The Bussard-Bagga and Other Distance-Bounding Protocols under Attacks

Aslı Bay, Ioana Boureanu, Aikaterini Mitrokotsa, Iosif Spulber, and Serge Vaudenay

EPFL, Lausanne, Switzerland
http://lasec.epfl.ch

Abstract. The communication between an honest prover and an honest verifier can be intercepted by a malicious man-in-the-middle (MiM), without the legitimate interlocutors noticing the intrusion. The attacker can simply relay messages from one party to another, eventually impersonating the prover to the verifier and possibly gaining the privileges of the former. This sort of simple relay attacks are prevalent in wireless communications (e.g., RFID-based protocols) and can affect several infrastructures from contactless payments to remote car-locking systems and access-control verification in high-security areas. As the RFID/NFC technology prevails, a practical and increasingly popular countermeasure to these attacks is given by distance-bounding protocols. Yet, the security of these protocols is still not mature. Importantly, the implications of the return channel (i.e., knowing whether the protocol finished successfully or not) in the security of some distance-bounding protocols have not been fully assessed. In this paper, we demonstrate this by a series of theoretical and practical attacks.
We first show that the Bussard-Bagga protocol DBPK-Log does not fulfill its goal: *it offers no protection against distance fraud and terrorist fraud*. Then, we show how to mount several concrete MiM attacks against several follow-up variants, including the protocol by Reid et al.

1 Introduction

Relay attacks are man-in-the-middle (MiM) attacks that enable an adversary to impersonate a prover to a verifier by acting as a carrier for their legitimate messages. *Distance-bounding* (DB) protocols are lightweight authentication protocols considered as a main countermeasure against relay attacks. As their name suggests, distance-bounding protocols enable a verifier to establish an *upper bound* on the physical distance to an un-authenticated prover. This is achieved by measuring the round trip time-of-flights during several challenge-response bit-exchanges. Such DB protocols were first introduced in 1993 by Brands and Chaum [7] in order to preclude MiM attacks against ATMs, whilst the idea of measuring times-of-flight to protect against MiM goes back to Beth and Desmedt [4]. A broad range of distance-bounding protocols followed, being proposed for RFID communications [21, 24, 25, 30, 33, 39], ultra-wideband (UWB) devices [18, 26, 27], wireless ad-hoc networks [11, 14, 38], sensor networks [29], etc.

The importance of distance-bounding protocols in preventing relay attacks can be easily assessed if we simply look at nowadays ubiquitous applications such as access-control to high-security areas and contactless payments. Indeed, relay attacks have also been launched against bankcards [16] and the demonstrated countermeasure against this type of attacks was again based on an implementation of a distance-bounding protocol [16]. Another aspect to consider is that of car manufacturers, who are now using RFID protocols to lock/unlock cars remotely, even if these protocols are themselves susceptible to relay attacks [17]. Thus, there is a stringent need for secure distance-bounding protocols in order to safeguard the growingly spread use of, e.g., RFID-based security-sensitive protocols.

Some MiM attacks also assume that the adversary has access to a side channel showing whether the protocol completed successfully or not. In the case of, e.g., car-locking systems, this would just consist in looking at whether the car opens. We call this the *return channel*. In the RFID community, it was not immediate to realize that the return channel strongly influences the security of the protocols therein invoked. For instance, the HB+ protocol [23] was proposed to resist MiM attacks, but—as soon as the adversary is given access to the return channel [19]—the protocol was found in fact vulnerable to MiM attacks. In the formalism by Vaudenay [40], adversaries with *no* access to the return channel are called *narrow adversaries*. Again, addressing *non-narrow adversaries* in RFID proved challenging. In this paper, we show how non-narrow adversaries can successfully mount a series of attacks on several DB protocols.

Distance-bounding protocols should resist to *distance fraud*. I.e., a malicious prover who is far away from a verifier should not succeed to pass the protocol. Another well established threat model against distance-bounding is the so-called notion of *terrorist fraud* [15]. In this model, a malicious prover who is far away from a verifier tries to mount a distance fraud with the help of an adversary, but without giving him any credential or an advantage that he may later abuse. To defeat terrorist fraud, Bussard and Bagga [8–10] proposed a protocol in which passing the distance-bounding phase would require the knowledge of a key. Many follow-up protocols were inspired from [8–10].

Contribution & Structure. In this paper, we first look at the Bussard-Bagga DBPK-Log protocols [8–10]. We show that its main goals, namely, resistance to distance fraud and terrorist fraud, are not fulfilled. Then, we consider variants and successors. We provide a non-narrow MiM adversary who tries to learn the credential (i.e., the distance-bounding secret key) from an honest prover. This will enable the MiM to impersonate the prover during the distance-bounding phase at a later stage. Consequently, a distance-bounding protocol susceptible to these attacks would not be compliant with its very *raison d'être*. Namely, we show concrete forms of this MiM attack mounted onto variants/successors of the Bussard-Bagga protocols [9, 10] e.g., instances of the Reid *et al.* protocol [34]. Whilst the main pattern of the attacks follows a simple idea [25] and our extension on it, these frauds become possible via pretty intricate statistical and general theoretical analyses that we detail herein.

This paper is organized as follows. In Section 2, we remind the structure of numerous [34, 39] distance-bounding protocols, and we explain the idea behind a basic MiM attack-scenario [25] that can be launched against this family of distance-bounding protocols. We conclude this section by showing how to extend this simple attack, potentially rendering it more dangerous. In Section 4, we describe several MiM attacks, concrete instantiations of the scenario presented in Section 2. In Section 3 we describe the Bussard and Bagga [8–10] or the so-called DBPK-Log (distance-bounding proof of knowledge based on the discrete logarithm) protocols. We mount a distance fraud and a terrorist fraud against DBPK-Log and motivate to consider variants and successors of DBPK. In Section 4 we consider several proposed encryption functions to be used inside the DBPK distance-bounding protocols and variants, and we expose their failures. In Section 4.4, we present the implementation and evolution of the most elaborate of these attacks, mounted onto the most involved successor of the DBPK distance-bounding protocol proposed in [8, 9]. In Section 5, we briefly discuss about the possible fixes to the DB protocols exposed to our kind of threat. Finally, Section 6 concludes the paper.

2 Distance-Bounding & MiM Attacks

We first describe a family of distance-bounding protocols [34, 39] that use similar methods to generate the responses to be employed in the challenge/response phase. Figure 1 depicts a general view of this family of protocols.

More precisely, in all the protocols that belong in this family, the following steps are executed.

