 2015 [30]. Besides the initials results in [30], we add several additional contributions into this article. We expand the discussions on the effects of the errors and noise, the practical GRS threat,

and the Matlab simulation. We provide a more comprehensive security discussion in Section 4 on the basic distance-bounding performance. We give a comparison of some selected distance-bounding protocols, but the main focus of the paper remains on GRS resistance of HB⁺. We also practically implement our modified receiver architecture on a software-radio platform to test the bit-error rate in a more realistic setting, which is more challenging and subject to external influences not always taken into account in a simulation. The results show that our proposed modification is feasible in practice, even though the performance degrades slightly from the theoretical results. Furthermore, we propose and provide a proof-of-concept implementation of a practical distance-bounding channel for detecting MiM attackers based on an ISO 14443 Type B channel. The results show that it is realistic for a low resource device to reliably early detect a received bit, quickly calculate the response function and send a response while the final challenge bit is still being received. The novelty in this channel design is that a rapid response time is achieved through the use of a duplex channel, where the prover can send his response while the verifier is still transmitting the challenge. This channel could be used to implement any distance-bounding protocol.

2. Preliminaries

2.1. Notations

In the paper, k will be used for the length of the secret keys, the random nonces and the challenges exchanged. Vectors will be written in bold font, e.g., $\mathbf{a}, \mathbf{b} \in \{0, 1\}^k$, when necessary, to *number* the vectors we will use upper indexes surrounded by brackets: $\mathbf{a}^{(i)}$ and $\mathbf{b}^{(i)}$, $i \in \{1, \dots, n\}$. To refer to the j th component of a vector $\mathbf{b}$ ($\mathbf{a}^{(i)}$ resp.) we will use the notation $\mathbf{b}_j \in \{0, 1\}$ ($\mathbf{a}_j^{(i)} \in \{0, 1\}$ resp.). We denote by $\text{HW}(\mathbf{a}) = \sum_{j=1}^k \mathbf{a}_j$ the Hamming weight of a vector $\mathbf{a}$. For each bit $\mathbf{a}_j$, its flipped bit is $\bar{\mathbf{a}}_j = \mathbf{a}_j \oplus 1$. Honest entities in the protocols will be denoted as $\mathcal{V}$ (the verifier) and $\mathcal{P}$ (the prover). When the prover acts maliciously it will be denoted as $\mathcal{P}^*$. The attacker is referred to as $\mathcal{A}$. For a message m , we denote by m^* its tampered version. We denote by $\epsilon \sim \text{Be}(\alpha)$ a random variable ϵ with Bernoulli distribution of parameter $\alpha \in (0, 1)$, i.e., $P[\epsilon = 1] = \eta$, $P[\epsilon = 0] = 1 - \eta$, $\epsilon \in \{0, 1\}$. The binomial distribution is denoted by $\text{Binomial}(n, \alpha)$.

2.2. The HB and HB⁺ protocols

Hopper and Blum proposed an authentication protocol, known as HB [1] that was designed for “human authentication”. The HB protocol runs between a trusted verifier $\mathcal{V}$ and an untrusted prover $\mathcal{P}$ and requires that $\mathcal{V}$ and $\mathcal{P}$ share a pair of secret keys $\mathbf{x}, \mathbf{y} \in \{0, 1\}^k$. At each run of the protocol $\mathcal{P}$ sends a random *blinding factor* $\mathbf{b} \in \{0, 1\}^k$ to $\mathcal{V}$, waits for the random challenge $\mathbf{a} \in \{0, 1\}^k$, computes the response:

$$z = (\mathbf{a} \cdot \mathbf{x}) \oplus (\mathbf{b} \cdot \mathbf{y}) \quad (1)$$

and eventually sends z back to $\mathcal{V}$.

However, the HB authentication protocol [1] was found vulnerable to an active attack aiming at recovering the secret keys shared between a verifier $\mathcal{V}$ and a prover $\mathcal{P}$ [2]. More precisely, after collecting $2k$ linearly independent equations of the form (1), an adversary $\mathcal{A}$ is able to determine $\mathbf{x}$ and $\mathbf{y}$ by simply running the Gaussian elimination algorithm. In order to make the protocol resistant to this attack Juels and Weis introduced the HB⁺ variant [2] (depicted in Fig. 1), which was proposed as an authentication protocol appropriate for RFID tags, secure against passive eavesdroppers. The idea of [2] is to introduce some *errors* in the computation of the responses, in order to mask the exact

Fig. 1. One round of the HB^+ protocol. The entire authentication requires n rounds. The prover $\mathcal{P}$ is authenticated if it fails in less than some threshold t rounds.

value of z . More precisely, in [2] the response is computed in the following way: $(a \cdot x) \oplus (b \cdot y) \oplus \epsilon$, where $\epsilon \sim \text{Be}(\eta)$, i.e., $P[\epsilon = 1] = \eta$, $P[\epsilon = 0] = 1 - \eta$, $\epsilon \in \{0, 1\}$. In this way, to solve the noisy linear system collected during the eavesdropping attack would imply to find a solution to a *learning with parity noise* (LPN) problem [3], which is proven to be $\mathcal{NP}$ -hard. The LPN-problem is defined as follows [3]:

Definition 1 (LPN-Problem). Let $\mathbf{A}$ be a random $n \times k$ binary matrix, $\mathbf{x}$ a random k -bit vector, $\eta \in (0, \frac{1}{2})$ a constant noise parameter, and $\mathbf{v}$ a randomly distributed k -bit vector such that $\text{HW}(\mathbf{v}) \leq \eta n$. Given $\mathbf{A}$, η , and $\mathbf{z} = (\mathbf{A} \cdot \mathbf{x}) \oplus \mathbf{v}$, find a k -bit vector $\mathbf{x}'$ such that $\text{HW}(\mathbf{A} \cdot \mathbf{x}' \oplus \mathbf{z}) \leq \eta n$.

Katz and Shin [31] proved the parallel concurrent security property of the HB and the HB^+ -based protocols and their analysis implied meaningful security for $\eta < \frac{1}{4}$, which Katz and Smith [32] later on extended to any arbitrary $\eta < \frac{1}{2}$.

The BKW [33] algorithm, was at first considered as the most efficient algorithm to solve the LPN problem. However, Gilbert et al. [8] have shown that in the context of BKW algorithm a security level with at most 52 bits can be achieved for a secret key of length $k = 512$ and $\eta = 0.25$.

The main drawback of the HB-based protocols is that they are vulnerable to the GRS attack [4]. There exists however a variant, called $\text{RANDOM-HB}^\#$ [8], which is resistant to it. The strength of $\text{RANDOM-HB}^\#$ lies on the fact that the secret keys are binary matrices $\mathbf{X}$, $\mathbf{Y}$ of dimensions $(k_X \times n)$ and $(k_Y \times n)$ respectively. Thus, the $\text{RANDOM-HB}^\#$ protocol is essentially equivalent to n iterations of the HB^+ , where a new pair of keys is used at each repetition. This also implies that a larger storage is needed, and that more complex computations are required (e.g., vector–matrix product, instead of vector–vector inner-product). Additionally, for the $\text{RANDOM-HB}^\#$, the response z is itself a vector. Gilbert et al. already pointed out that the storage costs of $\text{RANDOM-HB}^\#$ are insurmountable for a constrained device, and therefore propose the $\text{HB}^\#$ protocol, which is based on fewer matrices (the Toeplitz matrices) and is also secure against the GRS attack. In this paper, we decided to lighten the storage and computational requirements of the $\text{RANDOM-HB}^\#$ in a different way: we propose a distance-bounding protocol in which the responses are based on the HB^+ protocol. Based on this choice, at each run of the distance-bounding protocol, we obtain n (number of rounds) repetitions of the same HB^+ protocol. This corresponds to considering the sub-case of the $\text{RANDOM-HB}^\#$ where all columns of $\mathbf{X}$ ($\mathbf{Y}$ respectively) are equal, and gain all the *benefits* (including resistance against GRS and MiM attacks) that come with distance-bounding.

2.3. Distance-bounding protocols

The key concept in distance-bounding protocols (DB) is to equip a verifier $\mathcal{V}$ with a clock and to measure the time-of-

flight of the challenge-responses exchanged between $\mathcal{V}$ and an untrusted prover $\mathcal{P}$. The timing of the exchanged challenge-responses enables $\mathcal{V}$ to bound the physical distance of the untrusted prover $\mathcal{P}$. DB protocols are mainly composed of three phases: an *initialization* phase, a *distance-bounding* phase, and a *verification* phase.

Initialization phase: This phase is not time critical and is used by the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ to exchange the messages necessary to setup the variables/nonces to be used in the subsequent phase(s).

Distance-bounding phase: This phase is time critical and usually involves a *rapid-bit exchange*, which means a sequence of challenge-responses performed at maximum bit rate. The number n of rounds (challenges-responses) is a security parameter. The verifier $\mathcal{V}$ selects n unpredictable challenges, starts a clock and sends the challenges to the prover $\mathcal{P}$. Upon receiving the challenge, the prover $\mathcal{P}$ generates the response using a response function that is connected to the secret information¹ shared between $\mathcal{P}$ and $\mathcal{V}$ and sends it back to $\mathcal{V}$. As soon as the response reaches the verifier, $\mathcal{V}$ stops the clock and stores the time Δt_i where $i \in \{1, \dots, n\}$ that corresponds to the time taken between sending a challenge and receiving a response.

Verification phase: After the *distance-bounding* phase, a final *verification* phase may be required during which some more non-time critical exchange of messages might be performed. The result of this final phase depends on the correctness of the received responses and on the estimated distance between $\mathcal{V}$ and $\mathcal{P}$.

