On the Need for Secure Distance-Bounding

Ioana Boureanu[1], Aikaterini Mitrokotsa[2], and Serge Vaudenay[1]

[1] Ecole Polytechnique Fédérale de Lausanne (EPFL),
[2] University of Applied Sciences of Western Switzerland (HES-SO)

Abstract. Distance-bounding is a practical solution to be used in security-sensitive contexts, mainly to prevent relay attacks. But subtle security shortcomings related to the PRF (pseudorandom function) assumption and ingenious attack techniques based on observing verifiers' outputs have recently been put forward. In this extended abstract, we survey some of these security concerns and attempt to incorporate the lessons taught by these new developments in ideas of distance-bounding protocol design.

1 Introduction

In [4], Brands and Chaum introduced the notion of distance-bounding (DB) protocols. The aim is to have a prover demonstrate his proximity to a verifier, and authenticate himself to this verifier. The proof of proximity can be an efficient deterrent against relay attacks [9]. DB protocols [11,12,14,15] generally consist of an *initialisation phase* (where the parties establish some short-term secret) and a *distance-bounding* (DB) phase. This latter phase is time-critical. It imposes very fast computation, typically of less than a single clock cycle per round, and the verifier measures the time-of-flight of the messages exchanged. This is how the verifier ascertains a distance-bound between him and the prover.

Most distance-bounding protocols follow the following design pattern, where f is a PRF and F is a response-function for the DB phase.

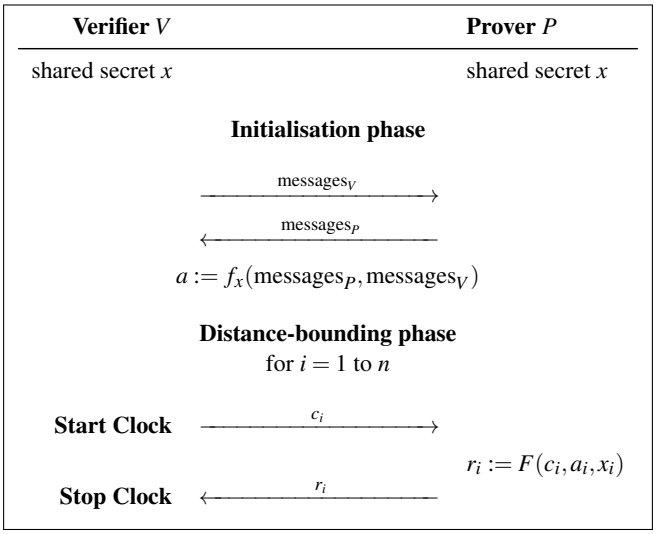


Fig. 1. The General Structure of Numerous DB Protocols (see [15,14])

In the literature covering such protocols, the threat-model comprises three well-established types of attacks. The first is *distance-fraud* (DF), in which a prover tries to convince the verifier that he is closer than he really is. In the second type, *mafia-fraud* (MF), an adversary communicates with both

a prover and a verifier which are far apart, and the adversary tries to convince the verifier that the prover would be close enough to be granted privileges. Finally, in a *terrorist-fraud* (TF), an adversary is getting the necessary and sufficient help from a coerced, far-away prover in order to pass the protocol only during this corrupted run, but not in a later, coercion-free session. Generalisations of these frauds have also been imagined. In [8], Cremers *et al.* describe distance-hijacking as a mixture between distance-fraud and terrorist-fraud: one dishonest, far-away prover exploits several honest provers to gain privileges. Impersonation-fraud is presented in [10]; as its name suggests, one dishonest prover tries to impersonate an honest one.

In this extended abstract, we first survey the most recent such attacks that have been proposed. Then, we attempt to incorporate the lessons taught by these new developments in ideas of *secure* distance-bounding protocol design.

2 DB: Instability of Security Results

2.1 PRF-based Unfortunate Arguments

Some security models [10] contain incorrect proofs/arguments: i.e., they replace a PRF by a random function at a place where the adversary has access to the PRF key or at a place where the PRF key is simultaneously used at other places in the protocol. If we observe Fig. 1, we can see that a dishonest prover holds the key x of the PRF instance, which is also used inside the call to the response-function F. So, in a distance-fraud resistance proof, we cannot simply call the PRF assumption when speaking about the "leakage" produced via using f.

