Private and Secure Public-Key Distance Bounding

Application to NFC Payment

Serge Vaudenay

EPFL
CH-1015 Lausanne, Switzerland
http://lasec.epfl.ch

Abstract. Distance-Bounding is used to defeat relay attacks. For wireless payment systems, the payment terminal is not always online. So, the protocol must rely on a public key for the prover (payer). We propose a generic transformation of a (weakly secure) symmetric distance bounding protocol which has no post-verification into wide-strong-private and secure public-key distance bounding.

1 Introduction

Several wireless payment systems such as toll payment systems and NFC credit cards have recently been spread. These methods allow to pay small amounts without any action from the holder (no confirmation, no PIN code) other than approaching their device to the payment terminal.

In relay attacks, a man-in-the-middle $\mathcal{A}$ passively relays messages between two participants: a *prover* P and a *verifier* V [9,10]. The prover P is a credit card (of the payer) and the verifier V is a payment terminal (of the vendor). $\mathcal{A}$ can be run by two players: a malicious customer $\mathcal{A}_1$ mimicking a payment in a shop to buy some service to V, and a malicious neighbor $\mathcal{A}_2$ to the victim P. $\mathcal{A}_1$ and $\mathcal{A}_2$ relay messages between P and V. The payer may remain clueless.

So far, the most promising technique to defeat relay attacks is distance-bounding (DB) [5]. A DB protocol has several fast challenge/response rounds during which the verifier/vendor sends a challenge bit and expects to receive a response bit within a very short time from the prover/payer. The protocol fails if some response arrives too late or is incorrect. Due to the time of flight, if P is too far from V, his time to compute the response is already over when the challenge reaches him. Here are the traditional threat models for DB.

- Honest-prover security: *man-in-the-middle attacks* (MiM) (including *impersonation fraud* [1] and the so-called *mafia fraud* [8] including *relay attacks*).
- Malicious-prover security: *distance fraud* (DF) [5], in which a far-away malicious prover pretends that he is close; *distance hijacking* (DH) [7], in which the malicious prover relies on *honest* close-by participants; *collusion frauds* (CF) [3] (including the so-called *terrorist fraud* [8]), in which a malicious prover colludes with close-by participants (but without leaking credentials).

– *Privacy*, where we want that no man-in-the-middle adversary can learn the identity of the prover. Wide/narrow privacy refers to whether the adversary can see if a protocol succeeds on the verifier side. Strong/weak privacy refers to whether the adversary can corrupt provers and get their secret.

For payment systems, we cannot assume an online connection to a trusted server nor a shared secret between the payer and the vendor: we must have a public-key based protocol. We can further wonder which threat models are relevant. Clearly, the man-in-the-middle attacks are the main concern. Privacy is also important as payers want to remain anonymous to observers. For undeniability, a malicious payer shall not do a distance fraud then deny having made a payment on the basis that he was too far. Distance fraud shall also be prevented to be able to catch red handed people who pay with a stolen credit card.

Table 1. Existing Public-Key Distance Bounding Protocols

protocol	MiM	DF	DH	CF	Privacy	Strong privacy
Brands-Chaum [5]	secure	secure	insecure	insecure	insecure	insecure
DBPK-Log [6]		insecure		insecure	insecure	insecure
HPO [13]	secure	secure		insecure	secure	insecure
GOR [11]	secure	secure	insecure	insecure	insecure	insecure
ProProx [18]	secure	secure	secure	secure	insecure	insecure
privDB	secure	secure	secure	insecure	secure	secure

(Missing entries correspond to absence of proof in either direction.)

Not many public-key DB protocols exist: the Brands-Chaum protocol [5], the DBPK-Log protocol [6], the protocol by Hermans, Peeters, and Onete [13] (herein called the HPO protocol), its recent extension by Gambs, Onete, and Robert [11] (the GOR protocol, herein)[1], and ProProx [18] (see Table 1). None except ProProx resist to collusion frauds. The Brands-Chaum protocol does not resist to distance hijacking [7]. In [2], Bay *et al.* have broken DBPK-Log. Neither the Brands-Chaum protocol nor ProProx protect privacy but the HPO and GOR protocols were designed for this. However, HPO does not offer strong privacy and privacy in GOR can be broken, as this will be proven in a subsequent paper.