- **Initialization phase:** Let the prover P and the verifier V share a secret key x. This initialization phase is not time critical and is executed as follows. The verifier V chooses a random number N_V and transmits it to the prover P. After receiving N_V, the prover P also chooses a random number N_P and computes a session key a such that $a = f_x(C, N_V, N_P)$, where f denotes a pseudorandom function (PRF) and C represents any additional parameters that may be required, e.g., the identifiers of the prover P and the verifier V. We should note here that the computation of the session key a varies. In particular, in the protocol in [39] the session key a depends on both random nonces N_V and N_P.
- **Distance-bounding phase:** This time-sensitive phase starts right after the initialization phase and involves the exchange of challenges-responses at maximum bit-rate over a period of time. It is repeated m times (rounds), with i varying from 1 to m. The actual number of rounds m is normally dictated by a security parameter. At each round i, the challenge-response delay Δt_i is measured. V starts by choosing a random bit c_i, initializing the clock with zero and transmitting c_i to P. The values received by P are

Verifier V		**Prover** P
shared key x		shared key x
	Initialization phase	
$N_V \leftarrow \{0,1\}^m$	$\xrightarrow{N_V}$	
		$N_P \leftarrow \{0,1\}^m$ $a := f_x(C, N_V, N_P)$
	$\xleftarrow{N_P}$	
$a := f_x(C, N_V, N_P)$		
	Distance-bounding phase for $i = 1$ to m	
Pick $c_i \in_U \{0,1\}$ **Start Clock**		
	$\xrightarrow{c_i}$	
		$r_i := \begin{cases} a_i, & if\ c_i = 0 \\ a_i \oplus x_i, & if\ c_i = 1 \end{cases}$
	$\xleftarrow{r_i}$	
Stop Clock verify the responses and that for all rounds $\Delta t_i \leq 2\mathbb{B}$		
	$\xrightarrow{Out_V(\ldots)}$	

Fig. 1. The General Structure of Numerous Distance-Bounding Protocols (present in [34, 39])

denoted by c_i. Next, P answers by sending

$$r_i := \begin{cases} a_i, & if\ c_i = 0, \\ a_i \oplus x_i, & if\ c_i = 1. \end{cases}$$

Assuming noiseless communication, we can denote the values received by V also by r_i. Upon receiving r_i, V stops the clock and stores the received answer and the delay time Δt_i.

After the end of the distance-bounding phase, a final *verification* phase is performed. The verifier V checks the correctness of the received responses and if the response-times Δt_i are bellow the maximum allowed response-time $2\mathbb{B}$. At the end of this phase the verifier V indicates if the prover P is authenticated or not ($Out_V = 1$ or $Out_V = 0$, respectively).

On noisy/noiseless channels. On the one hand, some of the DB provers used in distance-bounding are cheap and, whilst having time constraints also, do not usually cater for error correction. On the other hand, adversaries can be equipped with powerful devices which are less error-prone and/or are also able to correct all the errors/noise of the channels. For simplicity, we will mainly consider noiseless channels in our MiM attacks to follow. However, we note here that noise can be considered easily [21]. I.e., we can augment the model, requiring that the verifier allows at most $m - \tau$ incorrect answers (answers with errors or delayed). Consider

the probability p of one response being correct (non-erroneous and not delayed). Then, the probability that at least τ responses out of m are of the correct kind is given by

$$B(m, \tau, p) = \sum_{i=\tau}^{m} \binom{m}{i} p^i (1-p)^{m-i}.$$

This could then be easily factored into the success probability of our attacks.

On (the feasibility of) MiM attacks. The MiM attacks to be described in here are mounted along the following intuitive pattern. In a *learning phase*, the MiM adversary $\mathcal{A}$ can "play" with a prover P, which could be close to V, and "play" with a verifier V. The subsequent *attack phase* allows interaction several times with a far-away prover and one time with the verifier.

We should note here that if the legitimate prover is located far-away from the legitimate verifier then the MiM attack can be easily deployed. But, we believe that MiM attacks with a prover and his neighboring verifier are also easy to mount when the adversary can initiate protocols with both of them, using a different channel with each. For instance, an attacker could interfere with the initial frequency-synchronization phase so that each of the participants (prover and verifier) would end up communicating with the adversary through two different channels (i.e., frequency bands). Then, the prover does not even realize that another concurrent conversation is taking place (as such a prover and the verifier cannot confer, i.e., two users cannot communicate with each other as long as they "occupy" different frequency bands).

Non-Narrow MiM Attack by Flipping One Challenge [25]: A Case in Point for DB Insecurity. Linear, active MiM attacks can be launched against any protocol from the family in Figure 1. The generic sketch of these attacks was briefly described[1] in [25] against [39]. In this attack, a powerful, non-narrow adversary $\mathcal{A}$ acts as an active MiM during the distance-bounding phase. We consider a scenario where a legitimate verifier and a prover run successfully the initialization phase of the distance-bounding protocol and both of them compute the session key a (see Figure 1). We further assume that the prover and the verifier are close to each other. During the distance-bounding phase, the adversary injects a modified version c'_j of a challenge c_j (for some fixed $j \in \{1, \ldots, n\}$). Let r'_i be the response that the attacker sends to V for c_i and r_i be the response that the prover sends to $\mathcal{A}$. In fact, the attacker will act such that $r'_i = r_i$ for $i \neq j$ and will let r'_j be a random bit. In noiseless condition, it is the case that by looking at the output of the verifier and knowing what choice he made for r'_j, this non-narrow attacker can make a simple calculation to find the jth bit of the key x, i.e., :

$$x_j = r'_j \oplus r_j \oplus \overline{Out_V}$$

[1] And, to this end, the *Swiss-Knife* protocol [25] was designed to defeat this attack by introducing a MAC in the protocol exchanges.

By repeating this strategy for $j = 1, \ldots, m$, the adversary is able to deduce all the bits of the secret key x. In noiseless conditions, the success probability of the attack is 1.

We now extend this attack in [25] to flipping more than one challenge in one distance-bounding session. We will use both flavors of this scenario in what follows.

An Extended Variant of Non-Narrow MiM Attacks: Flipping a Batch of Challenges. As a variant of this attack, the attacker can select a "small size" batch of challenges, i.e., $J \subseteq \{1, \ldots, m\}$, and do $c'_j = \bar{c}_j$ for all $j \in J$ and $c'_i = c_i$ for all $i \notin J$. Let the responses r'_js by the attacker for all $j \in J$ be as he pleases and let $r'_i = r_i$ for all $i \notin J$. If $Out_V = 1$, the attacker deduces $x_j = r_j \oplus r'_j$, for all $j \in J$. This happens with probability $2^{-|J|}$ in a single protocol version, but may allow the finding of a batch of the bits of the key at once.

3 The Bussard-Bagga Protocols and Terrorist Fraud

3.1 The Bussard-Bagga Protocols

Bussard and Bagga have proposed in [8–10] a distance-bounding protocol relying on public key cryptography, e.g., commitments and proofs of knowledge. It is also called DBPK (distance-bounding proof of knowledge). The protocol uses a proof of knowledge in order to protect against terrorist frauds.

In the generic DBPK protocol, the prover P has a secret key x and a published certificate on its public key $y = \Gamma(x)$. The protocol is composed of four phases: the *initialization* phase, the *distance-bounding* phase, the phase for the *opening of commitments* therein used, and the *proof of knowledge* phase.