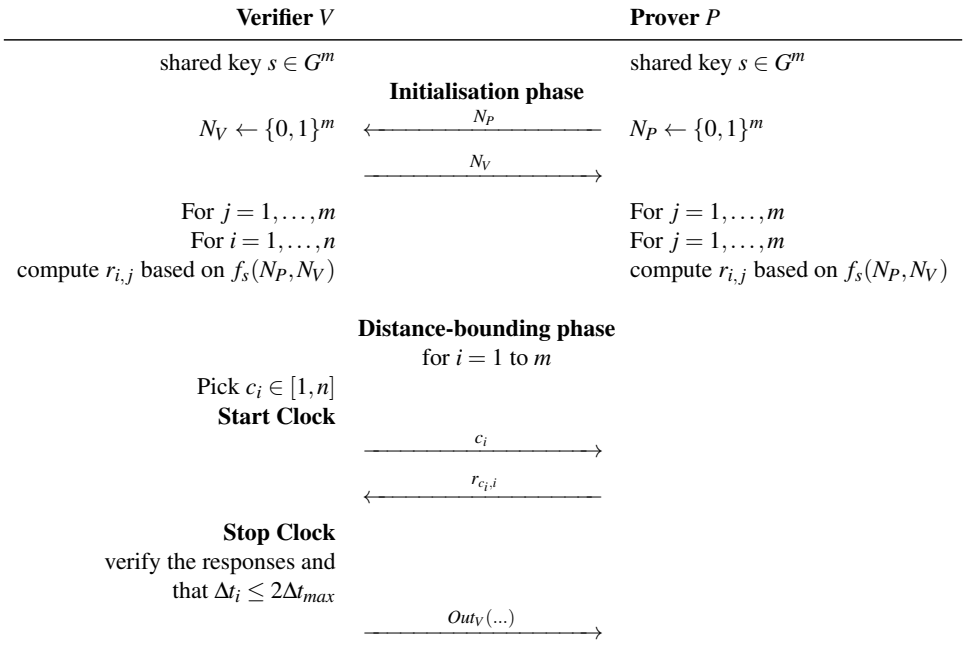


Fig. 2. The TDB [1] Protocol

Indeed, in [3], the authors have recently shown that if pseudorandom functions existed, then trapdoor PRFs can be constructed such that if used in instances of numerous DB protocols, they would lead to

DF attacks and generalised MF attacks. We leave the reader to consult [3] for the technicalities of the general construction of such programmed PRFs. In here, we only recall how one such PRF can bring DF-insecurity if used inside an instance of the TDB [1] protocol.

In Figure 2, we first recall the TDB protocol:

Now, consider one suggested instance of the TDB protocol, i.e., one where $n = k = 3$, $G = F_2$ and the $3 \times m$ response-matrix is of the form: $\mathcal{R}_1 = \begin{pmatrix} r_{1,1} & \cdots & r_{1,m} \\ r_{2,1} & \ddots & r_{2,m} \\ s_1 \oplus r_{1,1} \oplus r_{2,1} & \cdots & s_m \oplus r_{1,m} \oplus r_{2,m} \end{pmatrix}$

Then, assuming that PRFs exist, [3] shows that the PRF assumption on f is not enough to protect against DF in the above TDB instance. Namely, if PRFs exist, then let g be a PRF from $\{0,1\}^{2m}$ to itself. Take the PRF f as follows:

$$f_s(N_P, N_V) = \begin{cases} s \| s, & \text{if } N_P = s \\ g_s(N_P, N_V), & \text{otherwise.} \end{cases}$$

According to the general construction in [3], if g is a PRF, then this programmed f is a PRF as well. Then, let P^* be a dishonest prover who chooses N_P=s. In this case, the TDB instance above implies that the $\mathcal{R}_1$ response-matrix has all its rows equal to s. Then, for all challenges c_i, the response will be the i-th bit of the secret key s, hence P^* can win in a DF by sending the responses before the challenges even arrive at him.