In this paper, we transform a symmetric DB protocol symDB with no post-verification into a public-key DB protocol privDB. Assuming some weak form of DF, MiM, and DH security for symDB, we prove that privDB is DF, MiM, DH secure, and strong-private. It is the first to be provably DH-secure and the first to be strong private. We propose a suitable symDB protocol called OTDB.

Acknowledgements. The author would like to thank Erik-Oliver Blass, Tom Chothia, and Yvo Desmedt for valuable remarks. This work is part of the ICT COST Action IC1403 (*Cryptacus*).

[1] The GOR protocol is a bit different from others as it provides *anonymous authentication*. The verifier does not identify the prover in the protocol.

2 Definitions

We recall and adapt the framework of [4,18]. We assume a multiparty setting in which participants have a *location* and information travels at the speed of light. Participants receive inputs and produce outputs. Honest participants run their purported algorithm. Malicious participants may run an arbitrary probabilistic polynomial-time (PPT) algorithm. The definition below is adapted from [4,18] to accommodate identification protocols and also to bridge public-key and symmetric distance bounding.

Definition 1. *A* distance-bounding protocol (DB) *consists of what follows. 1. B: a distance bound. 2.* K_P *and* K_V*: two PPT* key generation *algorithms depending on a security parameter* λ*. For a* public-key DB identification protocol*, "setting up the keys" for P and V means running* $K_P \to (\mathsf{sk}_P, \mathsf{pk}_P)$ *and* $K_V \to (\mathsf{sk}_V, \mathsf{pk}_V)$*. For* Symmetric DB*, provers/verifiers are paired and "setting up the keys" for a pair* (P,V) *means running* $K_P \to \mathsf{sk}_P$ *then setting* $\mathsf{sk}_V = \mathsf{sk}_P$ *and* $\mathsf{pk}_P = \mathsf{pk}_V = \perp$*. 3.* $(P(\mathsf{sk}_P, \mathsf{pk}_V), V(\mathsf{sk}_V))$*: a two-party PPT protocol where* $P(\mathsf{sk}_P, \mathsf{pk}_V)$ *is the* proving algorithm *and* $V(\mathsf{sk}_V)$ *is the* verifying algorithm*. At the end of the protocol,* $V(\mathsf{sk}_V)$ *has a private output and sends a final message* Out_V*. He accepts (*$\mathsf{Out}_V = 1$*) or rejects (*$\mathsf{Out}_V = 0$*).*

The protocol must be complete*. I.e., such that "setting up the keys" for* (P,V) *then making* $P(\mathsf{sk}_P, \mathsf{pk}_V)$ *and* $V(\mathsf{sk}_V)$ *interact together, at locations within a distance up to B always makes* $V(\mathsf{sk}_V)$ *accept (*$\mathsf{Out}_V = 1$*) and output* pk_P*.*

Moving to noisy settings [16] follows standard techniques which are omitted herein. Verifiers are assumed to be able to validate pk_P (e.g., by means of a PKI). In what follows, $\mathsf{Validate}(\mathsf{pk}_P)$ denotes this operation.

Security of DB. Like in [4,18], all security notions are formalized by a game with three types of participants: provers, verifiers, and actors. Each participant can have several instances at different location or time. Without loss of generality, actors are malicious. The purported algorithm is P for provers and V for verifiers. There is a distinguished instance of the verifier denoted by $\mathcal{V}$. Instances of participants within a distance to $\mathcal{V}$ up to B are called *close-by*. Others are called *far-away*. We say that the adversary *wins* if $\mathcal{V}$ accepts. In security models, we only consider without loss of generality (several instances of) one verifier who is honest. In Def. 2–3, we consider without loss of generality (several instances of) one prover with an identity corresponding to the key pk_P.