- **Initialization phase:** In the initialization phase, the prover generates a random secret session key $k \in_R \{0,1\}^m$ and uses this session key in order to encrypt his private key x. The encryption of x is done using a publicly known symmetric key encryption method $\mathcal{E} : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}^m$. Thus, we have $e = \mathcal{E}_k(x)$. Of course, the knowledge of both e and k reveals the private key x, i.e., $x = \mathcal{D}_k(e)$, where $\mathcal{D}_k$ is the decryption function inverting, under k, the encryption $\mathcal{E}_k$. After encrypting x and computing e, the prover P uses a secure bit commitment scheme to commit to each bit of k and e using randomnesses v and v' respectively. More precisely, if we let i be a bit-index ($i \in 1, \ldots, m$), then the commitments to the ith bit of the session key k and of the key e are respectively denoted as

$$z_{k,i} = commit(k_i, v_i) \text{ and } z_{e,i} = commit(e_i, v'_i).$$

- **Distance-Bounding phase:** In the distance-bounding phase, a number of m single-bit challenge-response exchanges take place at maximum bit rate. At each round i, the challenge-response delay Δt_i is measured. The verifier

V selects a random bit as the challenge c_i and the prover responds with a response r_i such that

$$r_i := \begin{cases} k_i, \ if \ c_i = 0, \\ e_i, \ if \ c_i = 1. \end{cases}$$

Verifier V		**Prover** P
public key $y = \Gamma(x)$		private key x
	Initialization phase	
		session key $k \leftarrow \{0,1\}^m$ $e = \mathcal{E}_k(x) \in \{0,1\}^*$ $v, v' \leftarrow (\{0,1\}^*)^m$ for $i = 1$ to m $z_{k,i} = commit(k_i, v_i)$ $z_{e,i} = commit(e_i, v'_i)$ endfor $z_k := (z_{k,i})_{i \in \{1,\ldots,m\}}$ $z_e := (z_{e,i})_{i \in \{1,\ldots,m\}}$
	$\xleftarrow{z_k, z_e}$	
	Distance-bounding phase for $i = 1$ to m	
Pick $c_i \in_U \{0,1\}$ **Start Clock**		
	$\xrightarrow{c_i}$	
		$r_i := \begin{cases} k_i, \ if \ c_i = 0 \\ e_i, \ if \ c_i = 1 \end{cases}$
	$\xleftarrow{r_i}$	
Stop Clock		
	Commitment opening phase for $i = 1$ to m	
		$\gamma_i := \begin{cases} v_i, \ if \ c_i = 0 \\ v'_i, \ if \ c_i = 1 \end{cases}$
	$\xleftarrow{\gamma_i}$	
verify the responses $z_{k,i} \stackrel{?}{=} commit(k_i, \gamma_i)$, if $c_i = 0$ $z_{e,i} \stackrel{?}{=} commit(e_i, \gamma_i)$, if $c_i = 1$		
	Proof of knowledge phase	
	$\xleftrightarrow{PK[(\alpha,\beta): z = \Omega(\alpha,\beta) \wedge y = \Gamma(\alpha)]}$	

Fig. 2. The DBPK Protocol proposed by Bussard and Bagga [8–10]

– **Commitments' opening phase:** In this phase, the prover P opens some commitments on the bits of k and e corresponding to his answers in the distance-bounding phase. I.e., to vouch for his DB responses, the prover sends the randomness v_i to open $z_{k,i}$ if challenge i was 0, and he sends v'_i otherwise. This is denoted in Figure 2 through sending the value γ_i. In case that the openings of $z_{k,i}$ and $z_{e,i}$ do not pass, the verifier V sends an error notification message to the prover P.

– **Proof of knowledge phase:** In this phase, the prover P convinces the verifier V with a zero-knowledge interaction that he has generated the commitments which correspond to a unique private key x and this private key corresponds to the public key y that is used by the verifier to authenticate the prover. The proof of knowledge is denoted as

$$PK[(\alpha, \beta) : z = \Omega(\alpha, \beta) \wedge y = \Gamma(\alpha)],$$

where the knowledge of α, β is being proven, while z, y are as per the protocol, known to the verifier. In the protocol, we have $y = \Gamma(x)$ and $z = \Omega(x, (v, v'))$. The value of z can be computed from the $z_{k,i}$ and $z_{e,i}$.

The number m of DB rounds and the size m of the key is dictated by a security parameter. Typically, m varies between 128 and 1024.

3.2 Commitments and the Proof of Knowledge in DBPK-Log

The only instances of DBPK providing concrete commitments and proofs of knowledge are based on the discrete logarithm in $\mathbf{Z}_p^*$ and are called DBPK-Log. We now describe these commitments and proofs of knowledge.

We use a strong prime p, two generators g, h of $\mathbf{Z}_p^*$, an element x of $\mathbf{Z}_{p-1}^*$, and $y = g^x \bmod p$.[2]

We have $commit(b, v) = g^b h^v \bmod p$. The main property of this commitment is that given all $z_{k,i}$, $z_{e,i}$, v_i, v'_i, we can form $z = \prod_i (z_{k,i} z_{e,i})^{2^{i-1}}$, $v = \sum_i (v_i + v'_i) 2^{i-1}$, and obtain that

$$z = commit((k + e) \bmod (p - 1), v).$$

The proposed encryption methods use $e = (ux - k) \bmod (p - 1)$ with either $u = 1$ [10] or u random and publicly revealed [8, 9]. So, the proof of knowledge consists of proving knowledge of x and v such that $y = g^x$ and $z = g^{ux} h^v$ in $\mathbf{Z}_p$.

The proof of knowledge [9] is repeating t times what follows: the prover sends $w_1 = g^{u\rho_1} h^{\rho_2} \bmod p$ for some random $\rho_1, \rho_2 \in \mathbf{Z}_{p-1}$; the verifier sends some challenge $c \in \{0, 1\}$; the prover responds by $s_1 = \rho_1 - cx \bmod (p - 1)$ and $x_2 = \rho_2 - cv \bmod (p - 1)$; the verifier checks $w_1 = z^c g^{us_1} h^{s_2} \bmod p$ and $w_2 = y^c g^{s_1} \bmod p$.

3.3 Terrorist Fraud and Distance Fraud against DBPK-Log

We now show that the public-key techniques which are used in the DBPK-Log protocol are ineffective to defeat terrorist fraud. For this, we consider a malicious prover who is far away from an honest verifier. There is an adversary close to the verifier who will get some help from the prover to pass the protocol without getting any advantage to further impersonate the prover. The attack is sketched in Fig. 3.

[2] In [8–10], h is not necessarily a generator and $x \in \mathbf{Z}_{p-1} \backslash \{q\}$ with $q = \frac{p-1}{2}$.

Verifier V	**Adversary** A	**Prover** P
	Initialization phase	
	$\xleftarrow{z}$	$z = g^{ux}h^{v} \bmod p$
A guesses the future value c_1 and compute the commitments: set $z_{k,i} := commit(k_i, v_i)$, $z_{e,i} := commit(e_i, v'_i)$ for all $i > 1$ set $z_{k,1} := commit(k_1, v_i)$ if $c_1 = 0$ ($z_{e,1}$ is a free variable) set $z_{e,1} := commit(e_1, v'_i)$ if $c_1 = 1$ ($z_{k,1}$ is a free variable) solve $z = \prod_i (z_{k,i} z_{e,i})^{2^{i-1}}$ in the remaining free variable		
	for $i = 1$ to m	
$z := \prod_i (z_{k,i} z_{e,i})^{2^{i-1}}$	$\xleftarrow{z_{k,i}, z_{e,i}}$	
	Distance-bounding phase	
	for $i = 1$ to m	
	$\xrightarrow{c_i}$	if $i = 1$ and c_1 incorrect, abort
	$\xleftarrow{r_i}$	$r_i := \begin{cases} k_i \text{ if } c_i = 0 \\ e_i \text{ if } c_i = 1 \end{cases}$
	Commitment opening phase	
	for $i = 1$ to m	
	$\xleftarrow{\gamma_i}$	$\gamma_i := \begin{cases} v_i \text{ if } c_i = 0 \\ v'_i \text{ if } c_i = 1 \end{cases}$
	Proof of knowledge phase	
verify z	$\xleftrightarrow{PoK}$	prove x, v for z