When distance-bounding protocols are implemented in RFID tags, the upper bound on the distance of the prover $\mathcal{P}$ is computed based on the stored times Δt_i . Since the messages are sent as radio waves (and travel at the speed of light, c), the distance between $\mathcal{P}$ and $\mathcal{V}$ is upper bounded by $c \cdot \frac{t_{\max}}{2}$, where $t_{\max}$ is the maximum delay time between sending a challenge and receiving a response.

2.4. Threat model

In this paper, we propose a novel distance-bounding protocol that relies on the HB^+ protocol. Thus, we consider the main threats against DB protocols – distance fraud, mafia fraud, terrorist fraud – as well as the main threat against the HB^+ protocol known as GRS attack [8].

Distance fraud: In this attack, a legitimate but dishonest prover $\mathcal{P}^*$ located far away from a verifier $\mathcal{V}$ is trying to fool the latter into thinking that she is in $\mathcal{V}$'s proximity [10].

Mafia fraud: This attack involves an adversary $\mathcal{A}$, a legitimate prover $\mathcal{P}$, and verifier $\mathcal{V}$. $\mathcal{P}$ is located far away from $\mathcal{V}$ and $\mathcal{A}$ is trying to shorten this distance, while $\mathcal{P}$ is unaware of the attack [34].

Terrorist fraud: In this case, similarly to the mafia fraud three entities are involved: a verifier $\mathcal{V}$, a prover (outside $\mathcal{V}$'s proximity), and an adversary $\mathcal{A}$. Again $\mathcal{A}$ is trying to shorten the distance between the prover and the verifier. The main difference is that the dishonest prover denoted as $\mathcal{P}^*$ is actually aware of the attack and is colluding with $\mathcal{A}$ to help her succeed in her authentication attempt. The main restriction is that $\mathcal{P}^*$ should not provide to $\mathcal{A}$ any advantage in being authenticated in a subsequent run of the protocol on her own without $\mathcal{P}^*$'s help [34].

Many variations of these threats were proposed in the literature. For example, a *distance-hijacking* [35] attack considers the case where a far away dishonest prover exploits some

¹ The majority of the proposed DB protocols in the literature use the secret key model (i.e. a secret key shared between a prover and a verifier). However, there are a few protocols that rely on a public key model. In both cases, the main in structure of the protocol is very similar.

Fig. 2. The GRS attack against one round of the HB^+ protocol.

honest, active provers in the verifier's proximity in order to be authenticated by the verifier. An *impersonation fraud* [36] considers the case where an adversary $\mathcal{A}$ attempts to impersonate a prover $\mathcal{P}$ to a verifier $\mathcal{V}$. A *man-in-the-middle* [37,18] attack was introduced as a generalization of mafia fraud attacks and considers that during a *learning phase* the adversary $\mathcal{A}$ interacts in parallel with multiple provers and verifiers. Finally, a *collusion fraud* [37,18] was introduced as a generalization of terrorist fraud attacks, and in this case the adversary's goal is to mount a successful man-in-the-middle attack. In this paper, we focus on the most well known threats in DB protocols and investigate the security of the proposed protocol against: distance fraud, mafia fraud and terrorist fraud.

Finally, we describe the main attack against the HB^+ protocols proposed by Gilbert et al. and known as GRS attack [8]:

GRS attack: Gilbert et al. described a very simple active attack against the HB^+ protocol. In this attack, as depicted in Fig. 2, an adversary $\mathcal{A}$ manipulates the challenges sent by $\mathcal{V}$ to $\mathcal{P}$ and observes whether such manipulation or modification leads to authentication success or failure. More precisely, $\mathcal{A}$ chooses a constant k -bit vector d and uses it to modify the sent challenges a , i.e., d is XORed with each challenge a for each of the rounds of the authentication. We should note here that the same d is used in all rounds of the HB^+ protocol. The attack succeeds with high probability when $d \cdot x = 0$ and fails when $d \cdot x = 1$. Acceptance or failure of the authentication reveals one bit of the secret key x . In order to recover the whole secret key x , $\mathcal{A}$ needs to run k times the protocol (all rounds of the protocol) for linearly independent d values, and solve the resulting system. As soon as x is recovered $\mathcal{A}$ may easily impersonate $\mathcal{P}$ by setting $b = 0$ or by employing a similar strategy as the one described above to recover y . While theoretically the GRS attack is quite simple to mount from a practical perspective, it is quite challenging to perform, especially when timing constraints are added in the authentication process: this is also demonstrated in Section 5.

3. The HB^+ DB protocol

We propose a new protocol that relies on the structure of the HB^+ protocol that we call HB^+ distance-bounding protocol (HB^+ DB). We consider an honest verifier $\mathcal{V}$ and an untrusted prover $\mathcal{P}$. $\mathcal{V}$ and $\mathcal{P}$ share three secret keys $x, y, z \in \{0, 1\}^k$ as well as a secret real number $\eta \in (0, 1)$, which gives the error probability introduced by the prover $\mathcal{P}$ in the challenge-response phase.

The protocol is discriminated into three main phases: (i) an *initialization phase*, (ii) a *distance-bounding phase*, and (iii) a *verification phase* as depicted in Fig. 3.

Initialization phase: In this phase the prover $\mathcal{P}$ starts the initialization by sending a uniformly random generated nonce $s \xleftarrow{R} \{0, 1\}^k$ to the verifier $\mathcal{V}$. The nonce s is then used, together with the seed z , to evaluate the PRF $f: \{0, 1\}^k \times \{0, 1\}^k \rightarrow \{0, 1\}^k$. We should note here that both the prover and the verifier need to compute nk -dimensional binary vectors denoted as $b^{(i)}$, where $i \in \{1, \dots, n\}$, using the PRF f_z and the nonce s . This could be

Fig. 3. The HB^+ DB protocol. In the Verification phase $\mu = n\alpha$ is the (expected) mean of the sum of the Bernoulli random variables $(a^{(i)} \cdot x) \oplus (b^{(i)} \cdot y) \oplus r_i$. In the noiseless case $\alpha = \eta$.

done for example via $f_z^{(i)}(s) := (\bigcirc_{j=1}^i f_z)(s) =: b^{(i)}$, or $f_z^{(i)}(s) := f_z(s + i) =: b^{(i)}, \forall i \in \{1, \dots, n\}$.

Furthermore, in order to lighten the computations that the prover $\mathcal{P}$ needs to perform during the time-critical phase, $\mathcal{P}$ computes n vectors $c_i = (b^{(i)} \cdot y) \oplus \epsilon_i \in \{0, 1\}$, where $\epsilon_i \in \{0, 1\}$ such that $\mathbb{P}[\epsilon = 1] = \eta$. Additionally, the verifier $\mathcal{V}$ generates uniformly at random nk -bit vectors $a^{(i)} \xleftarrow{R} \{0, 1\}^k$.

Distance-bounding phase: The distance-bounding phase is time-critical and it is repeated n times (rounds). At each round i where $i \in \{1, \dots, n\}$ the verifier $\mathcal{V}$ sends to the prover $\mathcal{P}$ a vector $a^{(i)}$. Upon receiving $a^{(i)}$, the prover $\mathcal{P}$ answers with $r_i = (a^{(i)} \cdot x) \oplus c_i \in \{0, 1\}$ (i.e. $r_i \in \{0, 1\}$). The verifier $\mathcal{V}$ records the responses r_i together with the time Δt_i that elapsed from the moment $a^{(i)}$ was sent to the moment r_i was received.

Verification phase: The verifier $\mathcal{V}$ accepts a prover $\mathcal{P}$ if the following conditions hold:

1. The received responses r_i are 'correct'. More precisely, in order to check the correctness of the responses $\mathcal{V}$ computes $\tilde{r}_i := (a^{(i)} \cdot x) \oplus (b^{(i)} \cdot y)$. It is clear that due to the error probability η , it will always hold $\tilde{r}_i \oplus r_i = \epsilon_i$, where $\text{HW}(\epsilon) \sim \text{Binomial}(n, \eta)$. In other words, $\mathcal{V}$ accepts if $(a^{(i)} \cdot x) \oplus (b^{(i)} \cdot y) = r_i$ for at least $n(1 - \eta)$ equations.
2. The prover $\mathcal{P}$ is 'close enough', hence $\Delta t_i \leq 2t_{\max}$, where $t_{\max}$ denotes the maximum time required to transmit a message from $\mathcal{V}$ to $\mathcal{P}$.

We should note here that the threshold $t_{\max}$ must take into account the time needed to perform the computation $(a^{(i)} \cdot x) \oplus c_i$, which obviously depends on the length of the random nonces $a^{(i)}$.

3.1. Errors and noise

As depicted in Fig. 3, the HB^+ DB protocol requires the prover $\mathcal{P}$ to introduce *intentional errors* (the terms $\oplus \epsilon_i$), when computing

the responses r_i to the challenges $\mathbf{a}^{(i)}$. This is necessary in order to make the protocol resistant against eavesdropping attacks that aim at retrieving the secret key(s) [2]. Since the answers computed by an honest prover $\mathcal{P}$ will *always* contain *some* errors, the verifier $\mathcal{V}$ should accept even if $r_i \neq (\mathbf{a}^{(i)} \cdot \mathbf{x}) \oplus (\mathbf{b}^{(i)} \cdot \mathbf{y})$, for *some* $i \in \{1, \dots, n\}$. Actually $\mathcal{V}$ should not accept $\mathcal{P}$ when the responses do not contain *enough* errors (otherwise the LPN-security is lost). In order to make these statements more rigorous and to proceed with the security analysis we have to define how tolerant $\mathcal{V}$ should be in the verification phase. We do it by modeling the errors introduced by $\mathcal{P}$ as independent Bernoulli random variables with the same parameter η , i.e. $\epsilon_i \sim \text{Be}(\eta)$ for all $i = 1, \dots, n$. The total number of errors (coming from the LPN-security) in a full run of the protocol is thus $S = \sum_{i=1}^n \epsilon_i$. Let τ denote the tolerance that $\mathcal{V}$ has in the verification phase, and $\mu = n\eta$, then the probability of false rejection is given by $\mathbb{P}_{\text{FR}} = P[|S - \mu| \geq \tau]$. Since the random variable $S \sim \text{Binomial}(n, \eta)$ has a binomial distribution, the Hoeffding inequality (see [Appendix](#) and [Lemma 1](#)) applies and yields the following bound:

$$\mathbb{P}_{\text{FR}} \leq 2e^{-\frac{2\tau^2}{n}}. \quad (2)$$

Solving the inequality (2) with respect to the tolerance τ , we obtain a bound on τ according to the desired rate of false rejection: $\tau \leq \sqrt{\frac{n \log(2/\mathbb{P}_{\text{FR}})}{2}}$.

