On the Pseudorandom Function Assumption in (Secure) Distance-Bounding Protocols

PRF-ness alone Does Not Stop the Frauds!

Ioana Boureanu, Aikaterini Mitrokotsa, and Serge Vaudenay

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
{ioana.boureanu,katerina.mitrokotsa,serge.vaudenay}@epfl.ch

Abstract. In this paper, we show that many formal and informal security results on distance-bounding (DB) protocols are incorrect/ incomplete. We identify that this inadequacy stems from the fact that the pseudorandom function (PRF) assumption alone, invoked in many security claims, is insufficient. To this end, we identify two distinct shortcomings of invoking the PRF assumption alone: one leads to distance-fraud attacks, whilst the other opens for man-in-the-middle (MiM) attacks. First, we describe –in a more unitary, formal fashion– why assuming that a family of functions classically used inside DB protocols is solely a PRF is unsatisfactory and what generic security flaws this leads to. Then, we present concrete constructions that disprove the PRF-based claimed security of several DB protocols in the literature; this is achieved by using some PRF programming techniques. Whilst our examples may be considered contrived, the overall message is clear: the PRF assumption should be strengthened in order to attain security against distance-fraud and MiM attacks in distance-bounding protocols!

1 Introduction

Distance-bounding (DB) protocols were introduced by Brands and Chaum [3] with the view of combating man-in-the-middle attacks against ATM systems. The main idea of DB protocols is that a tag (RFID card, smart card, etc.) should prove a short distance between them and a reader, and –most often than not– authenticate themselves in front of this reader. The authentication part is based on a pre-established secret. By default, this shared secret is a key hard-coded on the tag which the reader associates to the tag's id via a stored database. The tag is often referred to as the prover whereas the reader is referred to as a verifier. In the vast literature covering such protocols (e.g., [10,12,14,16]), three *main/classical* types of possible attacks have been distinguished. The first is *distance-fraud* (DF), in which a prover tries to convince that he is closer than what he really is. The second type of attack is the *mafia-fraud* (MF) attack, which involves three entities: an honest prover, an honest verifier and an adversary. The adversary communicates with both the prover and the verifier and tries to demonstrate to the verifier that the prover is in the verifier's proximity although

A. Hevia and G. Neven (Eds.): LATINCRYPT 2012, LNCS 7533, pp. 100–120, 2012.

the prover is in reality far away from the verifier. Finally, the third type of attack is denoted as *terrorist-fraud* (TF). Here, the adversary has the same goal as in the mafia-fraud attack, but in this case the prover is dishonest and colludes with the adversary up to the non-disclosure of essential information, i.e., secret keys or any other information that may more easily facilitate later impersonations of the tag. Other more generic MiM attacks have been imagined [4,6], generalising mainly distance-fraud or mafia-fraud respectively.

Meant to protect against such intricate attacks [3], implemented versions of DB protocols have only proven to be efficient in preventing relay attacks [5]. This is undeniably an important step. However, given the clear view of progressing in secure remote unlocking (e.g., [8]), distance-bounding protocols should be designed to resist against more *generic* (MiM, DF, TF,etc.) attacks, as aimed [3]. Whilst some attempts of formal models and formal proofs of security have recently arisen [1,6], provably secure distance-bounding is not at all a stable, well-founded area. For instance, we consider that [1], addressing the protection against terrorist-fraud using secret sharing schemes, only provides rather heuristic security analyses, failing to pinpoint the (necessary and) sufficient conditions for preventing TF on distance-bounding. On a parallel front, the model of Dürholz *et al.* in [6] is more attentive to detail, moving closer to provably secure DB. But, whilst [6,7] claim some security results, we believe that their informal proofs of security for DF and for MF are flawed. Thus, it is therein common to replace a PRF by a random function in a game-reduction proof, even if the PRF key is held by the adversary. This practice is obviously flawed. In this paper, we will show where these formal or informal proofs fell short of the correct arguments. Whilst we leave the concrete amendments of these issues for future work, we underline some concrete aspects that the state-of-art on (secure) DB has overlooked in their assessments, aspects that fundamentally compromise the security of these protocols. We formalise these concerns. We provide supporting examples, using PRF programming techniques, on a list of (claimed-to-be-secure) DB protocols[1] in the literature. We suspect that there are many more DB protocols susceptible to the kind of attacks we exhibit, especially since the DB protocols bear clear resemblances amongst them; The list of attacks herein is summarised in Table 1; as one can see, it comprises the somewhat popular DB protocols.