Fig. 3. Terrorist Fraud against DBPK-Log

First, the malicious prover selects u and $v \in \mathbf{Z}_{p-1}$ (with either $u = 1$ or a random u, as specified in DBPK-Log), then computes $z = g^{ux}h^v \bmod p$ and sends z to the adversary. The adversary selects some random k_i, e_i, v_i, v'_i, $i = 1, \ldots, m$, and a random bit c_1. Then, he computes $z_{k,i} = commit(k_i, v_i)$ and $z_{e,i} = commit(e_i, v'_i)$ for $i = 2, \ldots, m$. If $c_1 = 0$, he sets $z_{k,1} = commit(k_1, v_1)$ and $z_{e,1}$ remains free. If $c_1 = 1$, he sets $z_{e,1} = commit(e_1, v'_1)$ and $z_{k,1}$ remains free. Then, he can solve the equation $z = \prod_i (z_{k,i} z_{e,i})^{2^{i-1}} \bmod p$ in the remaining free variable. Next, the adversary runs the DBPK-Log initialization phase, distance-bounding phase, and opening phase using these values. Note that if the value of the challenge c_1 received from the verifier differs from the value of the bit c_1 which were selected, the attack aborts.[3] Otherwise, it is straightforward to see that the adversary can answer all challenges and open all commitments. Then, the verifier will compute z which matches the one selected by the malicious prover. Finally, the adversary relays the proof of knowledge for x and v (such that $z = g^{ux}h^v \bmod p$).

Clearly, this attack succeeds with probability $\frac{1}{2}$ (or even 1 if the verifier allows an error in the first round). It is also clear that since the proof-of-knowledge is zero-knowledge and that x is not used anywhere else, that the adversary learns no information about x. So, it is a valid terrorist fraud.

Distance Fraud. In distance-fraud settings, the malicious prover could simulate the adversary in the above attack and select $k_i = e_i$, $i = 2, \ldots, m$. This way, the prover could answer to c_i by $r_i = k_i = e_i$ before receiving the challenge c_i and therefore making the correct response to arrive on time to the verifier.

Equivalently, a malicious prover could just simulate the honest prover selecting $k \approx \frac{x}{2}$ and $u = 1$ and answering to the challenges by anticipation. Concretely, if x is even, the malicious prover can select $k = e = \frac{x}{2}$ and $u = 1$ and we have $r_i = k_i = e_i$ for every i. If x is odd, he could select $k = \frac{x-1}{2}$, $e = \frac{x+1}{2}$, and $u = 1$ so that $r_i = k_i = e_i$ for $i \geq 2$. The $i = 1$ case can be counted in the error tolerance.

4 Non-Narrow MiM Attacks against DBENC

We now consider the protocol by Reid *et al.* [34] which is a variant of DBPK no longer based on public-key cryptography. Indeed, the prover and the verifier now share the secret x. In addition to this, k and e are derived from PRF computations (done at both sides) based on x and some nonces. (See Fig. 4.) This protocol, that we call DBENC, has no longer any commitment or proof of knowledge phase.

Clearly, the attack scenario in Section 2 lends itself to concrete attacks against the DBENC protocol. First consider that a non-narrow MiM adversary specializes in the batch-version of the attack on the DB phase. Namely, let first $|J| \geq 2$. This attacker selects indexes $J \subseteq \{1, \ldots, m\}$ for flipped challenges. Then, for this

[3] The attack could also go on with the adversary taking c_1 as the value he selected, and counting on that the verifier will accept this error as due to noise.

Verifier V		**Prover** P
shared key x		shared key x
	Initialization phase	
$N_V \leftarrow \{0,1\}^m$	$\xrightarrow{ID_V, N_V}$	$N_P \leftarrow \{0,1\}^m$ $k := f_x(ID_P \| ID_V \| N_V \| N_P)$ $e = \mathcal{E}_k(x)$
$k := f_x(ID_P \| ID_V \| N_V \| N_P)$ $e = \mathcal{E}_k(x)$	$\xleftarrow{ID_P, N_P}$	
	Distance-bounding phase for $i = 1$ to m	
Pick $c_i \in_U \{0,1\}$ **Start Clock**	$\xrightarrow{c_i}$	
		$r_i := \begin{cases} e_i, & if\ c_i = 0 \\ k_i, & if\ c_i = 1 \end{cases}$
	$\xleftarrow{r_i}$	
Stop Clock verify the responses and for all rounds $\Delta t_i \leq 2\Delta t_{max}$	$\xrightarrow{Out_V(\ldots)}$	

Fig. 4. Protocol proposed by Reid *et al.* [34]

protocol, with probability $2^{-|J|}$, he obtains the following: 1. the bits e_j and k_j, for all $j \in J$; 2. either the bit e_i or the bit k_i, when $j \notin J$. Secondly, for the case that $|J| = 1$, the success probability of the attacker is 1, as per Section 2. In any case, depending the encryption algorithm, this can help in recovering some piece of information about x.

The designers of DBPK and DBENC suggest [8–10, 34] several possible encryption methods ($\mathcal{E}$) to be used therein. In the next subsections, we consider them in a case-by-case fashion (see Sections 4.1, 4.2, 4.3), showing the concrete MiM attack that would break the corresponding instance of the DBENC protocol. In DBPK-Log, the encryption is based on the modulo $p-1$ arithmetic. We fully describe attacks in these cases (Sections 4.2 and 4.3).

4.1 MiM Attack on the "One-Time-Pad DBENC"

We consider here that $\mathcal{E}_k(x) = x \oplus k$ is to be used as the encryption inside the DBENC protocol, with $x \in \{0,1\}^m$ and $k \in_U \{0,1\}^m$. We denote it simply by $e = x \oplus k$. Of course, for any bit position $i \in \{1, \ldots, m\}$, we then have $e_i = x_i \oplus k_i$. Hence, the attack-schema in Section 2, recovers $x_j = e_j \oplus k_j$ (for $j \in J$) by using a single session with a flipped challenge c_j. By iterating at most m times (in the single-flip case), he fully recovers x.

Reid *et al.* [34] suggested to use a "semantically secure encryption function" and proposed one-time pad: $\mathcal{E}_k(x) = x \oplus k$. Clearly, instances of DBENC based on one-time pad fall under the attack in Section 2.

4.2 MiM Attack on the "Addition Modulo n DBENC"

We consider here that $\mathcal{E}_k(x) = x - k \bmod n$ is to be used as the encryption inside the DBENC protocol, with $x \in \mathbf{Z}_n$, $k \in_U \mathbf{Z}_n$, and some fixed n of m bits.[4] We denote this encryption simply by $e = x - k \bmod n$. Clearly, we have

$$e = x - k + cn, \tag{1}$$

where $c = 1_{k>x}$. For $e_1 = \mathsf{lsb}(e)$, we have

$$e_1 = \begin{cases} x_1 \oplus k_1, & if\ x \geq k, \\ x_1 \oplus k_1 \oplus n_1, & if\ x < k. \end{cases}$$

This can be written as $e_1 = x_1 \oplus k_1 \oplus c \times n_1$.

The actual tactics of the attack will be separated into subcases, upon the characteristic of n, e.g., considering whether n is odd or even. Some of these sub-cases contain cross-references to one another.

The n even case.

• **The subcase where $n = 2^m$.** For $n = 2^m$, of course, $n_1 = 0$. Then, based on (1), we reduce to $e_1 = x_1 \oplus k_1$. Hence, our attacker recovers $x_1 = e_1 \oplus k_1$ by using a single session. Then, once he learned x_1, he can recover x_2 by employing the fact that $e_2 = x_2 \oplus k_2 \oplus \bar{x}_1 k_1$. To see this, first note that if the query to the prover reveals e_1, the adversary can infer $k_1 = e_1 \oplus x_1$. Otherwise, the query reveals k_1 anyway. Hence, we can take for granted that k_1 is known to the attacker. Since a second iteration of our attack recovers e_2 and k_2, he can clearly deduce $x_2 = e_2 \oplus k_2 \oplus \bar{x}_1 k_1$. He can continue further to uncover all bits of x using m iterations.