With similar constructions (all based on their general design of trapdoor PRFs), [3] shows a series of attacks on numerous DB protocols and implies that possibly other similar DB protocols could be defeated similarly. A summary of the results in [3] is stated in Table 1.

protocol	distance fraud	man-in-the-middle attack
TDB Avoine-Lauradoux-Martin [ACM WiSec 2011]	√	√
Dürholz-Fischlin-Kasper-Onete [ISC 2011]	√	–
Hancke-Kuhn [Securecomm 2005]	√	–
Avoine-Tchamkerten [ISC 2009]	√	–
Reid-Nieto-Tang-Senadji [ASIACCS 2007]	√	√
Swiss-Knife Kim-Avoine-Koeune-Standaert-Pereira [ICISC 2008]	–	√

Table 1. A Summary of the PRF-based DB Attacks in [3]

2.2 PKC is not the solution for DB!

In some sense, in [2], the authors have recently shown that techniques from public key cryptography (PKC), i.e., commitments, proofs of knowledge, etc, are not the key to strengthen DB. More concretely, the authors have shown that the Bussard-Bagga protocols [6,7] suffer from TF attacks, even if their PKC techniques were supposed to prevent that.

In Figure 3, we first recall the generic Bussard-Bagga protocol, denoted DBPK. In their protocol, the prover P has a secret key x and a published certificate on its public key $y = \Gamma(x)$. 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 under a symmetric key encryption scheme $\mathcal{E} : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}^m$. After encrypting x and computing e, the prover P uses a bit commitment scheme to commit to each bit of k and e using randomnesses v and v' respectively.

In each distance-bounding round, the verifier V selects a random bit as the challenge c_i and the prover responds with a response r_i such that

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

In the commitments' opening phase, the prover P opens some commitments on the bits of k and e corresponding to his answers in the distance-bounding phase. This is denoted in Figure 3 through sending the value γ_i which stands for the respective randomness used at the commitment phase. 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.

In the Proof of knowledge 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.

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 $y = g^x \bmod p$.[3]

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 $u = 1$ [7] or u random and publicly revealed [7,6]. 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 [6] 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 $w_2 = y^c g^{s_1} \bmod p$.

Terrorist Fraud and Distance Fraud against DBPK-Log We now summarise how, in [2], the authors show that the above public-key techniques are ineffective in defeating 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. 4.

[3] In [6,7,5], h is not necessarily a generator and $x \in \mathbf{Z}_{p-1} \setminus \{q\}$ with $q = \frac{p-1}{2}$.