Table 1. Protocols Broken by DF or MiM attacks based On Faulty PRFs

Protocol	Distance-Fraud	MiM attack
TDB [1]	page 108	page 109
DFKO (Enhanced Kim-Avoine Protocol) [6]	page 110	–
Hancke and Kuhn's [9]	page 112	–
Avoine and Tchamkerten's [2]	page 114	–
Reid's *et al* [14]	page 115	page 116
Swiss Knife [12]	–	page 118

[1] In the concrete presentation, we will make this clear. Whilst a protocol may have not been claimed to be secure against all frauds, it was claimed to be secure against a specific fraud. In our analysis/exemplification, we will show the contrary.

Structure of the paper. The remainder of the paper is structured as follows. In Section 2, we present some reminders about distance-bounding protocols and PRF functions. In Section 3, we give one general construction of PRFs with trapdoors. This construction prompts to distance-fraud and MiM attacks in DB protocols. In Section 4, we present such attacks on the protocols listed in Table 1. We conclude in Section 5.

2 Distance-Bounding Protocols and the PRF Assumption

In this section, we will recall general facts about distance-bounding protocols and basic notions about pseudorandom functions.

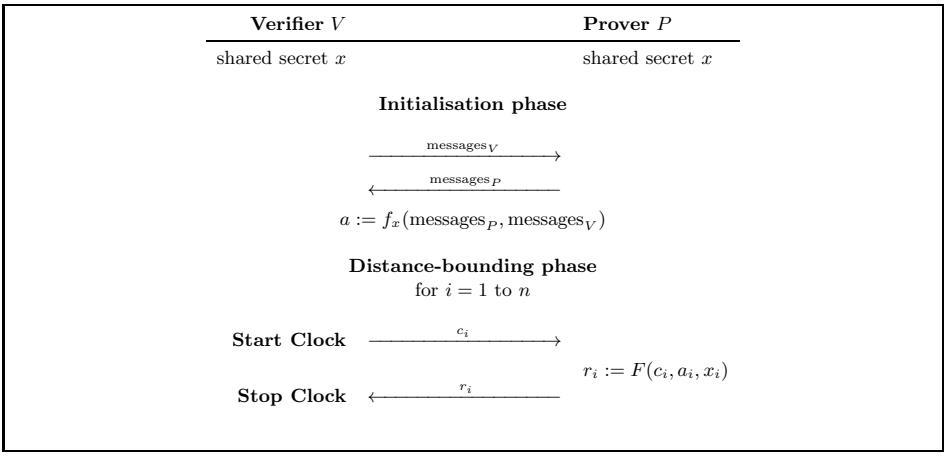

Fig. 1. Informative Sketch on Most Distance-Bounding Protocols

Distance-Bounding Protocols. The great majority of distance-bounding protocols [10,12,14,16] consist of a data-agreement phase or *initialisation phase* and a time-critical, fast computation-based *distance-bounding* phase. Fig. 1 captures the core of distance-bounding (DB). In the initialisation phase, a prover P and a verifier V use their randomnesses, their common secret x and a PRF f to exchange messages_P and messages_V respectively and establish a sub-secret a; a is normally a bitstring or a vector of elements in a finite space of small size. In the DB phase which is time-critical, the responses are normally defined via a response-function F. The i-th (one-bit) response r_i to the i-th randomly picked small-size challenge c_i is most often given by a computation of the sort $F(c_i, a_i, x_i)$, where $i \in \{1, \ldots, n\}$. The initialisation phase makes it possible for both parties to evaluate this function even though they do not have their counterpart's coins (i.e., the two honest parties have agreed over the vector a, they share x_i and they both know c_i).

Pseudorandom functions. A pseudorandom function (PRF) is a family of (polynomially computable) functions: a set of functions of arbitrary-length input and arbitrary-length output indexed on a set of keys. On this family, a computational assumption is taken, which is denoted as the *pseudorandom function (PRF) assumption*, i.e.,: for an instance sampled uniformly from the family, there exists no polynomial algorithm that distinguishes this instance from a real random function based on a black-box interaction with an oracle simulating them.