Definition 2 ([18]). *We consider the following honest-prover security notion. At the beginning of the game, we set up the keys (following Def. 1) and give* pk_V *as input to all participants,* sk_P *as input to the prover instances, and* pk_P *as input to all malicious participants. The prover is honest. The DB protocol is* MiM-secure *(man-in-the-middle) if for all such settings in which there is no close-by prover, the probability that* $\mathcal{V}$ *accepts and outputs* pk_P *is negligible.*[2]

The DB protocol is one-time MiM-secure (OT-MiM) *if the above is satisfied in settings where there is a single verifier instance and a single prover instance.*

[2] The key generation algorithms accepts as input a security parameter λ which is omitted for simplicity reasons. Hence, $\Pr[\mathcal{V}\ \mathsf{accepts}]$ is a function of λ. We say that $f(\lambda)$ is *negligible* if for every integer d we have $f(\lambda) = O(\lambda^{-d})$ for $\lambda \to +\infty$.

Definition 3 ([18]). *We consider the following malicious-prover security notion. At the beginning of the game, we use an arbitrary PPT algorithm $K(\mathsf{pk}_V)$ instead of K_P in the key setup. The DB protocol is* DF-secure *(distance fraud) if for all such settings where there is no close-by participant except $\mathcal{V}$, the probability that $\mathcal{V}$ accepts and outputs pk_P is negligible.*

Note that the key of the malicious prover is set up maliciously (even depending on pk_V) using an algorithm K which can differ from K_P.

Privacy. The most general and prominent model for privacy is the simulation-based privacy notion in [17] which was enriched in [15]. Hermans et al. [14] presented a simpler privacy model which we call the HPVP model.

Definition 4 (HPVP Privacy [14]). *We consider an adversary playing with the following oracles: 1.* $\mathsf{Create} \to (i, \mathsf{pk}_P)$ *runs K_P and sets pk_P as a valid key for a new prover whose number is i; 2.* $\mathsf{Corrupt}(i) \to \mathsf{state}$ *returns the current state (in permanent memory) of the ith prover; 3.* $\mathsf{Draw}(i, j) \to \mathsf{vtag}$ *draws either the ith prover (if in the left game) or the jth prover (if in the right game) and returns a pseudonym* vtag *(if the prover is already drawn, $\perp$ is returned); 4.* $\mathsf{Free}(\mathsf{vtag})$ *releases* vtag *so that it can be drawn again; 5.* $\mathsf{SendP}(\mathsf{vtag}, m) \to m'$ *sends a message m to a drawn tag* vtag *and gets a response m' (if* vtag *was released, $\perp$ is returned instead); 6.* $\mathsf{Launch} \to k$ *runs a new verifier whose number is k; 7.* $\mathsf{SendV}(k, m) \to m'$ *sends a message m to the kth verifier and gets a response m'; 8.* $\mathsf{Result}(k) \to \mathsf{Out}_V$ *gives the final result (whether the protocol succeeded or not) of the protocol on the kth verifier side. In the privacy game, the adversary interacts with these oracles and guesses if it is left or right. The game is formalized as follows: 1. run $K_V \to (\mathsf{sk}_V, \mathsf{pk}_V)$ and initialize all verifiers with sk_V and all provers and $\mathcal{A}$ with pk_V; 2. pick $b \in \{0,1\}$; 3. let $\mathcal{A}$ interact with the oracles (in the left game for $b = 0$ or the right game for $b = 1$) and make a guess β; 4. $\mathcal{A}$ wins if $\beta = b$. We have privacy if for every PPT adversary $\mathcal{A}$,* $\Pr[\mathcal{A}\ \mathsf{wins}] - \frac{1}{2}$ *is negligible. For* narrow *privacy, the adversary does not use the* Result *oracle. For* weak *privacy, he does not use the* Corrupt *oracle. Otherwise, the adversary is* wide*, respectively* strong.

Distance Hijacking. In distance hijacking [7], the prover is malicious, running an algorithm $\mathcal{A}$ and we add a honest prover $P(\mathsf{sk}_{P'}, \mathsf{pk}_V)$ with another identity P' associated to $\mathsf{pk}_{P'}$. The malicious prover runs $\mathcal{A}(\mathsf{sk}_P, \mathsf{pk}_P, \mathsf{pk}_{P'}, \mathsf{pk}_V)$. We formalize distance hijacking for DB protocols consisting of a regular (i.e., time-insensitive) initialization phase, a time-critical challenge phase, and a regular verification phase. $\mathcal{A}$ is playing a man-in-the-middle between $P(\mathsf{sk}_{P'}, \mathsf{pk}_V)$ and $V(\mathsf{sk}_V)$ except during the challenge phase when he remains passive. (See Fig. 1.)