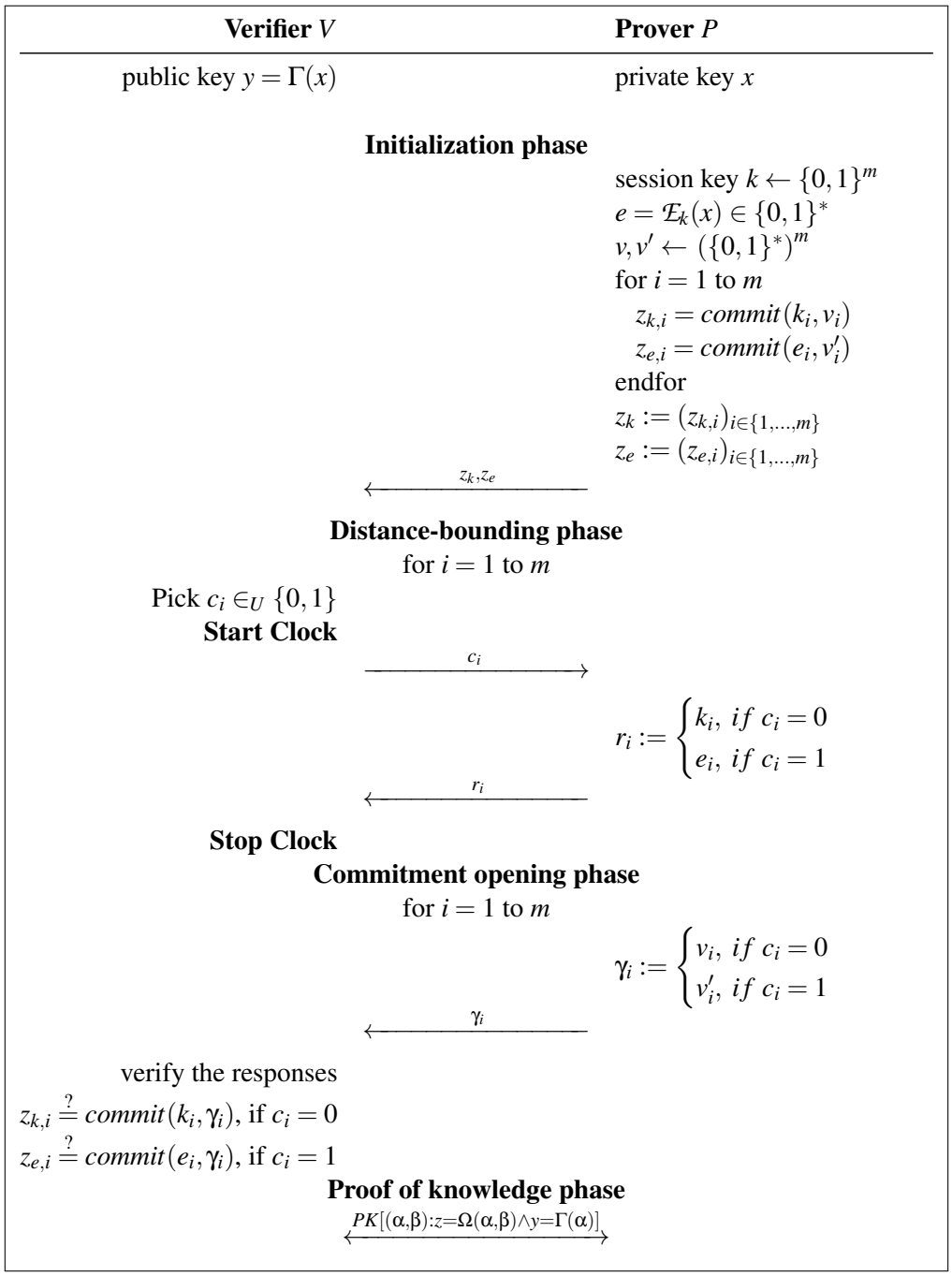


Fig. 3. The DBPK Protocol proposed by Bussard and Bagga [6,7,5]

