Short: HB+DB, Mitigating Man-in-the-Middle Attacks Against HB+ with Distance Bounding

Elena Pagnin[1], Anjia Yang[2], Gerhard Hancke[2], and Aikaterini Mitrokotsa[1]

[1]Chalmers University of Technology, Gothenburg, Sweden, {elenap, aikmitr}@chalmers.se

[2]City University of Hong Kong, China, {ayang3-c@my.cityu.edu.hk, ghancke@ieee.org}

ABSTRACT

Authentication for resource-constrained devices is seen as one of the major challenges in current wireless communication networks. The HB^+ protocol performs device authentication based on the *learning parity with noise* (LPN) problem and simple computational steps, that renders it suitable for resource-constrained devices such as *radio frequency identification* (RFID) tags. However, it has been shown that the HB^+ protocol as well as many of its variants are vulnerable to a simple man-in-the-middle attack. We demonstrate that this attack could be mitigated using physical layer measures from distance-bounding and simple modifications to devices' radio receivers. Our hybrid solution ($\mathsf{HB}^+\mathsf{DB}$) is shown to provide both effective distance-bounding using a lightweight HB^+-based response function, and resistance against the man-in-the-middle attack to HB^+. We provide experimental evaluation of our results as well as a brief discussion on practical requirements for secure implementation.

Keywords: Distance bounding, HB-protocol, HB^+, physical layer security.

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*. A few years later, Juels and Weis [2] proposed an enhanced variant of the HB protocol, called HB^+ [2], that is suitable for RFID systems whose resource constraints render them vulnerable to multiple attacks [3]. Gilbert *et al.* [4] showed the vulnerability of HB^+ against an active man-in-the-middle attack called the GRS attack. 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.

WiSec '15 June 22-26, 2015, New York, NY, USA

ACM 978-1-4503-3623-9/15/06 ...$15.00
DOI: http://dx.doi.org/10.1145/2766498.2766516.

Multiple variations of 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]. Two solutions resistant against the GRS attack are proposed in [8]. The first proposal, RANDOM-$\mathsf{HB}^\#$, has storage costs that are insurmountable for a constrained device, the second proposal $\mathsf{HB}^\#$, requires less storage while being secure against the GRS attack. The above cited proposals have solely focused on cryptographic solutions. In contrast, our work investigates how to prevent the GRS attack by adopting non-cryptographic measures such as modifying the communication channel of the receiver architecture.

Distance-bounding (DB) protocols are challenge response protocols that can be employed to prevent MiM attacks and especially relay attacks. In DB protocols the upper bound on the distance between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ 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 $c \cdot \frac{t_{\mathsf{max}}}{2}$, where t_{max} is the maximum delay time between sending a challenge and receiving a response. In the years, many DB protocols have been proposed, e.g. [9–11], along with frameworks for further analysis, e.g. [12, 13], and protocol vulnerabilities, e.g. [14–18]. Multiple issues have been raised about the feasibility of implementing HB-based protocols for ultra-constrained devices [19]. In this work, we design a novel distance-bounding/authentication protocol based on HB^+ that is feasible in constrained environments (RFID tags).

Contributions: Despite the structure similarities between HB^+-based protocols and DB protocols (multiple exchanges during protocol execution and single-bit responses), the potential interface between these two types of protocols has not been explored yet. This paper combines principles of DB (detection of physical-layer communication delay) and HB^+-based protocols (LPN-based response function) to construct a novel hybrid $\mathsf{HB}^+\mathsf{DB}$ protocol. This protocol provides resistance against the MiM attack affecting HB^+-based protocols, while also serving as a lightweight DB authentication protocol, and therefore contributes to the research areas of both distance-bounding and HB^+-based protocols. We provide a theoretical security analysis for our proposed protocol along with simulation and practical experiments. Furthermore, we explain that the DB phase of $\mathsf{HB}^+\mathsf{DB}$ could be securely implemented if only few practical requirements are adhered to.

2. THE HB$^+$DB PROTOCOL

We propose a new distance-bounding protocol that relies on the HB$^+$ scheme, and we name it HB$^+$DB. The HB$^+$DB is run between a (trusted) verifier $\mathcal{V}$ and a (possibly untrusted) prover $\mathcal{P}$. $\mathcal{V}$ and $\mathcal{P}$ share three secret keys $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \{0,1\}^k$ together with a (secret) real number $\eta \in (0, \frac{1}{2})$, which gives the error probability introduced by the prover $\mathcal{P}$ in the challenge-response phase. The value $t_{\mathsf{max}} \in \mathbb{R}_{>0}$ is fixed, and will be used to bound the maximum time required to transmit a message from $\mathcal{V}$ to $\mathcal{P}$ (and so the distance between the two parties).