On the other hand, the probability of false acceptance $\mathbb{P}_{\text{FA}}$ depends also on the chosen value of the threshold τ . The false acceptance rate is directly connected to the probability of success in each of the frauds against DB protocols, and will be computed in detail in Section 4.

Note that, we adopt a different acceptance policy than the one used by Gilbert et al. [8]. More precisely, in [8] a tag is accepted if the Hamming weight of the vector of errors satisfies: $\text{HW}(\mathbf{a} \cdot \mathbf{X} \oplus \mathbf{b} \cdot \mathbf{Y} \oplus \mathbf{z}) \leq t$, where the threshold is $t = un$ and $u \in (\eta, 1/2)$. This choice is a bit loose, since the Hamming weight of the vector of errors follows a binomial distribution, the Gilbert et al.'s solution [8] provides a lower false rejection probability but also a higher false acceptance probability than our acceptance policy.

4. Security analysis

In this section, we investigate the resistance of the proposed HB^+DB protocol against the main threats to distance-bounding: distance fraud, mafia fraud and terrorist fraud. We will also deal with the GRS here from a theoretical perspective, while commenting further on the practical implications of such an attack implementation in Section 5.

We assume that the vectors $\mathbf{b}^{(i)}$ and $\mathbf{a}^{(i)}$ are randomly distributed, i.e., $\mathbf{a}^{(i)}, \mathbf{b}^{(i)} \sim \text{Be}(\frac{1}{2})$ for all $i \in \{0, 1\}$. This hypothesis can be relaxed by requiring only the distribution of the vectors $\mathbf{a}^{(i)}$ being unknown (or uniform) to the attacker and to the malicious prover. By definition (in the HB^+DB protocol) the errors introduced by the prover in the responses are independent Bernoulli random variables of same parameter $\eta < \frac{1}{2}$, i.e., $\epsilon_i \sim \text{Be}(\eta)$ for all $i \in \{1, \dots, n\}$. On the same line, we model the channel-noise as $\delta_i \sim \text{Be}(\nu)$ for all $i \in \{1, \dots, n\}$ and $\nu < \frac{1}{2}$. Furthermore, we assume that the noise in the channel is independent from the error coming from the LPN-security, that is, the variables ϵ_i and δ_i are independent. Under this notation, the final response-bit that reaches the verifier $\mathcal{V}$ at each round of the distance-bounding phase is: $r_i = (\mathbf{a}^{(i)} \cdot \mathbf{x}) \oplus (\mathbf{b}^{(i)} \cdot \mathbf{y}) \oplus \epsilon_i \oplus \delta_i$. In order to authenticate the prover $\mathcal{P}$, $\mathcal{V}$ will check the distribution of $r_i \oplus (\mathbf{a}^{(i)} \cdot \mathbf{x}) \oplus (\mathbf{b}^{(i)} \cdot \mathbf{y}) = \epsilon_i \oplus \delta_i$. It is easy to check that the random variables $\omega_i = \epsilon_i \oplus \delta_i$ are independent and identically distributed as Bernoulli of parameter $\alpha = \eta + \nu - 2\eta\nu$. Thus, we will directly consider $\omega_i \sim \text{Be}(\alpha)$ as the

total noise, and use $\mu = n\alpha$ as its mean. Note that, however, the theoretical discussion holds also in the case of a noiseless channel by replacing α with η (i.e., setting $\nu = 0$), as both the ω_i and the ϵ_i have a Bernoulli distribution.

Distance fraud: As, by hypothesis, the challenges $\mathbf{a}^{(i)}$ appear completely random to the malicious prover $\mathcal{P}^*$, the (to be computed) answers will still be random to $\mathcal{P}^*$. Thus, the *best* strategy for the malicious prover is to early-send a random bit $r_i^* \sim \text{Be}(1/2)$ as answer. The probability to succeed in distance fraud therefore equals the probability of false acceptance $\mathbb{P}_{\text{FA}}$ when the error² $\zeta_i = r_i \oplus r_i^* \sim \text{Be}(1/2)$. Let $\Delta = (1/2 - \alpha - \tau/n)$ and $S_n = \sum_{i=1}^n \zeta_i$, then Hoeffding's inequality (see [Appendix A](#)) yields to:

$$\begin{aligned} \mathbb{P}_{\text{DF}} &= \mathbb{P}(\mu - \tau \leq S_n \leq \mu + \tau) \\ &\leq \mathbb{P}(S_n \leq \mu + \tau) = \frac{1}{2^n} \sum_{i=0}^{\mu+\tau} \binom{n}{i} \\ &\leq \mathbb{P}(S_n \leq \mathbb{E}[S_n] - \mathbb{E}[S_n] + n\alpha + \tau) \\ &\leq \mathbb{P}(S_n \leq \mathbb{E}[S_n] - n\Delta) \\ &\leq e^{-2n\Delta^2}. \end{aligned}$$

Mafia fraud: In distance-bounding literature in general, there is a common assumption made as to the capabilities of the mafia fraud (relay) attacker, in that $\mathcal{A}$ can simply forward messages and thus incurs no delay other than additional propagation time. This additional propagation time will be detected by the verifier so the attacker needs another approach. To perform a mafia fraud attack, $\mathcal{A}$ simply relays $\mathbf{s}$ during the initialization phase and attempts to send early answers in the time-critical phase. $\mathcal{A}$ can use two main strategies: (i) *anticipate the challenges*, and (ii) *guess the responses*. The former strategy concerns the communication between $\mathcal{A}$ and $\mathcal{P}$. More precisely, $\mathcal{A}$ guesses the challenge before having received it from $\mathcal{V}$. If $\mathcal{A}$ guesses the correct challenge, which she can verify when she receives (later on) the correct $\mathbf{a}^{(i)}$ sent by $\mathcal{V}$, she just forwards $\mathcal{P}$'s response r_i . Since we have multi-bit challenges $\mathcal{A}$'s probability to guess the correct challenge is 2^{-k} at each round, where k denotes the length of the challenge. On the other hand, the *guess the responses* strategy concerns the communication between $\mathcal{A}$ and $\mathcal{V}$. The probability of guessing *correctly* the value of r_i is 2^{-1} at each round. Note that the clever adversary would run both strategies consecutively. More precisely, $\mathcal{A}$ would try to *guess the responses* only if she failed to anticipate the *correct* challenge. The probability to guess correctly the response at one round of the HB^+DB protocol is $\gamma = 2^{-k} + (1 - 2^{-k})(1/2) = 1/2 + (1/2)^{k+1}$. Let r_i^* denote the response sent by $\mathcal{A}$ to $\mathcal{V}$, then, $\zeta_i = r_i^* \oplus r_i \sim \text{Be}(1 - \gamma)$. Consider $\Delta' = (1 - \gamma - \alpha - \tau/n)$ and $S_n = \sum_{i=1}^n \zeta_i$, then we have:

$$\begin{aligned} \mathbb{P}_{\text{MF}} &= \mathbb{P}(\mu - \tau \leq S_n \leq \mu + \tau) \\ &\leq \mathbb{P}(S_n \leq \mathbb{E}[S_n] - \mathbb{E}[S_n] + \mu + \tau) \\ &\leq \mathbb{P}(S_n \leq \mathbb{E}[S_n] - n(1 - \gamma - \alpha - \tau/n)) \\ &\leq e^{-2n\Delta'^2}. \end{aligned} \quad (3)$$

Terrorist Fraud: We remind that in this attack the adversary is located close to the verifier $\mathcal{V}$ and tries to shorten the distance between the prover and the verifier. In this case though, the prover is dishonest, denoted by $\mathcal{P}^*$ and colludes with the attacker $\mathcal{A}$.

² The total error (including transmission noise) is still a Bernoulli random variable of parameter $1/2$. It is possible to justify this fact as follows: XORing a bit-noise with a random bit will always result in a random bit, indeed the random variable obtained by XORing two Bernoulli of parameters ν (channel noise) and $1/2$ (random guess) respectively is a Bernoulli random variable of parameter $1/2 = 1/2 + \nu - 2(\nu \cdot 1/2)$.