One can use the game-methodology [15] to formalise the PRF assumption. To this end, we give the descriptions/definitions below. Let $\mathcal{F}$ be a family of functions with domain D and range R. Let b be a bit. Let $\mathcal{D}$ be a ppt. distinguisher that can interact in a black-box manner with an oracle $\mathcal{O}$. We denote this interaction as $\mathcal{D}^{\mathcal{O}}$ and it is depicted as follows in Fig. 2.

1: **Parameters:** security parameter s; *poly* a polynomial; a ppt. algorithm $\mathcal{D}$; $\ell := \ell(s)$; $L := L(s)$; $D = \{0,1\}^{\ell}$; $R = \{0,1\}^{L}$;

2: $view_{\mathcal{D}} := \emptyset$
3: **while** nb. of iterations $\leq poly(s)$ **do**
4: $\quad x \leftarrow \mathcal{D}(view_D; r_{\mathcal{D}})$; $x \in D$
5: $\quad$ if $x =$ "end : b" with $b \in \{0,1\}$, stop and return b
6: $\quad y \leftarrow \mathcal{O}(x)$; $y \in R$
7: $\quad view_{\mathcal{D}} := view_{\mathcal{D}} \cup \{y\}$
8: **end while**
9: return 0

Fig. 2. The $\mathcal{D}^{\mathcal{O}}$ Interaction

Below, we will simply refer to the oracle implementing f^0 or f^1 by f^b accordingly, responding with $f_b(x) \in R$ for a query $x \in D$. Assume the following description of the PRF game, in Fig. 2.

1: **Parameters:** security parameter s; $\ell := \ell(s)$; $L := L(s)$; $D = \{0,1\}^{\ell}, R = \{0,1\}^{L}$; a family $\mathcal{F} := \mathcal{F}(s)$ of functions from $D \to R$; a ppt. algorithm $\mathcal{D}$; a bit b.

2: $f^0 \longleftarrow_U [D \to R]$ // pick a random function from D to R
3: $f^1 \longleftarrow_U \mathcal{F}$ //sample a function from the family
4: $\overline{b} \longleftarrow \mathcal{D}^{f^b}$
5: return $\overline{b}$

Fig. 3. The PRF Game $PRF^b_{\mathcal{F},\mathcal{D}}$

The output of the above game (0 or 1) is denoted $Out(PRF^b_{\mathcal{F},\mathcal{D}})$.

Definition 1 (The PRF assumption[2]). *Let s be a security parameter, k, ℓ, L be some parameters taken as functions of s, $\mathcal{K} = \{0,1\}^k$, $\mathcal{D} = \{0,1\}^{\ell}$,*

[2] This is formalised similarly to [13].

$R = \{0,1\}^L$. *Let* $\mathcal{F}$ *be a family of functions* $(f_k)_{k\in\mathcal{K}}$ *with* $f_k : D \to R$[3] *(an indexed-set of functions over* $\mathcal{K}$*).*

We say that the family $\mathcal{F}$ *is a* PRF *or that the family* $\mathcal{F}$ *respects the* PRF assumption *if for any ppt. algorithm* $\mathcal{D}$*,*

$$\left|\Pr[Out(PRF^0_{\mathcal{F},\mathcal{D}}) = 1] - \Pr[Out(PRF^1_{\mathcal{F},\mathcal{D}}) = 1]\right| < negl(s),$$

where negl is a function over natural numbers eventually lower than the inverse of any polynomial and the probability is taken over the random coins of $\mathcal{D}$*.*

We will also employ the notion of a hard-core function.