Definition 5. *A DB protocol (B, K_P, K_V, P, V) is* DH-secure *if for all PPT algorithms K and $\mathcal{A}$, the following game makes $\mathcal{V}$ output pk_P with negligible probability:*

1: *for public-key DB: $K_P \to (\mathsf{sk}_{P'}, \mathsf{pk}_{P'})$, $K_V \to (\mathsf{sk}_V, \mathsf{pk}_V)$, $K(\mathsf{pk}_{P'}, \mathsf{pk}_V) \to (\mathsf{sk}_P, \mathsf{pk}_P)$; if $\mathsf{pk}_P = \mathsf{pk}_{P'}$, the game aborts*
for symmetric DB: $K_P \to \mathsf{sk}_{P'}$, $K \to \mathsf{sk}_P$, set $\mathsf{sk}_V = \mathsf{sk}_P$, $\mathsf{pk}_P = \mathsf{pk}_{P'} = \mathsf{pk}_V = \perp$;
2: *let $\mathcal{A}$ run $\mathcal{A}(\mathsf{sk}_P, \mathsf{pk}_P, \mathsf{pk}_{P'}, \mathsf{pk}_V)$, let $\mathcal{V}$, $V_1, V_2, \ldots$ run $V(\mathsf{sk}_V)$, and let P', $P'_1, P'_2, \ldots$ run $P(\mathsf{sk}_{P'}, \mathsf{pk}_V)$*

3: *let $\mathcal{A}$ interact with $P', P'_1, P'_2 \ldots$ and $\mathcal{V}, V_1, V_2, \ldots$ concurrently until the initialization phase ends for $\mathcal{V}$*
4: *let P' and $\mathcal{V}$ continue interacting with each other until the challenge phase ends for $\mathcal{V}$; $\mathcal{A}$ receives the exchanged messages but remains passive*
5: *let $\mathcal{A}$ continue interacting with $P', P'_1, P'_2 \ldots$ and $\mathcal{V}, V_1, V_2, \ldots$ concurrently during the verification phase*

A DB protocol is one-time DH-secure (OT-DH) *if the above holds when there are no P'_i and V_i.*

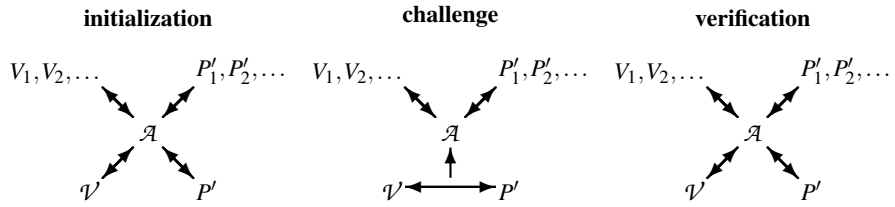

Fig. 1. Distance Hijacking

3 From Symmetric to Asymmetric Distance Bounding

3.1 The OTDB Protocol

We propose a one-time DB protocol OTDB based on the Hancke-Kuhn protocol [12]. It is represented on Fig. 2. We use a $2n$-bit secret s. It is XORed to a random mask m selected by the verifier. The answer to a challenge in iteration i is just the bit of $s \oplus m$ at position $2i-1$ or $2i$, depending on the challenge.

We define a sub-category of simple DB protocols.

Definition 6. *A symmetric DB protocol (B, K, P, V) follows the* canonical structure *if there exist 5 PPT algorithms P_{init}, P_{chall}, V_{init}, V_{chall}, V_{ver} such that $P(s)$ and $V(s)$ are defined as follows:*

1. *$P(s)$ and $V(s)$ run the initialization phase by running P_{init} and V_{init}. These algorithms do not use s. They produce a final state σ_P and σ_V.*
2. *$P(s)$ and $V(s)$ run the challenge phase by running $P_{\mathsf{chall}}(s, \sigma_P)$ and $V_{\mathsf{chall}}(\sigma_V)$, where V_{chall} does not depend on s and produces a final state σ'_V.*
3. *$V(s)$ computes $\mathsf{Out}_V = V_{\mathsf{ver}}(s, \sigma'_V)$.*

The canonical point is that there is no interactive verification and the secret is used by P only in the challenge phase and by V only in the final verification.