• **The subcase where $n = 2^{m'} n'$ with n' odd.** Let $n = 2^{m'} n'$ with n' odd. Again, we are in the case where $e = (x - k) \bmod 2^{m'} n'$. So, we have $e \equiv x - k \pmod{2^{m'}}$. Thus, using the previous attack, the attacker can first recover $x_1, \ldots, x_{m'}$ using m' iterations. Then, he will have to employ the attack-scenario for the case for a modulus which is odd, here for n'; this case is stated below. Combining the two results will give the attacker the bits $x_1, \ldots, x_{m'}$ of x.

The n odd case.

• **Calculating the least significant bit in the n odd case.** For n odd, we have $n_1 = 1$. Then, based on (1), we have $e_1 = x_1 \oplus k_1 \oplus c$ where $c = 1_{k>x}$. Let $p = \mathbb{P}[c = 1]$, then it holds that

$$p = \mathbb{P}[c = 1] = \mathbb{P}[k > x] = 1 - \frac{(x+1)}{n}.$$

Hence, when x is far from $n/2$, c is strongly biased. On the one hand, if $x > n/2$, most of c's are 0, so x_1 is the majority of the obtained $e_1 \oplus k_1$. On the other hand, if $x < n/2$, x_1 is the complement of the majority of the obtained $e_1 \oplus k_1$.

[4] In [10], it is proposed for DBPK-Log with $n = p - 1$ and a strong prime p.

Assuming that the adversary first guesses whether $x > n/2$, then he deduces x_1 by mounting a statistical attack. In more detail, assume that the adversary tries N sessions, i.e.,:

$$\mathsf{Bit}_i = e_1 \oplus k_1 = x_1 \oplus c_i,\ \forall i \in \{1, \ldots, N\},$$

where c_i depends on the session i since k is refreshed in every session. The adversary uses a majority function to find x_1 such as

$$\mathsf{Majority}(\mathsf{Bit}_1, \ldots, \mathsf{Bit}_N) \approx \begin{cases} x_1, & if\ x > \frac{n}{2}, \\ x_1 \oplus 1, & if\ x < \frac{n}{2}. \end{cases}$$

More precisely, $\mathbb{P}[\mathsf{Bit}_i = x_1 \oplus 1_{x<n/2}] = 1/2 + |p - 1/2|$. Thus, thanks to the Chernoff bound (*Lemma* 2, in the Appendix),

$$\mathbb{P}\left[\mathsf{Majority}(\mathsf{Bit}_1, \ldots, \mathsf{Bit}_N) = x_1 \oplus 1_{x<n/2}\right] \geq 1 - e^{-2N(p-1/2)^2}.$$

We can just take $N \approx (1/2 - (x+1)/n)^{-2}$ to deduce x_1 by the guess $1_{x<n/2}$.

• **Calculating the i-th least significant bit in the n odd case.** Assuming that $x_1, \ldots, x_{i-1}$ have been recovered (possibly from a guess on the sign of $x - n/2$), similarly as before, the queries to the prover leak $k_1, \ldots, k_{i-1}$. Thus, based on (1), we have

$$e_i = x_i \oplus k_i \oplus B(c, k_1, \ldots, k_{i-1}, x_1, \ldots, x_{i-1})\ , \tag{2}$$

where $c = 1_{k>x}$ and B is a Boolean function computing the carry on the ith bit in $x - k + cn$. To express the dependence of B on the overall value of c, we write

$$\begin{aligned} &B(c, k_1, \ldots, k_{i-1}, x_1, \ldots, x_{i-1}) = \\ &c\alpha(k_1, \ldots, k_{i-1}, x_1, \ldots, x_{i-1}) \oplus \beta(k_1, \ldots, k_{i-1}, x_1, \ldots, x_{i-1}). \end{aligned}$$

There are many cases where $\alpha(k_1, \ldots, k_{i-1}, x_1, \ldots, x_{i-1}) = 0$ (i.e., B does not depend on c and can be computed by the attacker using just the information on the bits 0 to $i-1$ of the known parts). So, by Equation (2), the attacker can recover x_i in one session. There are rare cases in which we have $\alpha(k_1, \ldots, k_{i-1}, x_1, \ldots, x_{i-1}) = 1$, whatever $k_1, \ldots, k_{i-1}$. Such a case is given by $i = 0$, as shown in the last paragraph. So, in these corner cases, a statistical attack is needed in order to recover x_i.

• **Calculating several most significant bits and the least significant one bit in the n odd case.** Going back to the attack on the least significant bit, we realize that the statistics on the estimate of $\mathbb{P}[c = 1] = \mathbb{P}[k > x]$ reveals several most significant bits of x.

Let $B = (1/N)\sum_{i=1}^{N} \mathsf{Bit}_i$, the average of Bit_i's. Note that $\mathbb{E}[\mathsf{Bit}_i] = \mathbb{E}[c_i]$ or $\mathbb{E}[\mathsf{Bit}_i] = 1 - \mathbb{E}[c_i]$. Hence, we get either $\mathbb{E}[B] = \mathbb{E}[c_i] = 1 - (x+1)/n$ or $\mathbb{E}[B] = 1 - \mathbb{E}[c_i] = (x+1)/n$. It holds that

$$\left|\frac{1}{2} - \mathbb{E}[B]\right| = \left|\frac{(x+1)}{n} - \frac{1}{2}\right| \tag{3}$$

and we can recover the ℓ most significant bits of x by using Hoeffding's Bound (*Lemma* 1, in the Appendix), i.e., $\mathbb{P}\left[|B - \mathbb{E}[B]| \leq \varepsilon\right] > 1 - 2e^{-2N\varepsilon^2}$. To recover the ℓ most significant bits, we set $\varepsilon = 2^{-\ell}$ and $N = 2^{2\ell}$. Notice that this attack is similar to the statistical attack described before, hence, the attacker can get x_1, as well.

• **Calculating the least significant bit in the n odd case, revisited.** Assuming that the adversary guesses the most significant bit x_m of x, then he pays attention to (x_m, c_m, r_m). In many cases, c can be deduced non-ambiguously from this. For instance, if $c_m = 0$ (so $r_m = k_m$) and $x_m \neq r_m$, then $c = k_m$ for sure. In the case where $x_m = 0$, if $c_m = 1$ (so $r_m = e_m$) and $r_m = 1$, then $c = 1$ for sure. If c cannot be deduced, the adversary waits for another session. Hence, with $J = \{1\}$, he directly deduces

$$x_1 = e_1 \oplus k_1 \oplus c.$$

He can proceed to recover all other bits by ruling out sessions in which c cannot be deduced. This way, he can always compute c and deduce

$$x_i = e_i \oplus k_i \oplus B(c, k_1, \ldots, k_{i-1}, x_1, \ldots, x_{i-1}).$$

This can be used as follows to recover all bits based on the guess of x_m: assuming $x_m, x_1, \ldots, x_{j-1}$ are known, the attack with $J = \{j\}$ will reveal the response to either k_i or e_i for $i = 1, \ldots, j-1$ (while x_i is known), and the response to both k_j and e_j. So, for each $i \leq j$ we know two out of the three values x_i, k_i, e_i. The above relation allows to compute the third one iteratively for each $i \leq j$, so deduce x_j. The expected number of sessions to recover all bits is here less than $4m$. Indeed, the key has length m and the probability of deducing c is larger than the probability that $c_0 = 0$ and $x_m \neq k_m$, which is $1/4$.