Definition 2 (Hard-core function). *Let* s *be a security parameter,* k*,* ℓ*,* L *be some parameters taken as functions of* s*,* $\mathcal{K} = \{0,1\}^k$*,* $\mathcal{D} = \{0,1\}^\ell$*,* $R = \{0,1\}^L$*. Let* $\mathcal{F}$ *be a family of functions* $(f_k)_{k\in\mathcal{K}}$ *with* $f_k : D \to R$*. A function* h *on* $\{0,1\}^*$ *is a hard-core function for* $\mathcal{F}$ *if for all polynomial-time oracle adversary* $\mathcal{A}$ *playing the following game, the probability that it wins is negligible.*

1: pick $k \in \mathcal{K}$
2: run $z = \mathcal{A}^{f_k(\cdot)}$
3: win if and only if $z = h(k)$

Note that if $\mathcal{F}$ is a PRF, then the identity function is hard-core. Further, observe that if h truncates to half of the first bits, it may not be hard-core for a PRF. Indeed, let $f_{k_0,k_1}(x) = k_0$ when $x = 0$ and $f_{k_0,k_1}(x) = g_{k_1}(x)$ when $x \neq 0$; if g is a PRF, then f is a PRF as well, but $h(k_0, k_1) = k_0$ is clearly not hard-core. We could still transform a PRF g into a PRF f for which h is hard-core, for instance with $f_{k_0,k_1}(b, x) = g_{k_b}(x)$.

3 PRFs with a Trapdoor

In this section, we are going to show how, out of a PRF $\mathcal{G}$, one can program another PRF $\mathcal{F}$ to accommodate a trapdoor making its instances leak a special value when called on that trapdoor. Otherwise, an instance of the thus-wise constructed PRF $\mathcal{F}$ "behaves" like the corresponding instance of $\mathcal{G}$. The ultimate goal of these constructions is to (help) show that the PRF assumption is not enough for the security of DB protocol, as claimed [1,6]. In fact, inappropriate PRFs used in DB protocols can lead to frauds: the first construction points to distance-fraud and the second to man-in-the-middle (MiM) attacks.

Consider the following informal explanations related to construction. Consider a function σ, with the aim of mapping an element of a domain $\mathcal{K} \times D$ onto an element of a domain R. Typically, σ embeds the input $k \in \mathcal{K}$ so that its output leaks k. Similarly, $\mathsf{correctPad}$ maps elements from $\mathcal{K}$ onto disjoint subsets of the set D above. Also, $\mathsf{correctPad}(k)$ must be such that its inverse is computable,

[3] We denote a function $f_k \in \mathcal{F}$, for a fixed $k \in \mathcal{K}$ as a PRF instance.

i.e., the token k is extractable out of any $\mathsf{correctPad}(k)$ element. We formalise this below and use it to formulate our result on PRF-constructions.

Theorem 3. *Let s be a security parameter. Let the following sizes of domains be expressed in function of s: ℓ, $\bar{\ell}$, L, $\bar{L}$, k. Consider the following three sets $D = \{0,1\}^{\ell}$, $\bar{D} = \{0,1\}^{\bar{\ell}}$, $R = \{0,1\}^{L}$, $\bar{R} = \{0,1\}^{\bar{L}}$, $\mathcal{K} = \{0,1\}^{k}$.*

Let h be a polynomially computable function on $\{0,1\}^*$.

Let $\mathcal{G}$ be a family of functions $(g_k)_{k\in\mathcal{K}}$ and $g_k : \bar{D} \to \bar{R}$. We assume that $\mathcal{G}$ is a PRF and that h is a hard core function for $\mathcal{G}$. Let $T^{\mathcal{O}}$ be a polynomial-time oracle-algorithm accessing $\mathcal{O}$, admitting inputs in D and outputs in R.

Consider a polynomially computable function σ from $\mathcal{K} \times D$ to R.

Consider a map $\mathsf{correctPad}$ from $\mathcal{K}$ to the set of subsets of D such that there exists a polynomial time oracle-algorithm $\mathsf{extract}^{g_k(\cdot)}$ from D such that for any $k \in \mathcal{K}$ and $x \in \mathsf{correctPad}(k)$, we have $\mathsf{extract}^{g_k(\cdot)}(x) = h(k)$. It is further assumed that given x and k, it can be decided in polynomial time whether x belongs to $\mathsf{correctPad}(k)$ or not.

Let a $\mathcal{F}$ be a family of functions $(f_k)_{k\in\mathcal{K}}$ and, for some arbitrarily fixed $k \in \mathcal{K}$, $f_k : D \to R$ defined as follows:

$$f_k(x) = \begin{cases} \sigma(k,x), & \text{if } x \in \mathsf{correctPad}(k) \\ T^{g_k(\cdot)}(x), & \text{otherwise .} \end{cases}$$

Then, the family $\mathcal{F}$ is a PRF.