As depicted in Figure 1, the HB$^+$DB is discriminated into the three main phases that characterise most DB protocols:

Initialisation Phase: The prover generates a uniformly random nonce $\mathbf{s} \xleftarrow{R} \{0,1\}^k$ and sends it to the verifier. Both $\mathcal{V}$ and $\mathcal{P}$ use the key $\mathbf{z}$ as a seed to the PRF (pseudo random function) $f : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}^k$, and then evaluate it on the shared nonce $\mathbf{s}$. From $f_{\mathbf{z}}(\mathbf{s})$, the two parties obtain n vectors $\mathbf{b}^{(i)}$, e.g. setting $\mathbf{b}^{(i)} := f_{\mathbf{z}}^{(i)}(\mathbf{s}) := \left(\circ_{j=1}^{i} f_{\mathbf{z}}\right)(\mathbf{s})$, or $\mathbf{b}^{(i)} := f_{\mathbf{z}}^{(i)}(\mathbf{s}) := f_{\mathbf{z}}(\mathbf{s}+i)$, $\forall i \in \{1,\ldots,n\}$.

In order to lighten the computations performed in the time-critical phase, $\mathcal{P}$ computes during this initialisation phase n vectors $c_i = (\mathbf{b}^{(i)} \cdot \mathbf{y}) \oplus \epsilon_i \in \{0,1\}$, where $\epsilon_i \in \{0,1\}$ satisfies $\mathbb{P}[\epsilon = 1] = \eta$. Similarly, $\mathcal{V}$ generates uniformly at random n k-bit vectors $\mathbf{a}^{(i)} \xleftarrow{R} \{0,1\}^k$.

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

Verification Phase: The verifier accepts the prover if both the following conditions hold.

1. The received responses are *'correct'*, i.e. let $\tilde{r}_i := (\mathbf{a}^{(i)} \cdot \mathbf{x}) \oplus (\mathbf{b}^{(i)} \cdot \mathbf{y})$ (computed by $\mathcal{V}$), where ϵ_i denotes the error $\tilde{r}_i \oplus r_i = \epsilon_i$. $\mathcal{V}$ will accept $\mathcal{P}$ if $\tilde{r}_i = r_i$ for at least $n(1-\eta)$ equations (indeed it must hold $\mathsf{HW}(\epsilon) \sim \mathsf{Binomial}(n,\eta)$).
2. The prover $\mathcal{P}$ is *'close enough'*, i.e. $\Delta t_i \leq 2t_{\mathsf{max}}$.

We stress that the value of t_{max} should take into account the time needed by $\mathcal{P}$ to perform the computation $(\mathbf{a}^{(i)} \cdot x) \oplus c_i$, which depends on the length of the random nonces $\mathbf{a}^{(i)}$.

Errors and noise.

In order to define how tolerant $\mathcal{V}$ should be in the verification phase, we model the errors introduced by $\mathcal{P}$ as i.i.d. (independent, identically distributed) Bernoulli random variables $\epsilon_i \sim \mathsf{Be}(\eta)$, $i = 1,\ldots,n$. The total number of errors (coming from the LPN-security) in a full run of the protocol is thus $S = \left(\sum_{i=1}^{n} \epsilon_i\right) \sim \mathsf{Binomial}(n,\eta)$. 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 $\mathbb{P}_{\mathsf{FR}} = P[|S - \mu| \geq \tau]$. By the Hoeffding inequality: $\mathbb{P}_{\mathsf{FR}} \leq 2e^{-\frac{2\tau^2}{n}}$, which leads to (solving in τ): $\tau \leq \sqrt{\frac{n\,log(2/\mathbb{P}_{\mathsf{FR}})}{2}}$. The choice of τ also influences the probability of false acceptance $\mathbb{P}_{\mathsf{FA}}$, which is directly connected to the probability of success in each of the frauds against DB protocols and will be discussed in detail in Section 3.

Verifier $\mathcal{V}$		**Prover $\mathcal{P}$**
$(\mathbf{x},\mathbf{y},\mathbf{z},\eta)$		$(\mathbf{x},\mathbf{y},\mathbf{z},\eta)$
Initialisation Phase		
	$\xleftarrow[\text{input nonce}]{\mathbf{s}}$	$\mathbf{s} \xleftarrow{R} \{0,1\}^k$
$for\ i = 1,\ldots,n$		$for\ i = 1,\ldots,n$
$\mathbf{b}^{(i)} = f_{\mathbf{z}}^{(i)}(\mathbf{s}) \in \{0,1\}^k$		$\mathbf{b}^{(i)} = f_{\mathbf{z}}^{(i)}(\mathbf{s}) \in \{0,1\}^k$
		$c_i = (\mathbf{b}^{(i)} \cdot \mathbf{y}) \oplus \epsilon_i$
$\mathbf{a}^{(i)} \xleftarrow{R} \{0,1\}^k$		$\epsilon \sim \mathsf{Be}(\eta)$
Distance-Bounding Phase (for $i = 1,\ldots,n$)		
Start Clock	$\xrightarrow[\text{challenge}]{\mathbf{a}^{(i)}}$	$r_i = (\mathbf{a}^{(i)} \cdot \mathbf{x}) \oplus c_i$
Stop Clock	$\xleftarrow[\text{response}]{r_i}$	
Verification phase		
$\mathcal{V}$ accepts if $\sum_{i=1}^{n}\left((\mathbf{a}^{(i)} \cdot \mathbf{x}) \oplus (\mathbf{b}^{(i)} \cdot \mathbf{y}) \oplus r_i\right) \in [\mu - \tau, \mu + \tau]$, and $\Delta t_i \leq 2t_{\mathsf{max}}$ for all i.		