This collusion implies that $\mathcal{P}^*$ is willing to give $\mathcal{A}$ some pieces of information in order to enable $\mathcal{A}$ to authenticate as the prover $\mathcal{P}^*$, but the knowledge acquired by $\mathcal{A}$ should not reveal any secret shared between $\mathcal{P}^*$ and $\mathcal{V}$. $\mathcal{P}^*$ in order to help $\mathcal{A}$ may disclose to her c_i , $\forall i \in \{1, \dots, n\}$. This implies that the adversary should be able to guess correctly $\mathbf{a}^{(i)} \cdot \mathbf{x}$ for which the success probability is $\frac{1}{2}$ in each round and in order to pass this run of the protocol her success probability is the same with the one in a mafia fraud attack. Thus, $\mathcal{A}$ does not have a significant advantage to pass later on a new run of the protocol by herself. In other words, this leakage of information (i.e., disclosing c_i) is not harmful, since $\mathcal{A}$ cannot re-use the c_i -s in other runs of the protocols.³ The central matter is the following: in order to defeat $\mathcal{V}$, $\mathcal{A}$ must be able to compute the value $\mathbf{a}^{(i)} \cdot \mathbf{x}$ for any possible incoming challenge $\mathbf{a}^{(i)}$. Thus, $\mathcal{P}^*$ should give the secret key $\mathbf{x}$ to $\mathcal{A}$. However, once the attacker knows $\mathbf{x}$ she can mount successful mafia frauds. Indeed, $\mathcal{A}$ can pre-ask $\mathcal{P}^*$ with some random challenges $\mathbf{a}^{*(i)}$, and get the values c_i from the received answers $r_i^* (c_i = r_i^* \oplus \mathbf{a}^{*(i)} \cdot \mathbf{x})$. Now the attacker is able to compute the correct answer $r_i = \mathbf{a}^{(i)} \cdot \mathbf{x} \oplus c_i$ for any challenge $\mathbf{a}^{(i)}$ sent by the verifier $\mathcal{V}$. However, according to the terrorist-fraud definition [19] the dishonest prover $\mathcal{P}^*$ should not disclose to $\mathcal{A}$ any secret information. Thus, for the HB⁺DB protocol, the knowledge of $\mathbf{x}$ is mandatory in order to (successfully) pass the protocol and the success of a terrorist-fraud attack is similar to the one given in Eq. (3).

GRS Attack: The GRS attack is a type of MiM attack in which the adversary $\mathcal{A}$ is placed in-between the prover and the verifier, and can relay (part of the) messages – as in a mafia fraud attack – as well as modify the value of the sent challenges on-the-fly. Recall that the goal of the GRS attack is to retrieve the secret key $\mathbf{x}$. In theory, this is achieved by XORing all the challenges sent by $\mathcal{V}$ with a vector $\mathbf{d}$, which is different at each run of the protocol (but the same for the n rounds of the distance-bounding phase). The $\mathbf{d}$ -s must be k linearly independent vectors. In practice (and to maximize the success probability), $\mathcal{A}$ will choose the $\mathbf{d}$ -s to be the vectors of the canonical basis.⁴ Thus, in order to get the j th bit of the key $\mathbf{x}$, the attacker has to first *learn* the value of the j th bit of each challenge (i.e., $\mathbf{a}_j^{(i)}$) and then *flip* it (to $\bar{\mathbf{a}}_j^{(i)}$), and succeed in making the prover receive the modified bit $\bar{\mathbf{a}}_j^{(i)}$.

At first it would seem reasonable to consider $\mathcal{A}$ with the same capability as a mafia fraud attacker, i.e., no other delay than the additional propagation. However, there is a significant difference between a mafia fraud and a GRS attack. From a practical point of view, a mafia fraud could be implemented simply by taking the verifier $\mathcal{V}$ signal, routing it to an RF up-mixer, transmitting the signal, routing the received signal to an RF down-mixer and then sending it to the prover $\mathcal{P}$ as proposed in [38]. If we assume that the delay of all these attack hardware components are zero (only possible from a theoretical perspective) then we have a system approximating the envisaged mafia fraud attacker. For the GRS attack the attacker $\mathcal{A}$ has to learn the value of the bit first before being able to forward the modified bit. This means the attacker has to wait for the j th bit to begin to be transmitted, wait for a short while⁵ (sample the bit), decide the value of the bit $\mathbf{a}_j^{(i)}$ and then transmit the flipped bit to $\mathcal{P}$. This extra time taken to sample and transmit the flipped version would have to induce additional delay, which would be detected by the verifier $\mathcal{V}$. In practice, the time the attacker needs to wait also influences her ability to accurately sample the bit value and present the correct

flipped value (as we will demonstrate in Section 5). The GRS attack does not succeed due to the *incorrectness* of the responses but because of a failed distance-bound (i.e., $\Delta t_i > t_{\max}$ for the i th round). Thus the attacker is not able to learn anything about the bit(s) of the key $\mathbf{x}$ from the fact that the protocol failed.

4.1. Comparison with selected distance-bounding protocols

Even though the main objective of the protocol is to resist the GRS attack, we compare our proposed HB⁺DB protocol with some well-known distance-bounding protocols as shown in Table 1. It can be seen that HB⁺DB protocol offers resistance against the three basic attacks to distance-bounding protocols, and is also resistant to the GRS MiM key-recovery attack which is a weakness of basic HB-based protocols. There might be more efficient distance-bounding attacks stemming from the use of the LPN response but in depth analysis of the protocol from that perspective is left to future work, with this paper mostly concerned with whether HB⁺DB could be an practically GRS resistant HB⁺protocol. Previous HB-based protocols try to prevent the GRS attack by modifying the cryptographic response functions, while we propose to detect the attacker by accurately timing the challenge-response exchanges, between prover and verifier, like one would do in a distance-bounding protocol. Given that this protocol is essentially the original HB⁺protocol simply run like a distance-bounding protocol, i.e., with timed exchanges, there is no additional burden on the constrained prover, and thus is as efficient as the original HB⁺ protocol.

5. Practical considerations

The purpose of this experimental evaluation is to show that the active MiM GRS attack proposed by Gilbert et al. [4] (described in Section 2.4) requires the attacker $\mathcal{A}$ to wait for a noticeable period before she can determine the bit value (of the challenge sent by $\mathcal{V}$) and transmit a modified bit. We also show that the attacker's success probability can be significantly reduced in practice by modifying the prover's receiver architecture. This modification effectively limits the attacker $\mathcal{A}$ while having minimal impact on the reliability of legitimate communication between the verifier $\mathcal{V}$ and the prover $\mathcal{P}$.

We remind the reader that if the GRS attack is applied in the proposed HB⁺DB protocol, the adversary $\mathcal{A}$ would need to manipulate the challenges sent by the verifier $\mathcal{V}$ to the prover $\mathcal{P}$ during the *distance-bounding* phase, and then to check whether this manipulation results in an authentication failure or success. To launch the attack, $\mathcal{A}$ must successfully flip the chosen bit during each round and also pass the distance-bound, i.e., introduce no additional delay. The time $\mathcal{A}$ takes to detect the bit value and retransmit the flipped value will be detected by $\mathcal{V}$. To introduce no additional delay $\mathcal{A}$ therefore has to follow the *early-detect* and *late-send* strategy proposed in [39]. This strategy exploits common low-resource receiver architecture, where the prover $\mathcal{P}$ is assumed to sample multiple times (or integrate) across the entire bit-period π . We note here that the bit-period π corresponds to the time period needed to transmit a bit. $\mathcal{A}$ attempts to learn the value of the bit (sent by $\mathcal{V}$) within a short duration of the start of π (the time the attacker allows himself here to learn the bit value is the *early* part of the bit period) and then would start sending the modified bit value later during π . Thus, $\mathcal{P}$ receives at first the original bit value, and later during π the modified bit value. $\mathcal{A}$ succeeds if $\mathcal{P}$ decodes the sent bit as the modified value. $\mathcal{A}$ also has to execute this operation within the confines of the transmitted π , so that she will not introduce any additional delay into the round-trip-time.

³ They depend on a nonce s and on an error ϵ_i , different at each run of the protocol.

⁴ This means that the first $\mathbf{d}$ will be $(1, 0, \dots, 0)$, the second will be $(0, 1, 0, \dots, 0)$, and the last $(0, \dots, 0, 1)$.

⁵ It could theoretically be a minuscule time but she has to wait.

Table 1
Comparison of selected distance-bounding protocols.

Protocols	DF	MF	TF	Response function
BC [10]	$(\frac{1}{2})^n$	$(\frac{1}{2})^n$	$\times$	1-bit lookup and $\oplus$
HK [12]	$(\frac{3}{4})^n$	$(\frac{3}{4})^n$	$\times$	2-bit lookup
Reid [13]	$(\frac{3}{4})^n$	$(\frac{3}{4})^n$	$(\frac{3}{4})^n$	2-bit lookup
Yang et al. [14]	$(\frac{3}{4})^n$	$(\frac{3}{4})^n$	$\times$	3-bit lookup
Zhuang et al. [15]	$(\frac{3}{4})^n$	$(\frac{3}{4})^n$	$(\frac{7}{8})^n$	3-bit lookup
S-K [23]	$(\frac{3}{4})^n$	$(\frac{1}{2})^n$	$(\frac{3}{4})^n$	2-bit lookup
SKI _{PRO} [37]	$(\frac{3}{4})^n$	$(\frac{3}{4})^n$	$(\frac{2}{3})^n$	3-bit lookup
HB ⁺ DB	$\leq (\frac{1}{e})^{2n\Delta^2}$	$\leq (\frac{1}{e})^{2n\Delta'^2}$	$\leq (\frac{1}{e})^{2n\Delta'^2}$	1-bit · and $\oplus$

5.1. Simulated MiM attack

In order to successfully pass the distance-bounding phase of the HB⁺DB protocol, $\mathcal{A}$ has to quickly determine the challenge $\mathbf{a}^{(i)}$ and retransmit the modified challenge $\mathbf{a}'^{(i)}$ to the verifier $\mathcal{V}$ within the remaining time of the bit-period π . This means that $\mathcal{A}$ has to early detect and guess the challenge $\mathbf{a}^{(i)}$ before receiving all parts of $\mathbf{a}^{(i)}$. We have performed a series of simulation experiments (in Matlab) in order to demonstrate the resistance of the HB⁺DB protocol to the GRS attack in a practical environment. To mount the GRS attack $\mathcal{A}$ adopts the following strategy. Without loss of generality, we describe the GRS attack in case of transmitting single bit C in distance-bounding phase. It is easy to see that the same procedure applies when the challenge is composed of k -bits. In the experiment, each bit is sampled by 1000 sampling points.