The proof of Theorem 3 is natural, following the game-reduction methodology [15], by indistinguishability between games based on failure-events.

Proof. We first observe that since membership of $\mathsf{correctPad}$ and σ can be computed in polynomial time, then f is polynomially computable as well.

Let $k \in \mathcal{K}$ be arbitrarily fixed. Consider the distinguisher $\mathcal{D}$ distinguishing $(f_k)_{k\in\mathcal{K}}$ in the $PRF^b_{\mathcal{F},\mathcal{D}}$ game. Let $(x_1, f_k(x_1)), \ldots, (x_n, f_k(x_n))$ be the query-reply tuples between $\mathcal{D}$ and the oracle in $PRF^1_{\mathcal{F},\mathcal{D}}$, for $n \leq poly(s)$, with $poly$ and s defined in $PRF^b_{\mathcal{F},\mathcal{D}}$, $x_i \in D$, $f_k(x_i) \in R$, for all $i \in \{1,\ldots,n\}$.

Clearly, $\Pr[\mathcal{D} \text{ wins in } PRF^0_{\mathcal{F},\mathcal{D}}] = \Pr[\mathcal{D} \text{ wins in } PRF^0_{\mathcal{G},\mathcal{D}}]$. Since $\mathcal{G}$ is a PRF, we further have

$$\left| \Pr[\mathcal{D} \text{ wins in } PRF^0_{\mathcal{G},\mathcal{D}}] - \Pr[\mathcal{D} \text{ wins in } PRF^1_{\mathcal{G},\mathcal{D}}] \right| = negl(s).$$

So, we just have to show that

$$\left| Pr[\mathcal{D} \text{ wins in } PRF^1_{\mathcal{F},\mathcal{D}}] - \Pr[\mathcal{D} \text{ wins in } PRF^1_{\mathcal{G},\mathcal{D}}] \right| = negl(s).$$

Unless $\mathcal{D}$ queries x_i ($i \in \{1,\ldots,n\}$) with $x_i \in \mathsf{correctPad}(k)$, his view is that of $\mathcal{D}$ in the "corresponding" $PRF^1_{\mathcal{G},\mathcal{D}}$ with the same random coins, i.e.,

$$\ldots, (x_{i-1}, T^{g_k(\cdot)}(x_{i-1})), (x_i, T^{g_k(\cdot)}(x_i)), (x_{i+1}, T^{g_k(\cdot)}(x_{i+1})), \ldots, (x_n, T^{g_k(\cdot)}(x_n))$$

In the contrary case, where he does query $x_i \in \mathsf{correctPad}(k)$, the view of $\mathcal{D}$ contains $\sigma(k, x_i)$ instead of $T^{g(x_i)}$ (for this fixed i).

So, the game $PRF^1_{\mathcal{F},\mathcal{D}}$ is indistinguishable from the game $PRF^1_{\mathcal{G},\mathcal{D}}$ unless the failure-event F of querying the specific $x_i \in \mathsf{correctPad}(k)$ occurs. In other words, $\Pr[\mathcal{D} \text{ wins in } PRF^1_{\mathcal{F},\mathcal{D}} | \neg F] = \Pr[\mathcal{D} \text{ wins in } PRF^1_{\mathcal{G},\mathcal{D}} | \neg F]$. So,

$$\left| Pr[\mathcal{D} \text{ wins in } PRF^1_{\mathcal{F},\mathcal{D}}] - \Pr[\mathcal{D} \text{ wins in } PRF^1_{\mathcal{G},\mathcal{D}}] \right| \leq \Pr[F].$$

What is left to be proven is that $\Pr[F]$ is negligible. To bound the probability $\Pr[F]$ of F occurring, we let p_i be the probability that $x_i \in \mathsf{correctPad}(k)$ and that $x_j \notin \mathsf{correctPad}(k)$ for $j \in \{1, \ldots, i-1\}$. Clearly, $\Pr[F] \leq \sum_i p_i$.