Figure 1: The HB$^+$DB protocol. The value $\mu = n\alpha$ represents the (expected) mean of the sum of the Bernoulli random variables $X_i = (\mathbf{a}^{(i)} \cdot \mathbf{x}) \oplus (\mathbf{b}^{(i)} \cdot \mathbf{y}) \oplus r_i$. In case the communication channel is not affected by noise $\alpha = \eta$.

Threat Model.

The threat model for HB$^+$DB resembles the existing one for DB protocols [20]– distance fraud, mafia fraud, terrorist fraud – with the addition of the main threat against HB$^+$, the GRS attack [8]. We define the main threats as follows:

Distance Fraud: In this attack a legitimate but dishonest prover $\mathcal{P}^*$ located far away from a verifier $\mathcal{V}$ tries to prove to the latter that she is in $\mathcal{V}$'s proximity.

Mafia Fraud: This attack involves an adversary $\mathcal{A}$ located between the verifier and a legitimate, far away (outside $\mathcal{V}$'s proximity), prover $\mathcal{P}$. The adversary's aim is to make the distance between $\mathcal{P}$ and $\mathcal{V}$ appear shorter (to $\mathcal{V}$) than it is in reality. Note that $\mathcal{P}$ is unaware of the attack.

Terrorist Fraud: The setting is the same as in Mafia Fraud, with the only difference that the dishonest prover $\mathcal{P}^*$ is actually aware of the attack and colludes with $\mathcal{A}$. The aim of $\mathcal{A}$ and $\mathcal{P}^*$ is to collaborate in order to make $\mathcal{V}$ believe that $\mathcal{A}$ is $\mathcal{P}$ (equivalently, $\mathcal{P}$ lies in $\mathcal{V}$'s proximity). The main restriction on the *help* $\mathcal{P}^*$ can give to $\mathcal{A}$ is that the leaked information should not provide to $\mathcal{A}$ any advantage in being authenticated in a subsequent run of the protocol on her own without the provers's help.

Variations and generalisations of these threats have been presented in the literature, e.g. [12, 21–23] .

A description of the GRS attack to HB$^+$ goes as follows:

GRS attack: The setting of this attack is similar to that in mafia fraud, we note that $\mathcal{A}$ has the power to manipulate the challenges sent by $\mathcal{V}$ to $\mathcal{P}$. As depicted in Figure 2, $\mathcal{A}$ chooses a fixed k-bit vector $\mathbf{d}$ and uses it at each round to alter the challenges $\mathbf{a}$ into $\mathbf{a}' = \mathbf{a} \oplus \mathbf{d}$. Note that the same $\mathbf{d}$ is used in all rounds of the HB$^+$ protocol. Acceptance or failure of the authentication reveals one bit of the secret key $\mathbf{x}$ (in the first case $\mathbf{x} \oplus \mathbf{d} = 0$, in the latter $\mathbf{x} \oplus \mathbf{d} = 1$). In order to recover the whole vector $\mathbf{x}$, $\mathcal{A}$ needs to run the entire protocol k times for linearly independent vectors[1] $\mathbf{d}$, and solve the resulting linear system. As soon as $\mathbf{x}$ is recovered $\mathcal{A}$ may easily impersonate $\mathcal{P}$ by setting $\mathbf{b} = 0$ or

[1]E.g. the vectors of the canonical basis $\mathbf{d}^{(i)} = (0,..,0,1,0..,0)$ with the only non-zero component in position $i = 1,..,k$.

by employing a similar strategy as the one described above to recover the other secret key $\mathbf{y}$.

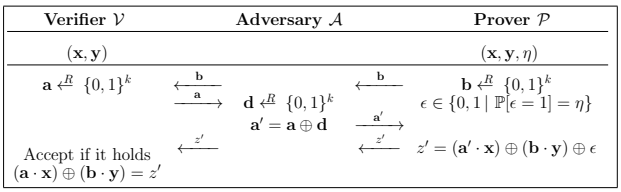

Figure 2: The GRS attack against one round of the HB^+.