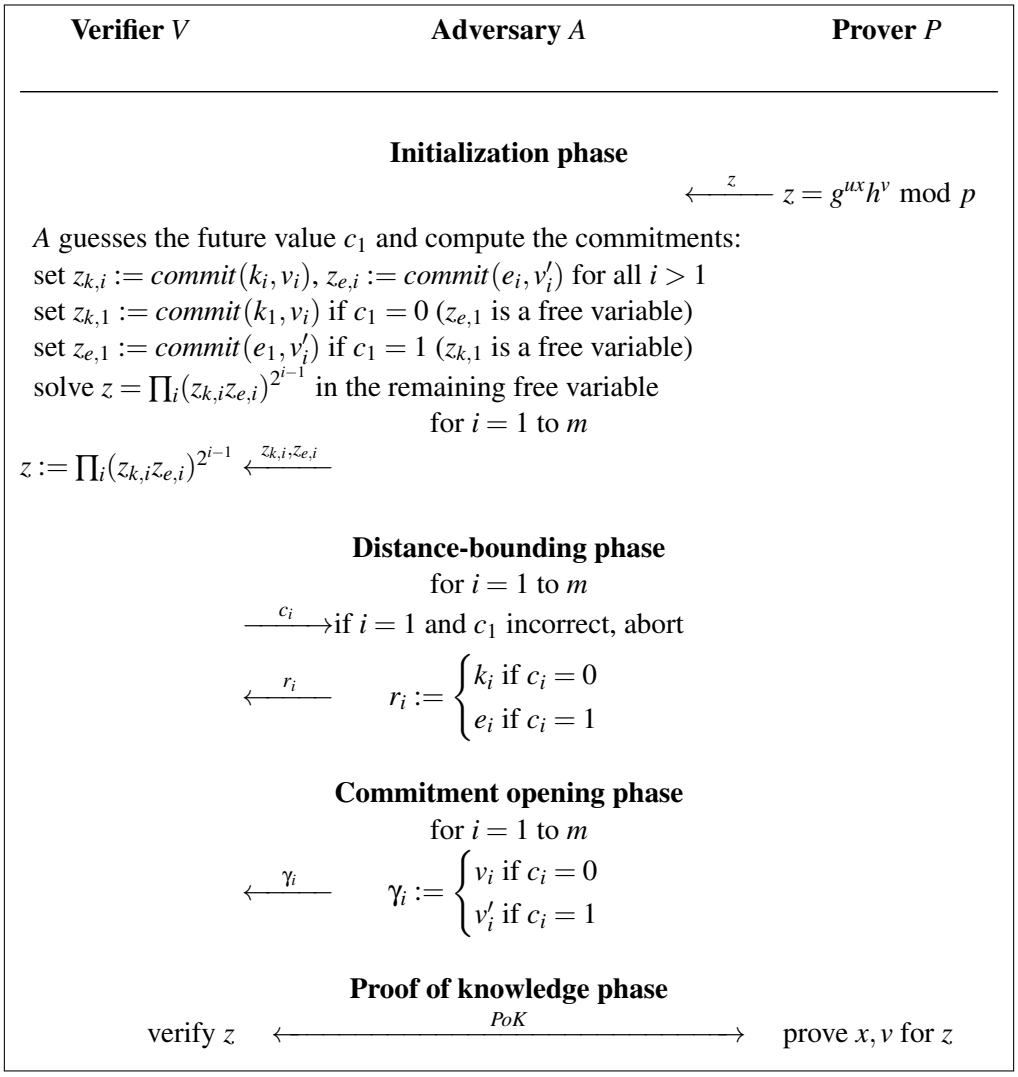


Fig. 4. Terrorist Fraud against DBPK-Log

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.[4] 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.

In fact, it is easy to see that the attack above can be transformed into a DF as well. This is also exhibited in [2].

In fact, [2] further proposes a series of attacks on follow-ups of the Bussard-Bagga protocols, this time protocols based only on symmetric key techniques. The authors of [2] notice that the information leaked from the combination of the protocol with the answers on the verifier's return channel can be exploited by a man-in-the-middle attacker to mount generalised mafia-fraud attacks.

We conclude this section by saying that recently alarming shortcomings of the security claims on DB have been published. Thus, we call for provable security of DB. In the following section, we are going to give some directive lines towards that, drawn from the lessons taught by the attacks above.

3 Towards Provably Secure DB

3.1 PRF Masking

First recall the general schema of DB in Fig.1. Also, remember that in the DF attack presented in Section 2.1, a dishonest prover was adaptively choosing a nonce N_P to influence the distribution of the output of the PRF instance $f_x(\ldots, N_P)$ and eventually mount a DF attack. To avoid such situations, we propose enhancing the DB protocols with a technique that we call **PRF masking**:
— instead of $f_x(\ldots)$ being identically calculated on $\text{messages}_P, \text{messages}_V$ on P's and V's sides, a bitstring a is chosen by the verifier and sent encrypted using the PRF instance f_x, i.e.,
$[M = a \oplus f_x(\text{messages}_P, \text{messages}_V)]$ is s sent by V to P.

3.2 Circular Keying Security

The MiM attacks presented in [3], attacks that follow the idea in Section 2.1, are possible due to the use of the key x both in the PRF instance and in the response-function F. This is not how PRF instances are normally used, when invoked for their security properties, i.e., for their random-like outputs. Therefore, we would like to amend the shortfalls presented in [3] by requiring that the PRF instances used inside DB protocols are such that their underlying key can be re-used outside of the function in some linear fashion, without this causing a key leakage. More precisely, we would require that a PRF f is *circular*