So, this reduces to proving that p_i is negligible for each i. To do so, we construct a new algorithm $\mathcal{A}^{g_k(\cdot)}$. Namely, $\mathcal{A}$ simulates $\mathcal{D}$ and T until it computes x_i. Then, the algorithm $\mathcal{A}$ uses x_i to get $k' = \mathsf{extract}^{g_k(\cdot)}(x_i)$.
In the case that $x_j \notin \mathsf{correctPad}(k)$ for $j = 1, \ldots, i-1$, the simulation is perfect. If $x_i \in \mathsf{correctPad}(k)$, then $k' = h(k)$ and we obtain that $\mathcal{A}$ outputs $h(k)$. So, $\Pr[\mathcal{A} \text{ yields } h(k)] \geq p_i$. Since h is hard-core, p_i is negligible. □

The first note on Theorem 3 is that a PRF can be constructed, if PRFs exist. I.e., starting from $\mathcal{G}$ being some PRF, Theorem 3 gives the concrete construction of another PRF $\mathcal{F}$, with a trapdoor.

Then, one of the aims of this result is to indicate that if an inappropriate PRF $\mathcal{F}$ is used in (the initialisation phase of) DB protocols, then a distance-fraud can be mounted onto those protocols. To see this easily, you may want to refresh the notations in Fig. 1 informally describing the DB protocols. Now, imagine a dishonest prover P^* (who of course has the shared-key x and) that wants to mount a distance-fraud onto a DB protocol using a trapdoor-enhanced PRF $\mathcal{F}$ as the one in Theorem 3. By applying an input from $\mathsf{correctPad}(x)$, he sends messages to the verifier V such that $(\text{messages}_P, \text{messages}_V) \in \mathsf{correctPad}(x)$. Then, $f_x(\text{messages}_P, \text{messages}_V) = \sigma(x, \text{messages}_P, \text{messages}_V)$. Usually, in DB protocols (e.g., [1,6,9,2], etc.), messages_P is in fact a nonce N_P and messages_V is a nonce N_V. So, an example of such adaptive choices and exploitation of poor PRFs is the following: P^* can choose adaptively N_P to be, say, x and then $f_x(\text{messages}_P, \text{messages}_V)$ becomes $f_x(x, N_V)$ which is equal –by the trapdoor property– with, say, $x\|x\|\ldots\|x$. Since the responses are based on this output and x, this usually enables P^* to answer any challenge before they even arrive at him. This means that he successfully mounts a distance-fraud attack. Of course, this sort of artificial function and its trapdoor depend on the protocol under discussion, as Section 4 will show. I.e., we need appropriate special $\sigma(x, N_P, N_V)$ to be output of the PRF instances. (Usually, it simply implies that the response is a constant in terms of the challenge). Also, other forms of output of the PRF instances can be imagined, as long as they facilitate the responses of the DB phase to be independent from the challenge (i.e., instead of $\sigma(x, \text{messages}_P, \text{messages}_P)$ we could directly some constant *cte* known to P^* and lying in the appropriate domains used in the above theorem). It is also needed that the distribution of such outputs and the domains we have at hand, $\sigma(x, \text{messages}_P, \text{messages}_P)$ seems a reasonable choice for the "conned" protocols participants.

This construction of a "trapdoor PRF" (from a given PRF) uses an oracle in the inversion of correctPad only for the purpose of it giving raise not only to DF but to MiM attacks also. The basic idea of MiM attacks of this sort relies on a PRF $(f_x)_{x\in\mathcal{K}}$ such that $f_x(y) = x$ when $y = g_x(cte) + x$, where g_x is a PRF instance from a given PRF $\mathcal{G}$. By adapting this generic construction, we could have an adversary first getting y by querying a specific set of challenges c_i to the prover, then using y as a nonce to extract x from the prover. A specific, detailed description of an attack of the sort is presented in page 109 against the TDB protocol [1]. For the proof of such $(f_x)_{x\in\mathcal{K}}$ being a PRF when constructed as in Theorem 3, there is the need that the inversion of correctPad is made via an access to an oracle of the stated sort.

4 PRF-Based Attacks

4.1 TDB Protocol

In this protocol, due to [1] and depicted in Fig. 4, the prover P and the verifier V share a secret s that can be viewed as a vector $(s_1, \ldots, s_m)$ of m coordinates over a group G, i.e., $s_i \in G$, $i \in \{1, m\}$. The prover P and the verifier V use an (n, k) threshold scheme on some sub-secrets obtained via a pseudo-random function instance f_s. Like in most cases, the protocol is divided into two phases: the *initialisation* phase and the *distance-bounding* phase.