In Section 4 we demonstrate that while theoretically the GRS attack is quite simple to mount, from a practical perspective it is real challenging to perform. Moreover, time constraints added in the authentication process (DB protocol) constitute an insurmountable obstacle for the GRS against our proposed protocol (HB^+DB).

3. SECURITY ANALYSIS

In this section we investigate the resistance of the proposed HB^+DB protocol against the main threats: 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 4.

In the sequel, index $i = 1, \ldots, n$. We assume[2] that the vectors $\mathbf{a}^{(i)}, \mathbf{b}^{(i)} \sim \mathsf{Be}(\frac{1}{2})$ and that the *errors* introduced by the prover in the responses are $\epsilon_i \sim \mathsf{Be}(\eta)$, $\eta < \frac{1}{2}$. On the same line, we model the *channel-noise* as $\delta_i \sim \mathsf{Be}(\nu)$ for $\nu < \frac{1}{2}$. 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. According to this notation, the final response-bit that reaches the verifier $\mathcal{V}$ at each round of the DB 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 $r_i \oplus (\mathbf{a}^{(i)} \cdot \mathbf{x}) \oplus (\mathbf{b}^{(i)} \cdot \mathbf{y}) = \epsilon_i \oplus \delta_i =: \omega_i$. The ω_i shall be i.i.d. Bernoulli random variables of parameter $\alpha = \eta + \nu - 2\eta\nu$. Thus, we will directly consider $\omega_i \sim \mathsf{Be}(\alpha)$ as the total noise, and use $\mu = n\alpha$ as its mean.

Distance fraud: The challenges $\mathbf{a}^{(i)}$ appear completely random to the malicious prover $\mathcal{P}^*$, hence the *best* strategy for the malicious prover is to early-send a random bit $r_i^* \sim \mathsf{Be}(1/2)$ as answer. The probability to succeed in distance fraud therefore equals the probability of false acceptance $\mathbb{P}_{\mathsf{FA}}$ where the error $\zeta_i = r_i \oplus r_i^* \sim \mathsf{Be}(1/2)$. Let $\Delta = (1/2 - \alpha - \tau/n)$ and $S_n = \sum_{i=1}^n \zeta_i$, then Hoeffding inequality yields to: $\mathbb{P}_{DF} = \mathbb{P}(\mu - \tau \leq S_n \leq \mu + \tau) \leq e^{-2n\Delta^2}$.

Mafia Fraud: To perform a mafia fraud attack, $\mathcal{A}$ simply relays $\mathbf{s}$ during the initialisation 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*. In the former case $\mathcal{A}$ has to guess the challenge before having received it from $\mathcal{V}$. With probability 2^{-k} at each round, $\mathcal{A}$ guesses the correct challenge[3], and so she can forward $\mathcal{P}$'s response r_i. In the *guess the responses* strategy, the attacker guesses the *correct* value of r_i with probability 2^{-1} at each round. However, it is possible for $\mathcal{A}$ to run both strategies consecutively. More precisely, the attacker will *guess the responses* only if she failed to anticipate the *correct* challenge. Finally, the probability of a successful mafia fraud attack against HB^+DB is $\gamma = 2^{-k} + (1 - 2^{-k})2^{-1} = 2^{-1} + 2^{-(k+1)}$. We can bound the probability of mafia fraud by: $\mathbb{P}_{MF} = \mathbb{P}(\mu - \tau \leq S_n \leq \mu + \tau) \leq e^{-2n\Delta'^2}$, where $\Delta' = (1 - \gamma - \alpha - \tau/n)$.

Terrorist Fraud: In this attack, $\mathcal{P}^*$ helps the adversary $\mathcal{A}$ and discloses to her the values c_i. This leakage of information is not harmful, since $\mathcal{A}$ cannot re-use the c_i-s in the future (as they depend on the nonce $\mathbf{s}$ and on the errors ϵ_i, different at each new run of the protocol), and even if she did, she still needed to guess correctly the value $\mathbf{a}^{(i)} \cdot \mathbf{x}$, that happens with probability 2^{-1} per round (same with mafia fraud attack). In order to certainly 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, it is easy to see that if $\mathcal{A}$ knows $\mathbf{x}$ she can mount (in the future) successful mafia frauds (using the *pre-ask* strategy). Thus, according to Avoine *et al.*'s definition [13], the probability of success in a terrorist-fraud attack against the HB^+DB protocol is the same as in mafia fraud: $\mathbb{P}_{TF} = \mathbb{P}_{MF}$.