- (1) $\mathcal{V}$ sends out a 1-bit challenge C . To simulate the noisy environment, we add Gaussian noise to C . We assume that the environmental noise is identical for $\mathcal{V}$, $\mathcal{P}$ and $\mathcal{A}$.
- (2) $\mathcal{A}$ integrates over *early* part of the bit-period π to decode the challenge in advance and then modifies the guessed challenge to C' , and sends $C' + \text{noise}$ to $\mathcal{P}$. Note that $\mathcal{A}$ has to finish all these steps within one bit-period π , otherwise she will introduce additional delay and be detected by $\mathcal{V}$. The modification of the challenge intends to flip the remaining $(1 - \text{early})$ part of the challenge (0–1, and 1–0 resp). Although it is not trivial to do this in practice, we assume that $\mathcal{A}$ can instantly modify a transmitted 1 symbol to 0 and vice versa.
- (3) Upon receiving $C' + \text{noise}$, $\mathcal{P}$ integrates over the entire bit-period π .

If $\mathcal{A}$ does not guess the challenge correctly, then she will definitely fail in the attack. In other words, suppose the original challenge is 1, but $\mathcal{A}$ guesses it to be 0 and thus flips it to be 1. Then $\mathcal{P}$ will receive 1, which is identical to the original challenge. This means $\mathcal{A}$ fails to modify the challenge. If $\mathcal{A}$ guesses the challenge correctly, then she will succeed only if $\mathcal{P}$ receives the modified challenge correctly. If either of these conditions are not met, $\mathcal{A}$ fails to modify her chosen bit during this round and the GRS attack fails.

Fig. 4 illustrates the attack flow. For clarity, we set the challenge to be 10-bits long and the attacker $\mathcal{A}$ will try to flip all of them (this is not how the actual attack works but we wish to show how effective the attacker is at flipping bits). We choose SNR = 5 dB where SNR denotes Signal to Noise Ratio and *early* = 10%. The first plot (a) shows the original challenge sent by $\mathcal{V}$ and (b) is the noisy challenge influenced by the environment noise. The challenge guessed by $\mathcal{A}$ using early detection is represented in plot (c) with a blue triangle, while the blue square is the challenge received by $\mathcal{P}$ without $\mathcal{A}$ modifying it. The red line indicates the decision threshold—the receiver compares the result of the bit integration with this threshold to determine if a 1 or a 0 was received. A triangle/square above the threshold line indicates a 1 and below the line indicates a 0. From this we can clearly see

Fig. 4. Simulation of Gilbert et al.'s attack. SNR = 5 dB, *early* = 10%. Attacker fails to flip bits 1, 2, 3, 4, 7 and 10.

the effect of the time taken before deciding the bit value. The attacker makes mistakes in bits 1, 2, 3, 4, 7 and 10 while $\mathcal{P}$ would have gotten all the bits correct since $\mathcal{P}$ integrates over the whole bit period. Plot (d) describes the modified challenge by $\mathcal{A}$ and (e) shows the modified noisy challenge. The final plot (f) shows the modified challenge received by $\mathcal{P}$. Note that due to the failure to accurately detect the bit, $\mathcal{A}$ fails to flip 6 of the bits.

In our second experiment, we use this attack flow to test the attacker's success probability under various communication conditions. $\mathcal{P}$ uses a standard receiver that samples across 100% of the bit-period π . $\mathcal{A}$ is asked to try and flip 1000 randomly chosen bits. We test the attacker's success rate for different values of SNR and *early*. SNR varies in the range from −5 to 10 dB with step 1 and *early* is in the range from 0.05 to 1 with step 0.05. Fig. 5(a) shows the relationship between the attacker's success probability $\mathbb{P}$, SNR, and *early*, where

$$\mathbb{P} = \frac{\text{number of bits modified successfully}}{\text{the total number of bits}}.$$

We see that under any value of SNR, $\mathbb{P}$ increases to a summit (even to 1) first and then decreases. The increase is because as *early* increases, it gives $\mathcal{A}$ more time to guess the challenge which results in a bigger chance to obtain the correct challenge bit. This also means the attacker has a bigger chance to flip the challenge bit correctly. However, as *early* becomes larger, the attacker has less time to modify the challenge and influence $\mathcal{P}$ (note that essentially $(1 - \text{early})$ of the challenge is modified), and thus the success probability $\mathbb{P}$ decreases and quickly reaches a point after which the success probability $\mathbb{P}$ is less than 0.5. In practice, this will happen if $\mathcal{A}$ waits longer than half the bit-period π to make her decision since $\mathcal{A}$ would then have received half of the original signal already and a modified signal of less than 0.5 of the bit-period π would not be enough to fool the $\mathcal{P}$ to believe that the result is the opposite value.

Fig. 5. The attacker's success probability. Note that this is the success probability for each round of the time-critical (distance-bounding) phase. There are totally n rounds in the proposed HB^+DB protocol.

This means that the attacker has at most half of the bit-period π ($early \leq 0.5$) to do the early detection.

Given a conventional receiver architecture for $\mathcal{P}$, Fig. 5(a) shows that the attacker's success probability $\mathbb{P}$ is very large (up to 1) when SNR is high (e.g., 6–10) and with the $early$ from 0.2 to 0.4. Thus, it seems that $\mathcal{A}$ can indeed launch a GRS attack in a realistic low-resource communication environment. In order to solve this problem, we propose that $\mathcal{P}$ also adopts the *early-detect* strategy. In other words, $\mathcal{P}$ integrates over px ($0 < px < 1$) part of the bit-period π rather than 100% of π . This forces the attacker to finish the early detection within $px/2$ of the bit-period π , i.e., the value of $early$ will be smaller. However, in the meantime, we should also pay attention to the bit error rate (BER) of the prover ($\mathcal{P}_{\text{BER}}$), since now $\mathcal{P}$ will use less time to do the integration, and this may affect the value that $\mathcal{P}$ computes.

For our third experiment, we tested different values of px , the portion of the bit-period on which the attacker integrates. We set px in the region $0.1 < px < 1$ and looked for the value px_0 for which the corresponding $\mathcal{P}_{\text{BER}}$ is low enough (i.e., $\mathcal{P}_{\text{BER}} < 0.1$) and thus effectively reduces the success probability of the attacker. Fig. 5(b) depicts the success probability of the GRS attack when $px = 40\%$. We observe how it is possible to significantly reduce $\mathbb{P}$ while maintaining an acceptable value of $\mathcal{P}_{\text{BER}}$ as shown in Fig. 6. As an example, when $\text{SNR} > 5$ dB we have $\text{BER} = 0$ on the prover's side, while the attacker's chances of flipping a bit is $\mathbb{P} < 0.55$. This means that, when HB^+DB is implemented with $n = 16$ rounds the practical success probability of the GRS attack is $\mathbb{P} \approx 2^{-13}$, which is approximately the same as guessing a 4-digit PIN. In case of $\text{SNR} = 10$ dB and $early = 18\%$, the highest success probability of the attacker at each round is $\mathbb{P} = 0.6104$, while for $\mathcal{P}$ there is no bit error rate: $\text{BER} = 0$. When $n = 32$, the success probability of the GRS attack is only $\mathbb{P} = (0.6104)^{32} = 1.4 \times 10^{-7} \approx 2^{-23}$, which is negligible.

Fig. 5(b) illustrates the new relationship between $\mathbb{P}$, SNR and $early$, where $\mathcal{P}$ integrates over 40% of π (the bit-period). Fig. 6 shows the relationship between the bit error rate of $\mathcal{P}$ ($\mathcal{P}_{\text{BER}}$) and SNR with $px = 40\%$ and 100% , respectively. The result shows that the value of px does not affect too much on $\mathcal{P}_{\text{BER}}$ when SNR increases. In practice, SNR is usually larger than 5 dB. Therefore, our simulation demonstrates that the GRS attack could effectively be prevented in our HB^+DB protocol using this approach.

Our simulated results show that our approach could be effective in mitigating active man-in-the-middle attacks. Although Matlab simulations are useful for prototyping and testing ideas these

are also ideal approximations of what happens in practice. For example, it is easy to control the exact nature of the noise whereas in real life this variable is more unpredictable. To better understand performance in practice we also perform a practical experiment in the following sections. The second possible limitation is that the approach is also tested against our imagined optimal attacker, but it is still not clear what capability a real attacker could have. There has been practical relay implementations that introduce minimal delay [40], including 'active relay attacks' where selected bits are flipped with only a few μs extra delay [41]. The latter attack was executed on ISO 14443 Type A data (106 kbps with a bit period of $9.4 \mu\text{s}$). However, even when considering this fast attack implementations the modified receiver approach remains feasible. If the additional time incurred by the attacker is $x \mu\text{s}$, the honest prover would just need to set its evaluation period smaller than $2x \mu\text{s}$. For example, if the additional time introduced by the adversary is $2 \mu\text{s}$ the attack will fail if the prover evaluates the bit value in less than $4 \mu\text{s}$. This is approximately 40% of the bit period in ISO 14443. Although the time needed to flip the bit could be reduced it can never be instantaneous in practice. The attacker would always need to spend some time sampling the incoming bit first to see what the value is in order to flip it correctly. She can also never do it quite as reliably as the honest prover because even in idealized circumstances (no propagation or other communication-caused delay), the attacker only has half the time to determine the bit value compared to the honest prover. If the prover minimized its evaluation time to a point, where she is at the limit of where he will start receiving bit errors it is most likely that the adversary will be making errors given that he has much less time. The prover is therefore in control of setting a target for the adversary, which is always more difficult than her own recovery of the data.