Theorem 7. OTDB *follows the canonical structure. It is DF-secure, OT-MiM-secure, and OT-DH-secure.*

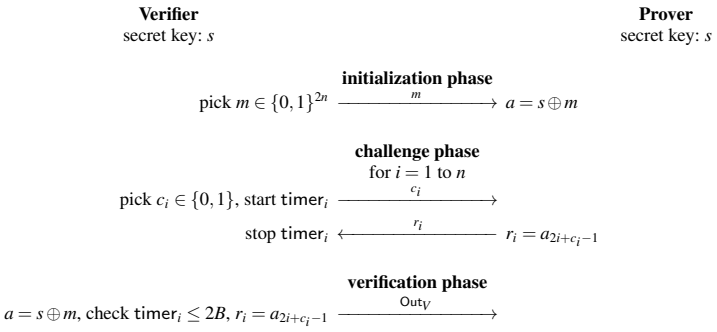

Fig. 2. The OTDB Protocol.

Proof. The canonical structure of OTDB is clear.

For DF-security, we observe that whatever the adversary is doing, the distribution of a on the verifier side is uniform in $\{0,1\}^{2n}$. Since there is no close-by participant, a response can be received on time only if it was sent before the challenge was known. If $a_{2i-1} = a_{2i}$, this can be done with probability 1. Otherwise, this can only be done with probability $\frac{1}{2}$. So, the optimal probability that all responses are correct is $\sum_{w=0}^{n} \binom{n}{w} 2^{-n-w} = \left(\frac{3}{4}\right)^n$ which is negligible.

For OT-MiM-security, we consider a distant $\mathcal{V} = V(s)$ and $P(s)$ with several actors. By playing with $P(s)$, the adversary can deduce for each i either s_{2i-1} or s_{2i} but not both. To answer to $\mathcal{V}$, he must know precisely which of these two bits is needed but when he learns it, it is too late to play with $P(s)$ to get it. So, the probability to pass one round is limited to $\frac{3}{4}$. So, the probability of success is also $\left(\frac{3}{4}\right)^n$, which is negligible.

For OT-DH-security, we consider P' who is set up with a random s' and V who is maliciously set up with an independent s. In the initialization part (which can be corrupted), we let m be the value sent by $\mathcal{V}$ and m' be the value received by P'. When they start the challenge phase, $\mathcal{V}$ uses $a = s \oplus m$ and P' uses $a' = s' \oplus m'$, where m' only depends on m and s. So, a' is uniformly distributed and independent from a. The challenge part between P' and V cannot be corrupted, by definition of the OT-DH-security. Hence, $\mathcal{V}$ accepts with probability 2^{-n}, which is negligible. □

As concrete parameters, we can use $n = 49$ for a 2^{-20} online security.

3.2 The privDB Protocol

We adapt the RFID protocol from [15,17] for DB. We assume that K_V generates a key pair for a public-key cryptosystem Enc/Dec and that K_P generates a key pair for a digital signature scheme Sign/Verify. The protocol runs as follows (see Fig. 3): 1. V sends a nonce N to P; 2. P picks a random s and sends $\mathsf{Enc}_{\mathsf{pk}_V}(s \| \mathsf{pk}_P \| \mathsf{Sign}_{\mathsf{sk}_P}(N))$ to V; 3. V decrypts, verifies the signature on N, and validates pk_P (if this step fails, V sends

$\mathsf{Out}_V = 0$ and aborts);[3] 4. P and V run a symmetric DB symDB based on the secret s (if this step fails, V sends $\mathsf{Out}_V = 0$ and aborts); 5. the private output of V is set to pk_P and the public one is set to $\mathsf{Out}_V = 1$. Compared to HPO [13], the encrypted channel can also be used to transmit a certificate in a private way.

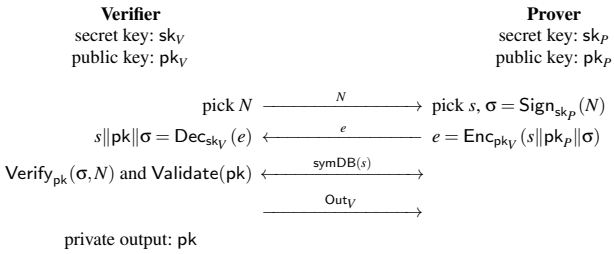

Fig. 3. privDB: Strong Private Public-Key DB from Symmetric DB.