GRS Attack: In this attack, the j-th bit of the key $\mathbf{x}$ is retrieved by first *learning* the value of the j-th bit of each challenge (i.e. $\mathbf{a}_j^{(i)}$) and sending it *flipped* to $\mathcal{P}$. 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, from a practical point of view, a mafia fraud is 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 $\mathcal{P}$ [24]. In the GRS attack, $\mathcal{A}$ 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 (theoretically short) time to perform the bit *sampling*, 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*. As we demonstrate in Section 4, the HB^+DB is resilient against the GRS attack due to the failure of the distance-bound (i.e. $\Delta t_i > t_{\mathsf{max}}$ for the i-th round).

4. PRACTICAL CONSIDERATIONS

The purpose of our experimental evaluations is to demonstrate that the active MiM key recovery GRS attack proposed by Gilbert *et al.* [4] is unsuccessful against our HB^+DB protocol. We preset a series of simulation experiments, run in Matlab that show the resistance of the HB^+DB protocol against the GRS attack in a practical environment. We take into account prominent attack strategies proposed in the literature for circumventing DB protocols, which take advantage of latency at the physical layer of the communication channel.

We remind the reader that in order to perform a GRS attack, 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 the final result of the protocol. Any additional delay introduced by the attacker would be detected by $\mathcal{V}$. In order to introduce no

[2]This hypothesis can be relaxed by requiring only the distribution of the vectors $\mathbf{a}^{(i)}$ to be unknown (or uniform) to the attacker and to the malicious prover.

[3]Which she can verify when she receives (later on) the correct $\mathbf{a}^{(i)}$ sent by $\mathcal{V}$

additional delay (mostly due to learning the bit-value) $\mathcal{A}$ will follow two strategies proposed in [25]: *early-detect* and *late-send*. The strategies exploit common low-resource receiver architecture: $\mathcal{P}$ is assumed to sample multiple times (or integrate) across the entire bit-period[4] π. In this setting, $\mathcal{A}$ can *early-detect*, i.e. decide on the value of the bit (sent by $\mathcal{V}$) in a time *shorter* than π, let us call this shorter-sampling-time *early*. In the same π, $\mathcal{A}$ will also *late-send* her modified bit value, let us refer to this shorter-transmitting-time as *late*. From the point of view of the prover, he at first samples the original bit value (for time $early < \pi$), and then the modified bit value (for the remaining $late \sim \pi - early$). The attack succeeds if $\mathcal{P}$ decodes the modified (flipped) value instead of the one sent by $\mathcal{V}$.

GRS *attack against* HB$^+$DB - *simulations.*

In the experiments we simulate the GRS attack against the HB$^+$DB protocol under the following assumptions:

1. The verifier transmits single-bit challenge(s) $C \in \{0,1\}$ (this can be done without loss of generality[5]).
2. The noise of the environment is Gaussian that adds to the transmitted challenge C, and the noise is the same for all parties ($\mathcal{V}$, $\mathcal{P}$ and $\mathcal{A}$).
3. The adversary integrates over the *early* part of π to guess the challenge and then modifies the guessed challenge to $C' \in \{0,1\}$, and she adds the noise.
4. The modification performed by $\mathcal{A}$ is to *flip the bit* for the *late* part of π. Although it is not trivial to do this in practice, we assume that an attacker $\mathcal{A}$ can *instantly* modify a transmitted 1 symbol to 0 and vice versa.
5. Upon receiving $C'+noise$, $\mathcal{P}$ integrates over the whole π.

The GRS attack fails whenever $\mathcal{A}$ does not guess the value of C correctly (no modification of the challenge happens), or $\mathcal{P}$ still decodes the challenge as C rather than C'. The only case in which the GRS attack succeeds is when $\mathcal{A}$ guesses the challenge correctly (C), and $\mathcal{P}$ receives the modified challenge (C') correctly.

The flow of the attack is illustrated in Figure 3. We consider $n = 10$ rounds of the protocol and the $\mathcal{A}$ will try to flip the challenge-bit at each round. Figure 3 shows the practical implications of the GRS attack: the early-detect strategy on $early = \pi/10$ makes $\mathcal{A}$ guess the wrong value on bits 2,4,6,7 and 9. On those same bits, regardless of the noise, $\mathcal{P}$ would have correctly decoded the right value. In other words, Figure 3 demonstrates that the GRS attack against HB$^+$DB fails on half of the challenges, thus it is unfeasible.

The second experiment tests the attacker's success probability under various communication conditions. Figure 4 depicts the relationship between the signal to noise ratio (SNR), the portion *early* on which $\mathcal{A}$ samples before making her guess on the bit-value and the success probability of the attack $P = \frac{number\ of\ bits\ modified\ successfully}{the\ total\ number\ of\ bits}$. The values of P in Figure 4 correspond to the success probability for each round of the DB phase, thus the actual success probability should be risen in the power of n (i.e. number of rounds in the HB$^+$DB protocol). It is interesting to notice

[4]The value of bit-period corresponds to the time needed to transmit a bit.