5.2. Experimental implementation

In this experiment, we test the bit error rate using our proposed early-detect strategy over an actual radio channel, in which the receiver only integrates over parts of the bit period π and makes the decision. We assume the attacker and the legitimate prover use the same proposed receiver architecture, which makes integrating time cause the same impact on bit error rate for the attacker and the prover.

We use two USRP B200 devices [42] setup, as shown in Fig. 7, as the reader and the tag respectively. The USRP B200 is a single channel transceiver, with continuous RF coverage from 70 MHz to 6 GHz, and it works well with GNU Radio [43], which is a

Fig. 6. Results of simulated experiments in Matlab: the relationship between the Prover's BER with SNR when integrating over px of π . We can see that when $px = 40\%$ and $SNR = 2$ and 5 , $\mathcal{P}_{BER} = 0.0044$ and 0.0002 respectively. When $SNR \geq 7$, $\mathcal{P}_{BER} = 0$. When $px = 100\%$, and $SNR \geq -1$, $\mathcal{P}_{BER} = 0$.

Fig. 7. Experiment devices: USRP B200.

free and open-source software development toolkit that provides signal processing blocks to implement software radios and can be used with low-cost external RF hardware such as USRP to create software-defined radios. Each of the USRP devices is connected with a transmitter antenna and a receiver antenna.

We use Matlab to generate a random NRZ-encoded baseband signal with sampling rate 10 kHz. We export the signal as a binary file and load it into GNU radio. Note that both our two USRP devices are interacting with GNU radio. The USRP sender transmits the NRZ data at a rate of 100 kHz, with the data being amplitude modulated on a 860 MHz carrier, similar to a UHF RFID. We transmit the noisy data through the first USRP device and at the same time we receive the signal through the second USRP device. After demodulation of the carrier, the received signal is sampled at a rate of 500 kHz and stored in a binary file generated by GNU radio. Finally, we analyze the file in Matlab using the same data recovery method described in the previous section (the correlating receiver integrating over the specified portion of the bit period). The experiment was conducted in a conventional office environment, in presence of everyday electronic devices, lighting, etc. In order to obtain desirable low SNR values, we add additional noise to the transmitted signal using the noise block in GNU radio. Fig. 8 shows the experiment flow.

5.2.1. Results

We analyze the recorded noisy data in Matlab and compute the bit error rates (BER) for different integrating times, i.e., integrating over *early* part of the bit period π , and under different SNR values.

Fig. 8. Experiment procedure.

Fig. 9. Results of practical implementation: the relationship between the BER with the integrating time *early*, with $SNR = -5$ dB, 10 dB, and 15 dB, respectively.

The relationship between BER and *early* with different SNRs is illustrated in Fig. 9. Fig. 9 implies that it is feasible to modify the receive architecture in practice. For example, with $SNR = 15$ dB if the receiver integrates over 40% of π , the bit error rate is only 0.0175 for the receiver. In this higher SNR setting the attacker, who has to integrate over less than 20% of π , is made to decode the bit incorrectly with probability $BER = 0.4725$. We should qualify these results by adding that the effect of SNR on the BER is dependent on the channel implementation choices, like modulation data coding, so these results should be considered in the context of our experiment, rather than an absolute indicator of performance across different channels in general. That said we consider our range of SNR to be reasonable, with -5 dB being very low and 15 dB being average. Just as an example for realistic values, and keeping in mind that these are different channels, the Cisco Wireless Site Survey used an SNR scale of 4 dB, for minimal data, to 25 dB for maximum data. The practical experiments in Fig. 9 show higher bit error rates than the ones presented in Fig. 6. The difference in the values is due to the fact that Fig. 6 depicts the theoretical values in the ideal scenario, while Fig. 9 collects the values from actual experiments and more realistic noisy conditions, including noise and inaccuracies introduced by hardware components. In many applications, a BER of 0.0175 is sufficiently low and error correcting codes can be employed to eliminate the small inference. In conclusion, our proposed method of modifying the receiver architecture can effectively prevent the GRS attack while negligibly increasing the bit error rate on the prover side.

5.3. Distance-bounding channel comments

The physical channel used during the distance-bounding phase is an important part of implementing a secure protocol and accurately bounding the time (distance) between the verifier $\mathcal{V}$ and

the prover $\mathcal{P}$. In the previous section, we have already discussed one example of a physical-layer attack strategy (*early-detect/late-send*). We have shown through our experiments that $\mathcal{P}$ must also early detect the challenge to reduce the probability that the GRS attack is successful. In both [44,39] protocol designers are also warned against using multi-bit challenges (or responses). The main attack strategies against multi-bit challenges are highlighted here and we should consider them in the case of HB^+DB , as the proposed hybrid protocol relies on a multi-bit challenge $\mathbf{a}^{(i)}$.

The first strategy involves the possibility that the MiM could influence the prover's system or data clock. The attacker can do this directly, e.g., an RFID gets its system clock from the RF carrier transmitted by the reader (in this case the attacker). She can also influence clocks indirectly, e.g., in some cases $\mathcal{P}$ derives a data clock from the incoming data and so it is possible for the MiM attacker to slightly speed up the signal or move the clock synchronization points earlier in time, which results in a faster data clock being derived.

In both cases, the prover $\mathcal{P}$ will then present its response up to one bit-period earlier (the attacker cannot gain more than one bit-period as she cannot send data she does not already have), which would give the MiM time to hide her activities. Therefore, if our protocol is to be practical $\mathcal{P}$ must have an independent clock source. This could be a simple clock source used for clocking the processor and device peripherals, not a high-frequency clock needed for measuring round-trip time, and thus we feel a realistic requirement for the prover. The second attack strategy is if a dishonest prover $\mathcal{P}^*$ guesses the last challenge bit and starts sending the response back after knowing $\mathbf{a}_{k-1}^{(i)}$. This is effective in protocols using a single multi-bit exchange as the dishonest prover can then with success probability $1/2$ commit distance fraud equal to one bit period. This attack is not effective in our proposed protocol since we have multiple rounds, each containing a multi-bit challenge, and thus the dishonest provers would need to guess the final bit correct for multiple rounds and his probability of success will tend towards that of conventional distance fraud (simply guessing the response early) given in Section 4. The other practical requirements of a good distance-bounding protocol is that the response function should be quick with a constant predictable time, and also that redundant data, such as framing and error correction should be kept to a minimum.

Works on practical channels for DB are limited but there are some good practical examples in the literature of potential channel designs, e.g. [45–47]. In this paper, our main objective was to illustrate that the HB^+DB protocol can be implemented using a channel that is realistic within an environment where this protocol might be used. The channel has to meet the below identified requirements:

- Prover early detects.
- Prover has own clock source, but maintains reliable data reception.
- Predictable, minimal response calculation.
- Earliest possible response (minimum redundant data).

We chose to implement a proof-of-concept channel for ISO 14443 contactless smart cards and we specifically choose to look at ISO 14443 Type B communication. ISO 14443 tokens operate at 13.56 MHz, and usually has a data communication rate of 106 kbps. It is a difficult environment to design a truly secure distance-bounding channel and to achieve good timing characteristics you need proprietary channels [48]. Both Reid et al. [13] and Munilla et al. [49] implemented channels based on ISO 14443 Type A. These two proposals have strong features but our design offers an improvement on both, as discussed below. In [13] and [49], the channels only take into account mafia fraud/MiM attackers and it could be argued that a smart card, a tamper-proof device and

trusted party of a system, is unlikely to commit distance or terrorist fraud. Our goal is therefore to only detect practical relay/MiM attacks, such as [38,41,50].

Since ISO 14443 Type A is generally more widely used we provide a brief comparison of the physical layer properties of Type A and Type B in Fig. 11.

5.3.1. Proof-of-concept implementation

For the implementation, we use a standard Omnikey Multi-ISO high-frequency RFID reader offering ISO 14443 and ISO 15693 functionality. We then use the reader utility to configure the register settings of the reader RF-frontend chipset in such a way that it transmits a 'stripped down' version of ISO 14443 Type B with no CRC (Cyclic Redundancy Check) checksum, or start and end frame indicators. This leaves only the 106 kbps NRZ-encoded (Non-Return to Zero) data modulated on the carrier using 10% ASK modulation (Amplitude Shift Key), together with a stop and start bit for each byte. Using Type B (10% ASK modulation) allows us to send back the response immediately as we can construct a duplex channel where the response is sent while still receiving the final challenge bit of $\mathbf{a}^{(i)}$. If we used Type A, which uses 100% ASK modulation for the challenges, i.e. carrier switches on and off, we would need for the carrier to turn back on before sending the response. We then build a token emulator with similar capabilities to a constrained contactless smart card—8-bit processor (Microchip 16F family, data recovery and response coded in assembly language) with a 13.56 MHz independent, local clock source (inexpensive 50 ppm crystal oscillator). The emulator can demodulate the 10% ASK data modulated onto the carrier and send back data to the reader side using load modulation. The response is only a single bit of Type B 106 kbps NRZ data modulated using BPSK (Binary Phase Shift Key) onto a 847 kHz sub-carrier. On the reader side, we do not actually recover the value using BPSK demodulation, but simply look at the first sub-carrier values within a short $2.48 \mu\text{s}$ window ($2 \times 1/(847 \times 10^3)$) during which the response is expected—a low-high is a one, with a high-low being a zero. A standard reader is not able to receive our modified responses, so we also build an analog demodulation circuit (simple envelope detector) on the reader side to recover the response.