Theorem 8. *If* symDB *is DF-secure then* privDB *is DF-secure.*

The reduction is quite trivial.

Definition 9. *We say the signature scheme is* Known-Key-UF-1CMA-secure (KK-UF-1CMA) *if for any PPT algorithm $\mathcal{A}$, the probability to win the following game is negligible: generate a key pair $(\mathsf{sk}_P, \mathsf{pk}_P)$ and pick a challenge N'; set the chosen message $N = \mathcal{A}(\mathsf{sk}_P, \mathsf{pk}_P, N')$ and sign it by $\sigma = \mathsf{Sign}_{\mathsf{sk}_P}(N)$. $\mathcal{A}$ wins if $N \neq N'$ and $\mathsf{Verify}_{\mathsf{pk}_P}(\sigma, N')$ accepts. We say the signature scheme is* simple-UF-1CMA-secure (S-UF-1CMA) *if the same holds but for $N = \mathcal{A}(\mathsf{pk}_P, N')$.*

Clearly, the standard UF-CMA security implies S-UF-1CMA security.

Theorem 10. *If* symDB *is OT-MiM-secure, the signature scheme is S-UF-1CMA-secure, the cryptosystem resists chosen-ciphertext attacks (IND-CCA secure), then* privDB *is MiM-secure.*

Proof. We let Γ_0 denote the MiM security game. In what follows, Γ_i is a game and p_i denotes the probability that Γ_i succeeds. We want to show that p_0 is negligible. We first reduce Γ_0 to a game Γ_1 in which no two verifiers select the same nonce and no two provers select the same s (so, their e are unique as well). Clearly, $p_1 - p_0$ is negligible. In Γ_2, we simulate every verifier V who is given a e produced by a prover P. We let $N, s, \mathsf{pk}_P, \sigma$ be the values from the viewpoint of P. In the simulation, if V produced N himself, the decryption and verifications are skipped and V proceeds with $\mathsf{symDB}(s)$ directly. We say that P and V are matching instances. Otherwise, only the decryption is skipped and V proceeds with s, pk_P, σ. Clearly, $p_2 = p_1$. In Γ_2, no e produced by

[3] In a previous version, N was part of the plaintext. At the conference, Erik-Oliver Blass suggested to remove it. This required to adapt the proofs.

any P needs to be decrypted. In Γ_3, we sequentially replace every $e = \mathsf{Enc}_{\mathsf{pk}_V}(s, \mathsf{pk}_P, \sigma)$ by some $e = \mathsf{Enc}_{\mathsf{pk}_V}(\mathsf{rand})$ and use the IND-CCA security to deduce that $p_3 - p_2$ is negligible. In Γ_3, no information about s or σ leaks from e.

To go from Γ_3 to Γ_4, we eliminate all signatures by repeating the following transformation: let $\sigma = \mathsf{Sign}_{\mathsf{sk}_P}(N)$ be the very first signature computation. We note that σ can only be used later by a $\mathsf{Verify}_{\mathsf{pk}_P}(\sigma, N')$ computation, for $N \neq N'$. If it is not immediately followed by a this verification, we postpone the signature computation to the very first moment when σ is used. Clearly, this does not affect the probability of success. If instead it is followed by $\mathsf{Verify}_{\mathsf{pk}_P}(\sigma, N')$, we replace $\mathsf{Verify}_{\mathsf{pk}_P}(\sigma, N')$ by 0 (rejection). (So, the next transformation continues to postpone the signature.) By replacing the generation of a random N' by a S-UF-1CMA challenge (and aborting if it is not the right N'), we use the S-UF-1CMA security to deduce that the probability of success is negligibly affected. After repeating this process, we eliminate the signing operations. We obtain a game Γ_4 in which a verifier instance has up to one matching prover instance and each prover instance has up to one matching verifier instance.