[5]It is easy to see that the same procedure applies when the challenge is composed of k-bits, as all the attacker needs is to flip one chosen bit.

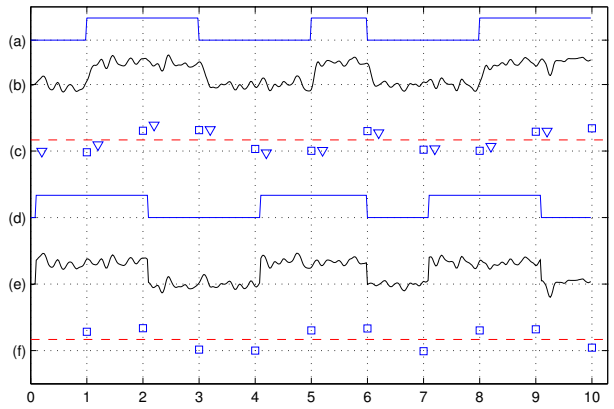

Figure 3: Simulation of the GRS attack. SNR= 5 dB (signal to noise ratio), $early = \pi/10$ (10% of the bit-period). Plot (a): original challenge-bit C sent by $\mathcal{V}$; plot (b): challenge influenced by the noise, $C + noise$; plot (c): the triangle represents $\mathcal{A}$'s guessed challenge via early detection, the square indicates the bit-value decoded by $\mathcal{P}$ without $\mathcal{A}$ modifying it (the red line indicates the decision threshold: a triangle/square above, resp. below, the line indicates a 1, resp. a 0). Plot (d): $\mathcal{A}$'s modified challenge C'; plot (e): $C' + noise$, the modified noisy challenge; plot (f): tampered challenge received by $\mathcal{P}$. Note that due to the failure to accurately detect the bit, $\mathcal{A}$ fails to flip 5 of the bits.

that for any value of SNR, P increases to a summit first, and then decreases. The increment is due to the fact that as *early* increases, $\mathcal{A}$ has more time to guess the challenge value, which translates in a higher chance to decode the correct bit. However, the larger *early* becomes, the shorter will be the time ($late = 1 - early$) for $\mathcal{A}$ to flip the bit and transmit the tampered value to $\mathcal{P}$. In practice, $P < 0.5$ happens if $early > \pi/2$, since $\mathcal{A}$ would then have received half of the original signal already and sending the modified signal for less than half bit-period π would not be sufficient to make $\mathcal{P}$ decode the tampered value.

In Figure 4, we see that P is very large (up to 1) when $6 < \text{SNR} < 10$ and $0.2 < early < 0.4$. This fact implies that, if the prover is given a conventional receiver architecture, the GRS attack would be realistic in a low-resource communication environment. We solve this inconvenience by requiring $\mathcal{P}$ to adopt the *early-detect* strategy too. In other words, now the prover integrates over a 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 $px/2$ instead than $\pi/2$. On the other hand, shortening the sampling time makes the bit error rate (BER) of the prover, $\mathcal{P}_{\text{BER}}$, increase. Thus, we ran a third experiment, to determine the value $px_0 \in (0.1, 1)$ for which the corresponding $\mathcal{P}_{\text{BER}}$ is low *enough*, i.e. $\mathcal{P}_{\text{BER}} < 0.1$, reducing effectively the success probability of the attack. Figure 5 depicts the success probability of the GRS attack when $px = 40\%$ of the bit-period. We investigated how to reduce P while maintaining an acceptable value of $\mathcal{P}_{\text{BER}}$. Our experiment showed that when $\text{SNR} > 5$ dB the $\mathcal{P}_{\text{BER}} = 0$ while $P < 0.55$. These results imply that, when the HB$^+$DB is implemented with $n = 16$ rounds, the practical success probability of the GRS attack is $\approx 2^{-13}$, which is approximately the same as guessing a 4-digit PIN. In case the SNR = 10 dB and $early = 18\%$ of π, the highest success probability of the attacker at each

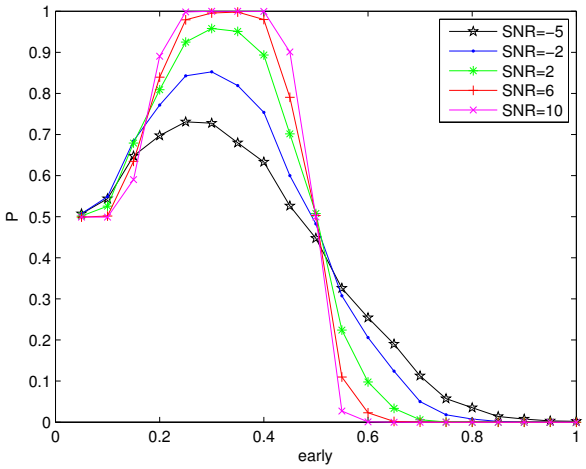

Figure 4: The attacker's success probability under different conditions. The prover is supposed to sample across 100% of the bit-period π, while $\mathcal{A}$ is asked to flip 1000 randomly-chosen bits. SNR varies in the range from -5 dB to 10 dB with step 1, and *early* is in the range from 0.05 to 1 with step 0.05. The plotted values are obtained as average of 10 independent run of the same experiments.