The emulator generates its own sampling sequence, only synchronized to the falling edge of the start bit of the first data byte. After that it samples each bit multiple times and tries to make a decision regarding the value of the bit in the first 40% of the bit period (3.8–4 μs of 9.44 μs). Given that we are using a 13.56 MHz clock the data recovery remains stable (e.g., data 106 kHz $\approx 13.56 \times 10^6/8$) and can reliably exchange 1–12 byte challenges we might use during one distance-bounding round. To optimize the time taken to calculate the final response the emulator calculates the inner product of the current bit of $\mathbf{a}^{(i)}$ and $\mathbf{x}$ after sampling each individual bit, and adds it to the intermediate result. When the final bit arrives the emulator only needs to calculate the inner product of the final bit of $\mathbf{a}^{(i)}$ and $\mathbf{x}$ and then XOR it to c_i . This requires 6 clock cycles, which results in a predictable response calculation of approximately 1.8 μs .

The main benefit of our approach over other proposals in similar systems is that the response can be sent as soon as it is calculated. A trace showing the relation between the final challenge bit and the response is shown in Fig. 10. This is an improvement on [49] as their channel required the token to wait for the carrier to be switched on again after a 'void' (carrier off) signal. We also improve on [13] as their scheme required the token to wait for a predetermined time before responding (potentially vulnerable to overclocking) and also required the reader to directly sample the carrier at 200 MHz. This gives their scheme better timing resolution but our implementation with a simple envelope detector, much slower sampler and early-detect feature can also

Fig. 10. The image shows the time period for the last two bits of the challenge ($a_{k-1}^{(i)} = 0$ and $a_k^{(i)} = 1$) and the stop bit. The yellow plot is the reader's 13.56 MHz carrier, showing the forward channel 10% ASK modulation and also the load-modulated response using BPSK with a 847 kHz sub-carrier. The blue plot shows the envelope of the signal. The red plot shows the response r , as calculated based on early-detected bit value and send before the last bit or stop bit is received. Δt is the time between the time the verifier starts sending the last bit of $a^{(i)}$ and the time r is received, and in this case is approximately 5.5 μ s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

detect most realistic relay/MiM attackers. Our expected response round-trip time is in the region of 5–6 μ s with the verifier's timing resolution about around 600 ns to 1.2 μ s, i.e., can detect divergence from the expected response time (the attack delay) greater than approximately 2 μ s (1 μ s variability in response plus 600 ns to 1.2 μ s variability in timing resolution). This is adequate to detect MiM attacks currently in literature on custom hardware (introduce approximately 20 μ s delay [41], and NFC-enabled mobile phones (tens of milliseconds delay) [50]. The relay attack described in [38] introduces 350–600 ns of delay so our system might not be able detect this attack all the time. That said, this attack closely follows the theoretical model of a relay attacker rather than a MiM attacker, with the attacker not considering sampling or modification of relayed bit values.

6. Conclusions

Distance-bounding and HB-based protocols are two active research topics within the larger areas of RFID and wireless security. We took aspects from both types of protocols to design a hybrid HB⁺DB protocol (combining the HB⁺ protocol and distance bounding ideas) for lightweight device authentication. From a distance-bounding perspective our work shows that using the LPN problem as the basis of the response function could be a promising direction but full analysis of this protocol's resistance against distance-bounding threat scenarios is left for future work. The protocol exhibits some resistance to all three the basic threat scenarios (mafia fraud, distance fraud and terrorist fraud) that DB aims to prevent, but the main purpose of the proposed HB⁺DB protocol is to provide a HB⁺ protocol variation that is resistant to the GRS MiM key-recovery attack, a weakness of basic HB-based protocols. Instead of trying to prevent the attack using cryptographic response functions, we propose to detect the MiM attacker by accurately timing the challenge-response

exchanges, between a prover and a verifier, like one would do in a distance-bounding protocol. This places no additional burden on the constrained prover as only the verifier is tasked with taking the round-trip measurement. The GRS attack against HB⁺DB causes authentication failure because the verifier detects the attack delay incurred in sampling and flipping bits, and is not a result of modified challenges. Thus, we successfully prevent the attacker from learning anything about the prover's key. We further evaluated the success probability of a GRS attack in a practical setting in case it is attempted to circumvent the distance-bound using the advanced *early-detect/late-send* relaying strategy. We show that in realistic conditions, when dealing with a conventional prover the attacker could use this approach to hide her presence. We subsequently show that a simple modification to the prover's receiver could significantly reduce the attacker's probability of success while having minimal impact on the reliability of the communication between the verifier and the prover. Finally, we propose and provide a proof-of-concept implementation of a practical DB channel for detecting MiM attackers based on an ISO 14443 Type B channel that is feasible to implement in contactless smart card/NFC devices. This implementation shows that it is realistic for a low resource device to reliably early-detect a received bit, quickly calculate the response function and to send a response while the final challenge bit is still being received.

In conclusion, this paper successfully serves as an example of how we can improve security through the combined use of conventional cryptography and physical layer techniques. Looking at ways of how to mitigate cryptographic weaknesses, such as the GRS attack against the HB⁺ implementation of authentication using the LPN problem, by integrating physical-layer properties, such as the communication delay or the receiver design, into the protocol could lead to interesting future work on novel cross-layer approaches to secure protocol design.

Acknowledgments

The work described in this paper was substantially supported by a grant from City University of Hong Kong (Project No. 7200375), and supported in part by a grant from the STINT Initiation grant (IB2014-5618) ("Cross-layer authentication for wireless networks"). This work was also partially supported by the SNF Sinergia project "SwissSenseSynergy" (CRSII2 154458), from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under the REA grant agreement no 608743.

Appendix. Useful formulas

If $X_1, \dots, X_n$ are independent Bernoulli random variables with $X_i \in \{0, 1\}$ and $\mathbb{P}(X_i = 1) = \alpha$ for all $i \in \{1, \dots, n\}$, then

$$\mathbb{P}\left(\sum_{i=1}^n X_i \leq u\right) = \sum_{i=0}^u \binom{n}{i} \alpha^i (1-\alpha)^{n-i}. \quad (\text{A.1})$$

Scheme	Reader $\rightarrow$ Token			Token $\rightarrow$ Reader		
	Coding	Modulation	Bit Rate	Coding	Modulation	Bit Rate
Type A	Modified	100% ASK	106 kbps	Manchester	ASK (847kHz sub-carrier)	106 kbps
Type B	NRZ	10% ASK	106 kbps	NRZ	BPSK (847kHz sub-carrier)	106 kbps

Fig. 11. ISO 14443 Type A and B comparison.

This probability can be bounded via *Hoeffding's inequality* [51]:

Lemma 1 (*Hoeffding*). For independent random variables $X_1, \dots, X_n$ such that $X_i \in [a_i, b_i] \subset \mathbb{R}$, with $\mu_i \triangleq \mathbb{E}X_i$ and $t > 0$, it holds:

$$\mathbb{P}\left(\sum_{i=1}^n X_i \leq \sum_{i=1}^n \mu_i + nt\right) = \mathbb{P}\left(\sum_{i=1}^n X_i \leq \sum_{i=1}^n \mu_i - nt\right) \leq \exp\left(-\frac{2n^2 t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right). \quad (\text{A.2})$$