4.3 MiM Attack on the "Modular Addition with Random Factor DBENC"

We consider here $\mathcal{E}_k(x) = (u, ux - k \bmod n)$ with $x \in \mathbf{Z}_n$, $k \in_U \mathbf{Z}_n$, $u \in_U \mathbf{Z}_n^*$ freshly selected for each encryption, and some number prime n of m bits.[5] We denote $e = ux - k \bmod n$.

We have

$$e = (ux \bmod n) - k + cn,$$

where $c = 1_{k > ux \bmod n}$.

By looking at the least significant bit (i.e., $J = \{1\}$), we note that

$$\mathsf{lsb}(ux \bmod n) = e_1 \oplus k_1 \oplus c.$$

[5] In [9], the value $u = 0$ is authorized. However, since it does not make decryption possible, we omit it in this section. Our results are not affected by this choice.

For simplicity reasons, we assume that $x \in \mathbf{Z}_n^*$. Therefore, $(ux \bmod n) \in_U \mathbf{Z}_n^*$. In sessions where k_m and k_{m-1} are revealed and happen to be both 1 (i.e., $c_m = c_{m-1} = 0$ and $r_m = r_{m-1} = 1$), the probability that $c = 1$ is

$$\mathbb{E}\left[\frac{k-1}{n-1}\middle| k \geq 2^{m-1} + 2^{m-2}\right] \geq \frac{3}{4}.$$

Hence, we can rule out other sessions (we need 16 sessions to have one unruled), and consider that c is a biased bit with $\mathbb{P}[c = 1] \geq 3/4$.

Next, by writing $u = 2u' \bmod n$, we observe that

$$\mathsf{lsb}(ux \bmod n) = \mathsf{lsb}\left(2(u'x \bmod n) - n1_{u'x \bmod n \geq \frac{n}{2}}\right) = 1_{u'x \bmod n \geq \frac{n}{2}}.$$

The function $1_{u'x \bmod n \geq \frac{n}{2}}$ is sometimes called the most significant bit of $u'x \bmod n$ in the literature, although it is not always the most significant bit in the binary representation. For this reason, we write it $M_1(u'x \bmod n)$. Hence, our attack recovers some $(u', M_1(u'x \bmod n) \oplus c)$ pairs. The problem of finding x is then called the *Hidden Number with Noise Problem (HNNP)* [5].

More generally, we have $\mathsf{lsb}_\ell(ux \bmod n) = e + k - cn \bmod 2^\ell$. By setting $J = \{1, \ldots, \ell\}$, we recover $\mathsf{lsb}_\ell(ux \bmod n)$ with probability $2^{-\ell}$. We filter sessions for which $k_m, \ldots, k_{m-\ell'+1}$ are revealed and are all 1 so that $\mathbb{P}[c = 1] \geq 1 - 2^{-\ell'}$. We need $2^{\ell+2\ell'}$ sessions to get one sample. We have that $\mathsf{lsb}_\ell(ux \bmod n) = \mathsf{lsb}_\ell\left(2^\ell(u'x \bmod n) - n\left\lfloor \frac{u'x \bmod n}{n/2^\ell}\right\rfloor\right)$, for $u = 2^\ell u' \bmod n$. This is an one-to-one mapping of $M_\ell(u'x \bmod n) = \left\lfloor \frac{u'x \bmod n}{n/2^\ell}\right\rfloor$.

The *Hidden Number Problem (HNP)* was introduced by Boneh and Venkatesan [5] to prove the hardness of the most significant bits of the secret keys in Diffie-Hellman schemes. In HNP, for a given prime n and a given generator g of $\mathbf{Z}_n^*$, the aim is to find a hidden number $\alpha \in \mathbf{Z}_n^*$ by querying an oracle which has access to a function $L_{n,\ell,\alpha}$ such that

$$L_{n,\ell,\alpha}(x) \triangleq M_\ell(\alpha \times g^x \bmod n).$$

This leads to two versions of the problem called *randomized* or *sampling* HNP. In our case, x is chosen uniformly and independently at random in $\mathbf{Z}_n^*$. Boneh-Venkatesan [5] solved this problem by providing an algorithm which works for any $\ell \geq \varepsilon\sqrt{\log n}$ in running time polynomial in $\log n$ for all $\epsilon > 0$. Hence, it can work in practice with a very low ℓ. It also works with noise provided that $2^{\ell'} \geq \log n$.

It requires D samples to recover x according to [32] (where, in [5], D is equal to $2\sqrt{\log n}$). The complexity is $n^{\mathcal{O}(1/\log\log n)}$ for $\ell = \log\log n$ (or even $\ell = 2$ but with an oracle to find the L_∞-closest vector in a lattice). Therefore, we have a practical attack (even though not polynomial) using $2^{2\ell'} \times 2^\ell \times D$ sessions.

The attack runs as follows:

1: **for** each session **do**
2: **if** it leaks $k_m, \ldots, k_{m-\ell'+1}$, and all bits are 1 **then**

```
    if the attack with J = {1, ..., ℓ} reveals lsb_ℓ(e) and lsb_ℓ(k) then
      deduce u' and M_ℓ(u'x mod n) (except with probability ≤ 2^{-ℓ'})
      stop when D such pairs are deduced
    end if
  end if
end for.
```

With the D pairs, run the hidden problem solver algorithm.

The HPN solving algorithm works roughly as follows. Given u'_i and $v_i = M_\ell(u'_i x \bmod n)$ for $i = 1, \ldots, D$, we define the lattice of dimension $D+1$ spanned by the vector $(u'_1, \ldots, u'_D, 1/n)$ and the D vectors $(0, \ldots, 0, n, 0, \ldots, 0, 0)$. In these last vectors, the value n appears iteratively from position 1 to position D, up to the exhaustion of the set. Then, the lattice vector closest to $(v_1, \ldots, v_D, 0)$ is likely to be $(u'_1 x \bmod n, \ldots, u'_D x \bmod n, x/n)$. Indeed, the Euclidean distance between them is about $(n/2^\ell)\sqrt{D}$ and the volume of the lattice is n^{D-1}. Since such a short distance is quite unusual in a volume of this size, this is likely to be indeed the closest vector. If it is found, then x can be deduced.

4.4 Feasibility of the Latter Attack

The latter attack is the most intricate and demanding of all the attacks in this section. Thus, we endeavored in studying its practical feasibility. We hereby report our findings.

Implementation Details. In our implementation, we used a modified version of the lattice reduction of HNP in [5]: the basis reduction is done with the BKZ (Block Korkin-Zolotarev) [36] algorithm[6], and the closest vector is found using Babai's *closest plane* algorithm [3]. These algorithms are implemented in `NTL` [37] and, in our coding of the attack, we use these implementations.

Implementation Results. In our implementation, we are mainly interested in the number of protocol sessions needed to perform the attack, for a given length of the modulus. Table 1 reports the above aim, together with the variation of several key parameters presented in the previous section. When we called the BKZ algorithm with its default NTL parameters, we obtained the respective number of protocol sessions reported in the table. When we manually tuned some of BKZ's parameters through extensive experiments, we succeeded in a slight reduction in this number of sessions. This improvement is reported with underlined figures in our table of results (i.e., in line 3, we show a drop from 2^{16} to 2^{15} in number of sessions, through adjusting BKZ's calling-parameters).