In Γ_4, either $\mathcal{V}$ uses a forged signature (but we eliminate this case with the S-UF-1CMA security), or $\mathcal{V}$ has a unique matching P and they both run $\mathsf{symDB}(s)$ on a random s. By simulating everything else but this instance of symDB, we obtain the OT-MiM-security game of symDB. Due to the OT-MiM-security of symDB, we conclude that p_4 must be negligible. □

Theorem 11. *If* symDB *follows the canonical structure and is OT-MiM and OT-DH secure; the signature scheme is S-UF-1CMA-secure; the cryptosystem resists chosen-ciphertext attacks (IND-CCA secure); then* privDB *is DH-secure.*

Proof. We let Γ_0 denote the DH security game. In what follows, Γ_i is a game and p_i denotes the probability that Γ_i succeeds. We want to show that p_0 is negligible. Since symDB has no interactive verification, Γ_0 consists of two phases after the key set up: the initialization phase and the challenge phase. The last phase matches the challenge phase of symDB between $\mathcal{V}$ and P' alone. For Γ_0 to succeed, $\mathcal{V}$ must identify P. So, we assume that $\mathcal{V}$ receives pk_P during the initialization. The main point is to realize that $\mathcal{V}$ and P' must then start with two independent keys s and s' with s' uniform. We conclude using the OT-DH-security of symDB.

We do the same reduction as in the proof of Th. 10 to the games $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ (with P' replacing P). Since $\mathcal{V}$ receives pk_P, he cannot match P'. Let s' be the randomly distributed value selected by P'. We first treat the case where there is no V_i matching P'. So, s' is never used before the challenge phase due to the canonical structure of symDB. Therefore, $\mathcal{V}$ is set up with some s which is independent from s'. Hence, we are in the situation of the OT-DH game of symDB. By using the OT-DH-security of symDB, p_3 is negligible.

Let now assume that one verifier instance matches P'. We know that it is unique and we assume that it is V_1 without loss of generality. If V_1 does not compute his Out_{V_1} before the challenge phase of the game, none of his messages depend on s' due to the canonical structure of symDB, so we can proceed as in the previous case.

Now, if V_1 sends out his Out_{V_1} before the challenge phase of the game, we define a new game Γ_5 in which Out_{V_1} is replaced by 0. In Γ_5, we can conclude as in the previous

case that p_5 is negligible. So, what is left to be shown is that $p_5 - p_4$ is negligible, or equivalently that $\mathsf{Out}_{V_1} = 1$ with negligible probability in Γ_4. For that, we observe that P' is only running the initialization of symDB (which does not depend on s' by assumption on symDB) until Out_{V_1} is released. Since V_1 is set up with a random s' and that no other algorithm depends on s' in this phase, we are in an impersonation attack case. We conclude using the OT-MiM security of symDB. □

Theorem 12. *If the signature scheme is KK-UF-1CMA-secure and the cryptosystem resists chosen-ciphertext attacks (IND-CCA secure),* privDB *is wide-strong private in the HPVP model.*[4]

Proof. We let Γ_0 denote the wide-strong HPVP privacy game. In what follows, Γ_i is a game and p_i denotes the probability that Γ_i succeeds. We want to show that $p_0 - \frac{1}{2}$ is negligible. We do the same reduction as in the proof of Th. 10 to the games $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ (but with KK-UF-1CMA security) and obtain that $p_4 - p_0$ is negligible. We observe that in Γ_4, the pk_P and σ by a drawn prover is never used. The public key is only important during Corrupt queries, but this does not apply on drawn provers in the HPVP model. So, drawn provers use no proper identity in Γ_4. It does not matter which prover is drawn (the left or the right), the simulation of the prover is the same. So the probability of correctly winning $\beta = b$ must be exactly $p_4 = \frac{1}{2}$. □