References

- [1] N.J. Hopper, M. Blum, Secure human identification protocols, in: Proc. of ASIACRYPT, 2001, pp. 52–66.
- [2] A. Juels, S.A. Weis, Authenticating pervasive devices with human protocols, in: Proc. of CRYPTO, in: LNCS, vol. 3621, Springer, 2005, pp. 293–308.
- [3] E. Berlekamp, R. McEliece, H. Van Tilborg, On the inherent intractability of certain coding problems (corresp.), IEEE Trans. Inform. Theor. 24 (3) (1978) 384–386.
- [4] H. Gilbert, M. Robshaw, H. Sibert, Active attack against HB^+ : a provably secure lightweight authentication protocol, Electron. Lett. 41 (21) (2005) 1169–1170.
- [5] J. Bringer, H. Chabanne, E. Dottax, HB^{++} : a Lightweight authentication protocol secure against some attacks, in: Proc. of Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, SecPerU, 2006, pp. 28–33.
- [6] D.N. Duc, K. Kim, Securing hb^+ against grs man-in-the-middle attack, in: Symposium on Cryptography and Information Security, SCIS, 2007.
- [7] J. Munilla, A. Peinado, HB-MP: A further step in the hb -family of lightweight authentication protocols, Comput. Netw. 51 (9) (2007) 2262–2267.
- [8] H. Gilbert, M.J.B. Robshaw, Y. Seurin, HB^* : Increasing the security and efficiency of HB^+ , in: Proc. of EUROCRYPT, Springer-Verlag, 2008, pp. 361–378.
- [9] C. Dimitrakakis, A. Mitrokotsa, Distance-bounding protocols: Are you close enough? IEEE Secur. Priv. 13 (4) (2015) 47–51.
- [10] S. Brands, D. Chaum, Distance-bounding protocols, in: EUROCRYPT'93, in: LNCS, Springer, 1993, pp. 344–359.
- [11] L. Bussard, W. Bagga, Distance-Bounding Proof of Knowledge Protocols to Avoid Terrorist Fraud Attacks. Tech. Rep. RR-04-109, EURECOM, 2004, May.
- [12] G.P. Hancke, M.G. Kuhn, An RFID distance bounding protocol, in: Proc. of SECURECOMM, ACM, 2005, pp. 67–73.
- [13] J. Reid, J.M. Gonzalez Nieto, T. Tang, B. Senadji, Detecting relay attacks with timing-based protocols, in: Proc. of ASIACCS, March 2007, pp. 204–213.
- [14] A. Yang, Y. Zhuang, D.S. Wong, An efficient single-slow-phase mutually authenticated RFID distance bounding protocol with tag privacy, in: Proceedings of ICICS, in: LNCS, vol. 7618, Springer, 2012, pp. 285–292.
- [15] Y. Zhuang, A. Yang, D.S. Wong, G. Yang, Q. Xie, A highly efficient RFID distance bounding protocol without real-time PRF evaluation, in: Proceedings of NSS, in: LNCS, vol. 7873, Springer, 2013, pp. 451–464.
- [16] A. Abu-Mahfouz, G.P. Hancke, Distance bounding: A practical security solution for real-time location systems, IEEE Trans. Ind. Inform. 9 (1) (2013) 16–27.
- [17] E. Pagnin, G.P. Hancke, A. Mitrokotsa, Using distance-bounding protocols to securely verify the proximity of two-hop neighbours, IEEE Commun. Lett. 19 (7) (2015) 1173–1176.
- [18] I. Boureanu, A. Mitrokotsa, S. Vaudenay, Practical and provably secure distance-bounding, J. Comput. Secur. 23 (2) (2015) 229–257.
- [19] G. Avoine, M. Bingöl, S. Kardaş, C. Lauradoux, B. Martin, A framework for analyzing RFID distance bounding protocols, J. Comput. Secur. (2011).
- [20] A. Mitrokotsa, C. Onete, S. Vaudenay, Location leakage in distance bounding: Why location privacy does not work, Comput. Secur. 45 (2014) 199–209.
- [21] C. Dimitrakakis, A. Mitrokotsa, S. Vaudenay, Expected loss analysis for authentication in constrained channels, J. Comput. Secur. 23 (3) (2015) 309–329.
- [22] A. Mitrokotsa, P. Peris-Lopez, C. Dimitrakakis, S. Vaudenay, On selecting the nonce length in distance-bounding protocols, Comput. J. (2013).
- [23] C.H. Kim, G. Avoine, F. Koeune, F.-X. Standaert, O. Pereira, The swiss-knife RFID distance bounding protocol, in: Proc. of ICISC, in: LNCS, vol. 5461, 2008, pp. 98–115. December.
- [24] A. Bay, I. Boureanu, A. Mitrokotsa, I. Spulber, S. Vaudenay, The Bussard-Bagga and other distance bounding protocols under man-in-the-middle attacks, in: Proc. of Inscrypt, in: LNCS, Springer, 2012.
- [25] G.P. Hancke, Distance-bounding for RFID: Effectiveness of terrorist fraud in the presence of bit errors, in: Proc. of IEEE Conference on RFID Technology and Applications, RFID-TA, 2012.
- [26] A. Mitrokotsa, C. Dimitrakakis, P. Peris-Lopez, J.C.H. Castro, Reid et al.'s distance bounding protocol and mafia fraud attacks over noisy channels, IEEE Commun. Lett. 14 (2) (2010) 121–123.
- [27] A. Mitrokotsa, C. Onete, S. Vaudenay, Mafia fraud attack against the rc distance-bounding protocol, in: Proceedings of the 2012 IEEE RFID Technology and Applications, IEEE RFID T-A, IEEE Press, Nice, France, 2012, pp. 74–79. November.
- [28] I. Boureanu, A. Mitrokotsa, S. Vaudenay, On the pseudorandom function assumption in (secure) distance-bounding protocols - prf-ness alone does not stop the frauds!, in: LATICRYPT, in: Lecture Notes in Computer Science, vol. 7533, Springer, 2012, pp. 100–120.
- [29] F. Armknecht, M. Hamann, V. Mikhalev, Lightweight authentication protocols on ultra-constrained RFIDs - myths and facts, in: Proc. of RFIDSec, 2014.
- [30] E. Pagnin, A. Yang, G. Hancke, A. Mitrokotsa, HB^+DB , mitigating man-in-the-middle attacks against hb^+ with distance bounding, in: Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WiSec'15, ACM, New York, NY, USA, 2015, pp. 3:1–3:6.
- [31] J. Katz, J. Shin, Parallel and concurrent security of the HB and HB^+ protocols, in: Advances in Cryptology—EUROCRYPT 2006, in: Lecture Notes in Computer Science, vol. 4004, Springer, Berlin Heidelberg, 2006, pp. 73–87.
- [32] J.S.J. Katz, A. Smith, Parallel and concurrent security of the HB and HB^+ protocols, J. Cryptology 23 (3) (2010) 402–421.
- [33] A. Blum, A. Kalai, H. Wasserman, Noise-tolerant learning, the parity problem, and the statistical query model, J. ACM 50 (4) (2003) 506–519.
- [34] Y. Desmedt, Major security problems with the 'Unforgeable' (Feige)-Fiat-Shamir proofs of identity and how to overcome them, in: Proc. of SecuriCom, SEDEP Paris, France, 1988, pp. 15–17.
- [35] C. Cremers, K.B. Rasmussen, S. Çapkun, Distance hijacking attacks on distance bounding protocols, in: Proc. of IEEE Symposium on Security and Privacy, 2012, pp. 113–127.
- [36] G. Avoine, A. Tchamkerten, An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement, in: Information Security, in: LNCS, vol. 5735, Springer, 2009, pp. 250–261.
- [37] I. Boureanu, A. Mitrokotsa, S. Vaudenay, Practical and provably secure distance-bounding, in: Proc. of the 16th Information Security Conference, ISC 2013, in: LNCS, Springer, 2013.
- [38] A. Francillon, B. Danev, S. Çapkun, Relay attacks on passive keyless entry and start systems in modern cars, in: Proc. of NDSS, 2011.
- [39] J. Clulow, G.P. Hancke, M.G. Kuhn, T. Moore, So near and yet so far: Distance-bounding attacks in wireless networks, in: Proc. of European Workshop on Security & Privacy in Ad-Hoc & Sensor Networks, in: LNCS, vol. 4357, Springer, 2006, pp. 83–97.
- [40] P.-H. Thevenon, O. Savry, S. Tedjini, On the weakness of contactless systems under relay attacks, in: Proceedings of the 19th International Conference on Software, Telecommunications and Computer Networks, SoftCOM'11, IEEE, 2011, pp. 1–5.
- [41] G.P. Hancke, K.E. Mayes, K. Markantonakis, Confidence in smart token proximity: Relay attacks revisited, Comput. Secur. 28 (7) (2009) 404–408.
- [42] USRP B200 description. <http://www.ettus.com/product/details/UB200-KIT>.
- [43] GNU radio description. <http://gnuradio.org/redmine/projects/gnuradio/wiki>.
- [44] G. Hancke, M. Kuhn, Attacks on time-of-flight distance bounding channels, in: Proc. of WISec 2008, ACM, 2008, pp. 194–202.
- [45] S. Drimer, S. Murdoch, Keep your enemies close: Distance bounding against smartcard relay attacks, in: Proceedings of USENIX Security Symposium, 2007.
- [46] K.B. Rasmussen, S. Çapkun, Realization of RF distance bounding, in: Proc. of USENIX Security Symposium, 2010.
- [47] A. Ranganathan, B. Danev, S. Çapkun, Low-power distance bounding, 2014 <http://arxiv.org/abs/1404.4435>.
- [48] G.P. Hancke, Design of a secure distance-bounding channel for RFID, J. Netw. Comput. Appl. (2010).
- [49] J. Munilla, A. Peinado, Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels, Wireless Communications and Mobile Computing 8 (9) (2008) 1227–1232.
- [50] L. Francis, G. Hancke, K. Mayes, K. Markantonakis, Practical relay attack on contactless transactions by using NFC mobile phones, in: Proceedings of Workshop on RFID and IoT Security, RFIDsec Asia 2012, IOS Press, 2012, pp. 21–32.
- [51] W. Hoeffding, Probability inequalities for sums of bounded random variables, J. Amer. Statist. Assoc. 58 (301) (1963) 13–30.

Elena Pagnin received her Bachelor Degree in Theoretical Mathematics at the University of Padova in 2011, and Master Degree in Applied Mathematics at the University of Trento in 2013, respectively. She is currently a Ph.D. Student in Network Security at the Department of Computer Science at Chalmers University of Technology, Sweden. Her research interests include lightweight authentication protocols, biometric authentication, and applied cryptography.

Anjia Yang received the B.S. degree from Jilin University in 2011. He is currently working towards the Ph.D. degree in the Department of Computer Science, City University of Hong Kong. His research interests include RFID security and privacy, applied cryptography, and cloud computing.

Qiao Hu received the B.S. degree from Hunan University in 2011 and M.Sc. degree from Wuhan University in 2013. He is currently working towards the Ph.D. degree in the Department of Computer Science, City University of Hong Kong. His research interests include RFID security and privacy, cloud computing and wireless communication security.

Gerhard P. Hancke received the B.Eng. and M.Eng. degrees from the University of Pretoria, Pretoria, South Africa, in 2002 and 2003, respectively, and the Ph.D. degree from the Computer Laboratory, University of Cambridge, Cambridge, U.K., in 2008. He is an Assistant Professor with the Department of Computer Science, City University of Hong Kong. His main research interests revolve around advanced sensing and security, especially for RFID/contactless, mobile and other pervasive technologies.

Aikaterini Mitrokotsa is an Associate Professor at the Department of Computer Science and Engineering at Chalmers University of Technology. Her main research interests lie in information security, privacy-preservation, authentication protocols and provable security. She has been awarded the young researcher grant from the Swedish Research Council, the Rubicon Research Grant by NWO and a Marie Curie Intra European Fellowship. She has served on the PCs of INFOCOM, ACNS, Indocrypt, and is currently serving as associate editor for the Computers & Security Journal and the Ad hoc Networks Journal (Elsevier).