Note that—up to some point—we can achieve a time/accuracy trade-off (and implicitly lower ℓ). Namely, one can *β-block reduce*[7] the basis of a lattice to

[6] This algorithm is a generalization of the famous LLL algorithm [28], from blocks of size 2 to blocks of larger sizes.

[7] Vaguely speaking, a lattice basis is block reduced with block size β if for every β consecutive vectors, the orthogonal projections of them in the span of some previous vectors are reduced in the sense of Hermite and Korkin-Zolotarev [35].

get much better closest vector solutions [36], where β is at most the size of the lattice (in our case, we have $\beta = D + 1$, with $D \in \mathcal{O}(\sqrt{\log n})$ as in the table). If the size of the modulus is small, β will increase, but we can always increase D to lower ℓ (and require less sessions), and choose a sensible value for β. This trade-off can reduce the number of sessions by a few, but it can dramatically increase the running time. These facts were also confirmed experimentally. In the implemented attack, we took $\ell' = 3$. Below, we report values that yield efficient attacks.

Table 1. Number of Sessions per Modulus Size

Prime size	ℓ	D	β	# sessions	Time (s)
8	3	$\lfloor 6\sqrt{\log n} \rfloor = 12$	13	2^{12}	0.002
16	4	$\lfloor 6\sqrt{\log n} \rfloor = 24$	25	2^{14}	0.005
32	4	$\lfloor 6\sqrt{\log n} \rfloor = 30$	31	2^{14}	0.357
64	5	$\lfloor 3\sqrt{\log n} \rfloor = 24$	25	2^{16},$\underline{2^{15}}$	0.18
128	5	$\lfloor 6\sqrt{\log n} \rfloor = 66$	31	2^{17}	71
256	7	$\lfloor 6\sqrt{\log n} \rfloor = 96$	25	2^{20},$\underline{2^{19}}$	267
512	9	$\lfloor 5\sqrt{\log n} \rfloor = 110$	20	2^{26},$\underline{2^{21}}$	1920
1024	15	$\lfloor 3\sqrt{\log n} \rfloor = 96$	23	2^{32},$\underline{2^{27}}$	1616

In the above table, we present the results for the attack working with a probability greater than or equal to 7/8 (i.e., we took $\ell' = 3$). The last column indicates the running time it took to find the secret on a 2GHz Dual Core processor with 4GB of RAM, given the ℓ most significant bits of the key (with ℓ respectively reported in the table).

There may be further optimization to perform onto our implementation. For instance, there is the new BKZ 2.0 [12] that could be implemented separately (i.e., it is not available in NTL). This could yield faster, more accurate solutions for the CVP, which in itself could allow β to go higher. With such a reduced basis, for a higher β, one could replace NTL's versions of Babai's Nearest Plane algorithm (found in NTL) with more efficient algorithms for the CVP such as Enum [1]. We feel however that this is a project in its own right, concerning optimizations of lattice-based implementations, which is out of our scope. Through this section, we desired to show that our computationally most expensive attack could still be deployed in practice, with a significant yet not impossible effort.

5 On Strengthening/Fixing DBENC (and DB in general)

As we showed, the DBENC protocols (e.g., variants of Reid *et al.* [34]) are vulnerable to reasonable MiM attacks. In this section, we assess the possible (existent or to-be-found) alternative.

DBPK appeared with the aim of protecting against terrorist-fraud [15]. Some other protocols followed its path in more recent years. We briefly discuss now such descendants and their countermeasures to attacks. On the one hand, the

Swiss-Knife protocol [25] can counter terrorist fraud, and—by using a MAC and a secret sharing scheme—it eliminates the pattern of insecurity by Kim [25] that we extend to full statistical attacks in here. On the other hand, the Tu and Piramuthu protocol [39] is fully susceptible to the type of attacks described in here (see [31] for more on the insecurity of Tu and Piramuthu's protocol). As per the actual DBPK protocol itself, we also consider that a first way in which it could improve its insecurity status consists of changing the proof of knowledge into the authentication of the protocol exchanges. Another way is that of employing different encryption functions or other cryptographic primitives to those proposed by the authors.

Unfortunately, changing the encryption to some more involved scheme leads to other vulnerabilities against terrorist fraud, as shown by Hancke [20].

Most importantly, the sad aspect in DB is that most arguments of security (and insecurity for that matter) are heuristic. And, other, completely different patterns of attacks have been exposed recently [6] for a large class of DB protocols. That attack pattern demolished the PRF assumption, which has been voidly invoked in some security arguments for DB protocols. In that sense, even in the Bussard-Bagga protocol, it should not be sufficient to replace the encryption by a PRF in a vacuous manner, i.e., with no further conditions/proofs.

We personally consider that the solution of basing the responses on pre-established sub-secrets and on secrets by using a secret sharing scheme [2] (along with other assessed modifications) is viable.

Nevertheless, this brief encounter into the status of DBPK-Log and DBPK's descendants goes to show that the threats exhibited are real and should carefully be considered, especially in implementations of (these terrorist-fraud protecting) DB.

Consequently, we emphasize that what is really missing is a clear, sound security model for DB, where resistance to terrorist-fraud and resistance to MiM can *provably* co-exist.

6 Conclusions

In this paper, we raise the signal that distance fraud, terrorist fraud, and MiM attacks can be launched against allegedly secure existing distance-bounding protocols. We show intricate applications of this attack-scenario, using statistical and theoretical analyses. These concrete frauds are respectively performed against the different proposed instantiations and successors of the DBPK distance-bounding protocols [8–10]. We present a distance fraud and a terrorist fraud against DBPK-Log, thus disproving its very purpose. Our non-narrow MiM adversaries can therein retrieve all or several bits of the secret key, depending on the used encryption scheme in the respective instantiation of DBENC, i.e. variants of Reid *et al.* [34]. We present an implementation and short evaluation of the most demanding of these attacks. Our results show that a bad choice of the encryption function inside concrete instantiations of proposed protocol-schemas can lead to MiM attacks. This vouches for our feeling that this serious attack-strategy

possibly finds applications in many other DB protocols. Simple fixes (replacing the encryption with a PRF) may not be the way forward either [6], since the core issue is the absence of necessary and sufficient conditions to describe DB security. Further, this dwells within the lack of sound, clear security models for DB with accompanying provably secure protocols. We herein provide some cases in point.

Acknowledgements

This work was partially supported by

- the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), under the Swiss National Science Foundation;
- the European Commission through the ICT program under contract ICT-2007-216646 ECRYPT II;
- the Marie Curie IEF Project "PPIDR: Privacy-Preserving Intrusion Detection and Response in Wireless Communications", Grant No. 252323.