References

1. G. Avoine, A. Tchamkerten. An Efficient Distance Bounding RFID Authentication Protocol: Balancing False-Acceptance Rate and Memory Requirement. In *Information Security ISC'09*, Pisa, Italy, Lecture Notes in Computer Science 5735, pp. 250–261, Springer-Verlag, 2009.
2. A. Bay, I. Boureanu, A. Mitrokotsa, I. Spulber, S. Vaudenay. The Bussard-Bagga and Other Distance-Bounding Protocols under Attacks. In *INSCRYPT'12*, Beijing, China, Lecture Notes in Computer Science 7763, pp. 371–391, Springer-Verlag, 2012.
3. I. Boureanu, A. Mitrokotsa, S. Vaudenay. Towards Secure Distance Bounding. In *Fast Software Encryption'13*, Singapore, Lecture Notes in Computer Science 8424, pp. 55–67, Springer-Verlag, 2013.
4. I. Boureanu, S. Vaudenay. Optimal Proximity Proofs. IACR Eprint 2014/693 report, 2014. To appear in the proceedings of INSRYPT'14.
5. S. Brands, D. Chaum. Distance-Bounding Protocols (Extended Abstract). In *Advances in Cryptology EUROCRYPT'93*, Lofthus, Norway, Lecture Notes in Computer Science 765, pp. 344–359, Springer-Verlag, 1994.
6. L. Bussard, W. Bagga. Distance-Bounding Proof of Knowledge to Avoid Real-Time Attacks. In *IFIP TC11 International Conference on Information Security SEC'05*, Chiba, Japan, pp. 223–238, Springer, 2005.
7. C.J. F. Cremers, K.B. Rasmussen, B. Schmidt, S. Čapkun. Distance Hijacking Attacks on Distance Bounding Protocols. In *IEEE Symposium on Security and Privacy S&P'12*, San Francisco, California, USA, pp. 113–127, IEEE Computer Society, 2012.

[4] KK-UF-1CMA was added in the final version of this paper after having removed N from the plaintext. It was necessary due to the adversary getting sk_P by corruption.

8. Y. Desmedt. Major Security Problems with the "Unforgeable" (Feige-)Fiat-Shamir Proofs of Identity and How to Overcome Them. In *Congress on Computer and Communication Security and Protection Securicom'88*, Paris, France, pp. 147–159, SEDEP Paris France, 1988.
9. A. Francillon, B. Danev, S. Čapkun. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. In *Network and Distributed System Security Symposium (NDSS'11)*, San Diego, CA, USA, The Internet Society, 2011.
10. L. Francis, G. Hancke, K. Mayes, K. Markantonakis. On the Security Issues of NFC Enabled Mobile Phones. *International Journal of Internet Technology and Secured Transactions (IJITST)*, vol. 2, pp. 336–356, 2010.
11. S. Gambs, C. Onete, J.-M. Robert. Prover Anonymous and Deniable Distance-Bounding Authentication. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS'14)*, Kyoto, Japan, pp. 501–506, ACM Press, 2014.
12. G.P. Hancke, M.G. Kuhn. An RFID Distance Bounding Protocol. In *Conference on Security and Privacy for Emerging Areas in Communications Networks SecureComm'05*, Athens, Greece, pp. 67–73, IEEE, 2005.
13. J. Hermans, R. Peeters, C. Onete. Efficient, Secure, Private Distance Bounding without Key Updates. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks WISEC'13*, Budapest, Hungary, pp. 195–206, ACM, 2013.
14. J Hermans, A. Pashalidis, F. Vercauteren, B. Preneel. A New RFID Privacy Model. In *Computer Security - ESORICS'11*, Leuven, Belgium, Lecture Notes in Computer Science 6879, pp. 568–587, Springer-Verlag, 2011.
15. K. Ouafi, S. Vaudenay. Strong Privacy for RFID Systems from Plaintext-Aware Encryption. In *Cryptology and Network Security, 8th International Conference CANS'12*, Darmstadt, Germany, Lecture Notes in Computer Science 7712, pp. 247–262, Springer-Verlag, 2012.
16. D. Singelée, B. Preneel. Distance Bounding in Noisy Environments. In *Security and Privacy in Ad-hoc and Sensor Networks ESAS 2007*, Cambridge, UK, Lecture Notes in Computer Science 4572, pp. 101–115, Springer-Verlag, 2007.
17. S. Vaudenay. On Privacy Models for RFID. In *Advances in Cryptology ASIACRYPT'07*, Kuching, Malaysia, Lecture Notes in Computer Science 4833, pp. 68–87, Springer-Verlag, 2007.
18. S. Vaudenay. Proof of Proximity of Knowledge. IACR Eprint 2014/695 report, 2014.