round is $P = 0.6104$, while $\mathcal{P}_{\mathsf{BER}} = 0$. When $n = 32$, the success probability of the GRS attack is definitely negligible (approximately 2^{-23}). In practice, SNR is usually larger than 5 dB. Therefore, our simulations demonstrate that we successfully prevent the GRS in our HB$^+$DB protocol by letting $\mathcal{P}$ integrate on a smaller part of the bit-period.

4.1 Further Comments on the Channel

The physical channel used during the DB phase is an important part of implementing a secure protocol and accurately bounding the time (distance) between the verifier $\mathcal{V}$ and the prover $\mathcal{P}$. Work on practical channels for distance-bounding are limited but there are some good practical examples in the literature of potential channel designs, e.g. [26, 27]. 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 this attack is successful. To conclude, we would like to briefly discuss some further practical channel issues pertinent to our protocol proposal and summarise the requirements needed to implement the protocol securely.

In both [25, 28] protocol designers are 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 synchronisation points earlier in time, which results in a faster data clock being derived. In both cases, $\mathcal{P}$ 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.

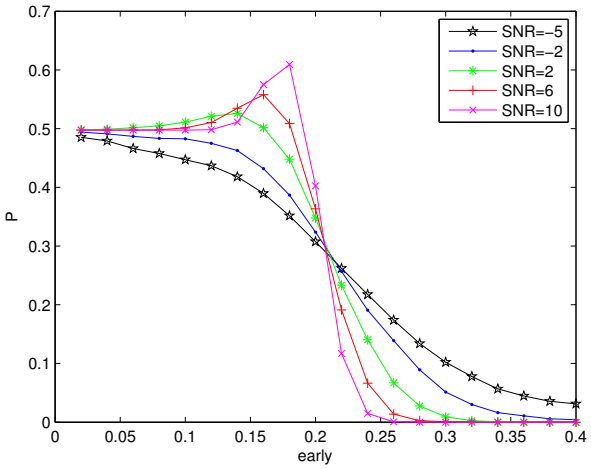

Figure 5: The success probability of GRS against HB$^+$DB when $\mathcal{P}$ integrates over $px_0 = 40\% of the \mathsf{bit-period}$ and $\mathcal{A}$ adopts early-detection strategy on *early*.

The second attack strategy is if a dishonest prover $\mathcal{P}^*$ guesses the last challenge bit and starts sending the response back after knowing $\mathbf{a}^{(i)}_{k-1}$. 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 the probability of success will tend towards that of conventional distance fraud (simply guessing the response early) given in Section 3.

The final practical requirements of a good DB protocol is that the response function should be quick with a constant, predictable time. In HB$^+$DB the response function consists of simple bit operations (which happens in a predictable short time period). The response itself is a single bit, which facilitates quick transmission and is resistant to the multi-bit issues discussed above pertaining to the challenge.

To summarise, the chosen channel has to meet the below identified requirements: the prover early detects; the prover has his own clock source, but maintains reliable data reception; and predictable, minimal response time (simple response calculation and single-bit response length).

5. 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 new *hybrid* protocol called HB$^+$DB. From a distance-bounding perspective, our work shows that using the LPN problem as the basis of the response function appears to be a promising direction. The protocol exhibits resistance to all

three main threat scenarios that distance-bounding aims to prevent, as well as to the GRS MiM key recovery attack (the main weakness of basic HB-based protocols). We evaluate 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 strategies. Our HB$^+$DB protocol detects the GRS attack by accurately timing the challenge-response exchanges (classical DB technique). We also briefly discuss our protocol proposal in light of other physical-layer attack strategies and list the practical requirements needed for secure implementation.

6. ACKNOWLEDGEMENTS

This work was partially supported by the STINT grant "Cross-layer authentication for wireless networks".

7. REFERENCES

[1] N. J. Hopper and M. Blum, "Secure human identification protocols," in *Proc. of ASIACRYPT*, pp. 52–66, 2001.

[2] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols," in *Proc. of CRYPTO*, vol. 3621 of *LNCS*, pp. 293–308, Springer, 2005.

[3] A. Mitrokotsa, M. Beye, and P. Peris-Lopez, *Unique Radio Innovation for the 21st Century*, ch. Security Primitive Classification of RFID Attacks, pp. 39–63. Springer, 2011.

[4] H. Gilbert, M. Robshaw, and H. Sibert, "Active attack against HB$^+$: a provably secure lightweight authentication protocol," *Electronics Letters*, vol. 41, pp. 1169–1170, Oct 2005.