References

1. Algorithms for the shortest and closest lattice vector problems. In *Proceedings of IWCC11 (Coding and Cryptology)*, volume 6639 of *Lecture Notes in Computer Science*, pages 159–190, 2011.
2. G. Avoine, C. Lauradoux, and B. Martin. How Secret-sharing can Defeat Terrorist Fraud. In *Proceedings of the 4th ACM Conference on Wireless Network Security – WiSec'11*, Hamburg, Germany, June 2011. ACM Press.
3. L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
4. T. Beth and I. Desmedt. Identification tokens - or: Solving the chess grandmaster problem. In *CRYPTO'90*, volume 537 of *LNCS*, pages 169–177. Springer, 1991.
5. D. Boneh and R. Venkatesan. Hardness of Computing the Most Significant Bits of Secret Keys in Diffie-Hellman and Related Schemes. In N. Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 129–142. Springer, 1996.
6. I. C. Boureanu, A. Mitrokotsa, and S. Vaudenay. On the Pseudorandom Function Assumption in (Secure) Distance-Bounding Protocols. In *International Conference on Cryptology and Information Security in Latin America Latincrypt 2012*, 2012.
7. S. Brands and D. Chaum. Distance-Bounding Protocols (Extended Abstract). In *EUROCRYPT*, pages 344–359, 1993.
8. L. Bussard. *Trust Establishment Protocols for Communicating Devices*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Institut Eurécom, Télécom Paris, 2004.
9. L. Bussard and W. Bagga. Distance-Bounding Proof of Knowledge Protocols to Avoid Terrorist Fraud Attacks. Technical Report RR-04-109, Institute EURECOM, May 2004.
10. L. Bussard and W. Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. In *Security and Privacy in the Age of Ubiquitous Computing, IFIP TC11 20th International Conference on Information Security (SEC 2005), May 30 - June 1, 2005, Chiba, Japan*, pages 223–238. Springer, 2005.

11. S. Čapkun, L. Buttyán, and J.-P. Hubaux. SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. In *ACM Workshop on Security of Ad Hoc and Sensor Networks - SASN*, pages 21 – 32, 2003.
12. Y. Chen and P. Q. Nguyen. BKZ 2.0: Simulation and better lattice security estimates. In *Proceedings of ASIACRYPT 2011*, 2011.
13. H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Statistics*, 23(4):493–507, 1952.
14. J. Clulow, G. P. Hancke, M. G. Kuhn, and T. Moore. So Near and Yet So Far: Distance-Bounding Attacks in Wireless Networks. In *Proceedings of the Third European Workshop on Security and Privacy in Ad hoc and Sensor Networks*, pages 83–97, 2006.
15. Y. Desmedt. Major Security Problems with the "Unforgeable" (Feige)-Fiat-Shamir Proofs of Identity and How to Overcome Them. pages 147–159, Paris, France, 15-17 March 1988. SEDEP.
16. S. Drimer and S. J. Murdoch. Keep your enemies close: distance bounding against smartcard relay attacks. In *Proceedings of 16th USENIX Security Symposium*, pages 7:1–7:16, Berkeley, CA, USA, 2007. USENIX Association.
17. A. Francillon, B. Danev, and S. Čapkun. Relay attacks on passive keyless entry and start systems in modern cars. Cryptology ePrint Archive, Report 2010/332, 2010. `http://eprint.iacr.org/`.
18. S. Gezici, Z. Tian, G. B. Biannakis, H. Kobayashi, A. F. Molisch, V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radius: a look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 22(4):70–84, 2005.
19. H. Gilbert, M. Robshaw, and H. Sibert. An Active Attack Against HB+ - A Provably Secure Lightweight Authentication Protocol. Technical report, IACR Cryptology ePrint Archive 237, 2005.
20. G. Hancke. Distance-bounding for RFID: Effectiveness of 'terrorist fraud' in the presence of bit errors. In *IEEE International Conference on RFID-Technology and Applications – IEEE RFID TA 2012*, IEEE Press, Nice, France, November 2012. IEEE.
21. G. P. Hancke and M. G. Kuhn. An RFID Distance Bounding Protocol. In *SECURECOMM*, pages 67–73, 2005.
22. W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.
23. A. Juels and S. A. Weis. Authenticating Pervasive Devices with Human Protocols. In *CRYPTO*, pages 293–308, 2005.
24. C. H. Kim and G. Avoine. RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks. In *Proceedings of the 8th International Conference on Cryptology and Networks Security (CANS 2009)*, volume 5888, pages 119–131, 2009.
25. C. H. Kim, G. Avoine, F. Koeune, F. Standaert, and O. Pereira. The Swiss-Knife RFID Distance Bounding Protocol. In *International Conference on Information Security and Cryptology – ICISC*, Lecture Notes in Computer Science. Springer-Verlag, December 2008.
26. M. Kuhn, H. Luecken, and N. O. Tippenhauer. UWB Impulse Radio Based Distance Bounding. In *Proceedings of the 7th Workshop on Positioning, Navigation and Communication 2010 (WPNC'10)*, 2010.
27. J.-Y. Lee and R. A. Scholtz. Ranging in a Dense Multipath Environment using an UWB Radio Link. *IEEE Journal on Selected Areas in Communications*, 20(9), 2002.
28. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen*, 261:515–534, 1982. 10.1007/BF01457454.

29. C. Meadows, P. Syverson, and L.Chang. Towards More Efficient Distance Bounding Protocols for Use in Sensor Networks. In *Proceedings of the International Conference on Security and Privacy in Communication Networks (SECURECOMM'06)*, pages 1–5, 2006.
30. J. Munilla and A. Peinado. Distance Bounding Protocols for RFID Enhanced by Using Void-challenges and Analysis in Noisy Channels. *Wireless Communications and Mobile Computing*, 8:1227–1232, November 2008.
31. J. Munilla and A. Peinado. Security Analysis of Tu and Piramuthu's Protocol. In *New Technologies, Mobility and Security – NTMS'08*, pages 1–5, Tangier, Morocco, November 2008. IEEE Computer Society.
32. P. Q. Nguyen and I. Shparlinski. The Insecurity of the Digital Signature Algorithm with Partially Known Nonces. *J. Cryptology*, 15(3):151–176, 2002.
33. V. Nikov and M. Vauclair. Yet Another Secure Distance-Bounding Protocol. In *Proceedings of the Conference on Security and Cryptography (SECRYPT 2008)*, pages 218–221, July 2008.
34. J. Reid, J. M. Gonzalez Nieto, T. Tang, and B. Senadji. Detecting Relay Attacks with Timing-based Protocols. In *ASIACCS '07: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security*, pages 204–213. ACM, 2007.
35. C. P. Schnorr. Block Reduced Lattice Bases and Successive Minima. *Combinatorics, Probability and Computing*, 3(04):507–522, 1994.
36. C. P. Schnorr and M. Euchner. Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. In *Math. Programming*, pages 181–191, 1993.
37. V. Shoup. NTL: A Library for Doing Number Theory. `http://shoup.net/ntl`.
38. D. Singelée and B. Preneel. Location Verification Using Secure Distance Bounding Protocols. In *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS'05)*, pages 834–840, 2005.
39. Y.-J. Tu and S. Piramuthu. RFID Distance Bounding Protocols. In *First International EURASIP Workshop on RFID Technology*, 2007.
40. S. Vaudenay. On Privacy Models for RFID. In *ASIACRYPT*, pages 68–87, 2007.

A Appendix

If $X_1, \ldots, X_n$ are independent Bernoulli random variables with $X_k \in \{0, 1\}$ and $\mathbb{P}[X_k = 1] = \mu$ for all k, then

$$\mathbb{P}\left[\sum_{k=1}^{n} X_k \geq u\right] = \sum_{k=0}^{u} \binom{n}{k} \mu^k (1-\mu)^{n-k}. \tag{4}$$

This probability can be bounded via *Hoeffding's inequality* [22], i.e.,:

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

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

Lemma 2 (Chernoff Bound [13]). *For independent Bernoulli random variables $X_1, \ldots, X_n$ with $\mathbb{P}[X_i = 1] = \mu > 1/2$, then the probability of simultaneous occurrence of more that $n/2$ of the events $\{X_k = 1\}$ has a lower bound, namely*

$$P \geq 1 - \exp(-2n(\mu - 1/2)^2).$$