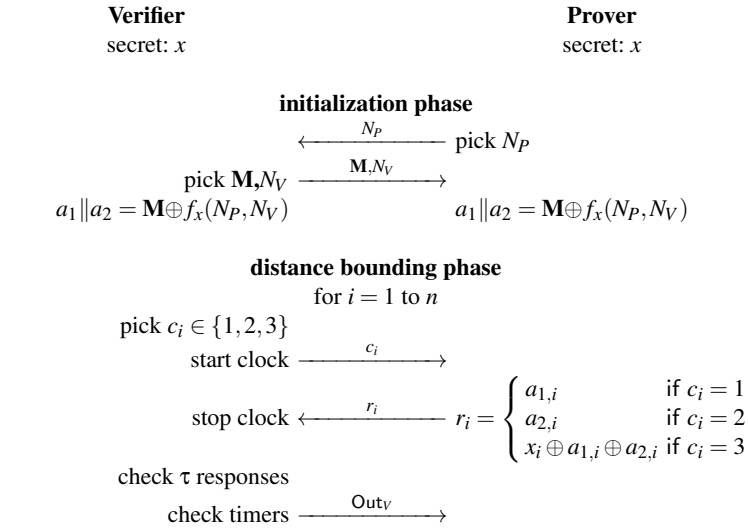
[4] The attack could also go on with the adversary taking c_1 as the value he selected, and counting on that the verifier will accept this error as due to noise.

keying secure :
— if $\mathcal{A}$ makes a query (y_i, a_i, b_i), the oracle answers $(a_i \cdot x') + (b_i \cdot f_x(y_i))$ and $\mathcal{A}$ cannot distinguish if $x = x'$ or x and x' are independent.

There is small caveat to this requirement, namely that for all $c_1, \ldots, c_q$ such that $c_1 b_1 + \cdots + c_q b_q = 0$, we must have $c_1 a_1 + \cdots + c_q a_q = 0$.

3.3 A Secure DB Attempt

Summing up the above requirements, we conjecture that a protocol of the following kind would resist the new attacks in [3].



NOTE: f is a **circular-keying secure** PRF and it has many possible variants

Fig. 5. A Protocol Directing Towards Provably Secure DB

4 Conclusions

With the above protocol in mind and with the recent threats summarised herein, we conclude that best-known attacks or security claims on DB are not enough. DB protocols will soon be implemented by car manufacturers or bank payment companies in their products, as platforms for such deployments arise [13]. Thus, what we would need now is solid communication/threat models, clear design and more than anything else reliable security proofs in these models. We launch therefore a call for this.

References

1. 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.
2. A. Bay, I. C. Boureanu, A. Mitrokotsa, I.-D. Spulber, and S. Vaudenay. The Bussard-Bagga and Other Distance-Bounding Protocols under Attacks. In *the 88th China International Conference on Information Security and Cryptology (Inscrypt 2012)*, 2012.

3. I. Boureanu, A. Mitrokotsa, and S. Vaudenay. On the Pseudorandom Function Assumption in (Secure) Distance-Bounding Protocols. In A. Hevia and G. Neven, editors, *Progress in Cryptology – LATINCRYPT 2012*, Lecture Notes in Computer Science, pages 100–120. Springer, 2012.
4. S. Brands and D. Chaum. Distance-Bounding Protocols (Extended Abstract). In *EUROCRYPT*, pages 344–359, 1993.
5. L. Bussard. *Trust Establishment Protocols for Communicating Devices*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Institut Eurécom, Télécom Paris, 2004.
6. 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.
7. 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.
8. C. Cremers, K. B. Rasmussen, and S. Čapkun. Distance hijacking attacks on distance bounding protocols. In *IEEE Symposium on Security and Privacy*, pages 113–127, 2012.
9. 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.
10. U. Dürholz, M. Fischlin, M. Kasper, and C. Onete. A Formal Approach to Distance Bounding RFID Protocols. In *Proceedings of the* 14^{th} *Information Security Conference ISC 2011*, LNCS, pages 47–62. SPRINGER, 2011.
11. G. Kapoor, W. Zhou, and S. Piramuthu. Distance Bounding Protocol for Multiple RFID Tag Authentication. In C.-Z. Xu and M. Guo, editors, *Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing - Volume 02 – EUC'08*, pages 115–120, Shanghai, China, December 2008. IEEE, IEEE Computer Society.
12. 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.
13. K. B. Rasmussen and S. Čapkun. Realization of rf distance bounding. In *Proceedings of the 19th USENIX conference on Security*, USENIX Security'10, pages 25–25, Berkeley, CA, USA, 2010. USENIX Association.
14. 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.
15. Y.-J. Tu and S. Piramuthu. RFID Distance Bounding Protocols. In *Proceedings of the First International EURASIP Workshop on RFID Technology*, 2007.