[5] J. Bringer, H. Chabanne, and 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)*, pp. 28–33, 2006.

[6] D. Duc and K. Kim, "Securing HB$^+$ against GRS Man-in-the-Middle Attack," in *Proc. of Symposium on Cryptography and Information Security (SCIS)*, 2007.

[7] J. Munilla and A. Peinado, "HB-MP: A further step in the hb-family of lightweight authentication protocols," *Computer Networks*, vol. 51, no. 9, pp. 2262–2267, 2007.

[8] H. Gilbert, M. J. B. Robshaw, and Y. Seurin, "HB$^\#$: Increasing the security and efficiency of HB$^+$," in *Proc. of EUROCRYPT*, pp. 361–378, Springer-Verlag, 2008.

[9] G. P. Hancke and M. G. Kuhn, "An RFID Distance Bounding Protocol," in *Proc. of SECURECOMM*, pp. 67–73, ACM, 2005.

[10] J. Reid, J. M. Gonzalez Nieto, T. Tang, and B. Senadji, "Detecting Relay Attacks with Timing-based Protocols," in *Proc. of ASIACCS)*, pp. 204–213, March 2007.

[11] A. Mitrokotsa, C. Onete, and S. Vaudenay, "Mafia fraud attack against the rc distance-bounding protocol," in *Proc. of IEEE RFID Technology and Applications (IEEE RFID T-A)*, pp. 74–79, 2012.

[12] I. Boureanu, A. Mitrokotsa, and S. Vaudenay, "Practical & provably secure distance bounding," *Journal of Computer Security*, 2015.

[13] G. Avoine, M. Bingöl, S. Kardaş, C. Lauradoux, and B. Martin, "A framework for analyzing RFID distance bounding protocols," *J. Comput. Secur.*, 2011.

[14] J.-P. Aumasson, A. Mitrokotsa, and P. Peris-Lopez, "A note on a privacy-preserving distance-bounding protocol," in *Proc. of ICICS 2011*, LNCS, (Beijing, China), pp. 78–92, November 2011.

[15] A. Mitrokotsa, C. Dimitrakakis, P. Peris-Lopez, and J. C. H. Castro, "Reid et al.'s distance bounding protocol and mafia fraud attacks over noisy channels," *IEEE Communications Letters*, vol. 14, pp. 121–123, February 2010.

[16] A. Mitrokotsa, P. Peris-Lopez, C. Dimitrakakis, and S. Vaudenay, "On selecting the nonce length in distance-bounding protocols," *The Computer Journal*, 2013.

[17] A. Mitrokotsa, C. Onete, and S. Vaudenay, "Location leakage in distance bounding: Why location privacy does not work," *Computers & Security*, vol. 45, pp. 199–209, 2014.

[18] C. Dimitrakakis, A. Mitrokotsa, and S. Vaudenay, "Expected loss analysis for authentication in constrained settings," *Journal of Computer Security*, 2015.

[19] F. Armknecht, M. Hamann, and V. Mikhalev, "Lightweight authentication protocols on ultra-constrained RFIDs - myths and facts," in *Proc. of RFIDSec*, 2014.

[20] Y. Desmedt, "Major Security Problems with the 'Unforgeable' (Feige)-Fiat-Shamir Proofs of Identity and How to Overcome them," in *Proc. of SecuriCom*, pp. 15–17, SEDEP Paris, France, 1988.

[21] C. Cremers, K. B. Rasmussen, and S. Čapkun, "Distance hijacking attacks on distance bounding protocols," in *Proc. of IEEE Symposium on Security and Privacy*, pp. 113–127, 2012.

[22] G. Avoine and A. Tchamkerten, "An Efficient Distance Bounding RFID Authentication Protocol: Balancing False-Acceptance Rate and Memory Requirement," in *Information Security*, vol. 5735 of *LNCS*, pp. 250–261, Springer, 2009.

[23] 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.

[24] A. Francillon, B. Danev, and S. Capkun, "Relay attacks on passive keyless entry and start systems in modern cars," in *Proc. of NDSS*, 2011.

[25] J. Clulow, G. P. Hancke, M. G. Kuhn, and 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*, vol. 4357 of *LNCS*, pp. 83–97, Springer, 2006.

[26] K. B. Rasmussen and S. Čapkun, "Realization of RF Distance Bounding," in *Proc. of USENIX Security Symposium*, 2010.

[27] G. P. Hancke, "Design of a secure distance-bounding channel for RFID," *Journal of Network and Computer Applications*, vol. 34, pp. 877–887, 2011.

[28] G. Hancke and M. Kuhn, "Attacks on time-of-flight distance bounding channels," in *Proc. of WISec 2008*, pp. 194–202, ACM, 2008.