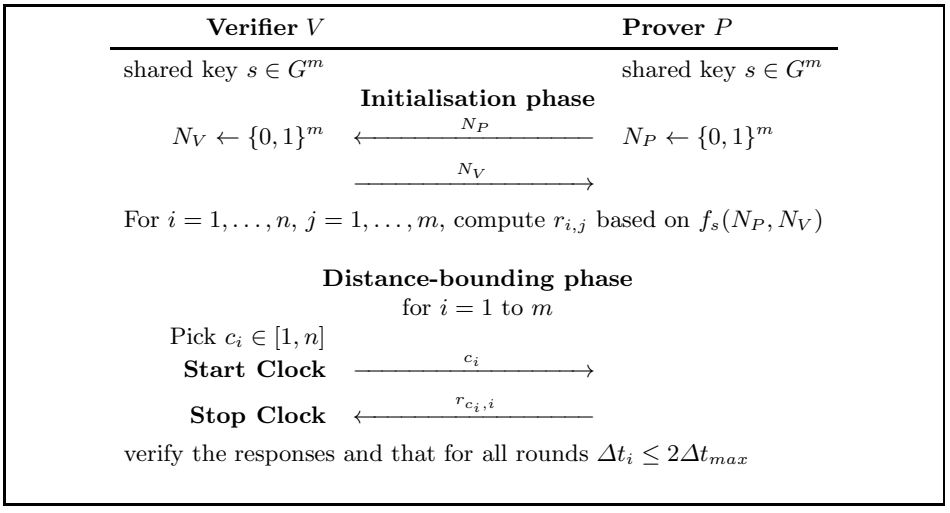

Fig. 4. The TDB protocol [1]

- **Initialisation Phase:** This phase is not time critical. The prover P and the verifier V select two random nonces N_P and N_V correspondingly and transmit them to each other. Then, both the prover P and the verifier V compute an $n \times m$ matrix $\mathcal{R}$, where each column $(r_{1,i}, r_{2,i}, \ldots, r_{n,i})^T$ of $\mathcal{R}$ is obtained using the

(n, k) threshold scheme applied on s_i. Namely, the $r_{c_i,i}$'s are generated with the help of a pseudorandom function f by computing $f_s(N_P, N_V)$ and the elements of the last row of the matrix are generated by summing all the elements in the upper rows on the same column and adding the corresponding secret bit of s. For instance, if we consider the case where $n = k = 3$, $G = F_2$ and a specific secret sharing scheme, then the response matrix will have dimension $3 \times m$ and will have the form:

$$\mathcal{R}_1 = \begin{pmatrix} r_{1,1} & \cdots & r_{1,m} \\ r_{2,1} & \ddots & r_{2,m} \\ s_1 \oplus r_{1,1} \oplus r_{2,1} & \cdots & s_m \oplus r_{1,m} \oplus r_{2,m} \end{pmatrix}$$

- **Distance Bounding Phase:** This phase is time critical and involves the exchange of challenges-responses (rounds) at maximum bit-rate. Such an exchange is repeated m times (i.e., there are m rounds). Assume a generic round i (for i varying from 1 to m). At each round i the challenge-response delay Δt_i is measured. The verifier V starts by choosing a random c_i in its domain $\{1, \ldots, n\}$, initialising the clock to zero and transmitting c_i to P. The prover P responds with $r_{c_i,i}$ which denotes the element located at the c_i-th row and the i-th column of the table $\mathcal{R}$. On receiving $r_{c_i,i}$, V stops the clock and stores the received response and the delay time Δt_i.

After the end of the distance-bounding phase, a verification phase is performed and the verifier V checks if the received responses are correct and if for the response times Δt_i it holds that $\Delta t_i \leq 2\Delta t_{max}$, where Δt_{max} denotes the time it takes for a bit to be transmitted from the prover to the verifier.

Based on the construction and significance of Theorem 3, we construct the following attack on an instantiation of the TDB protocol. The same sort of attack would work for other instantiations of the TDB protocol (i.e., with different secret sharing schemes inside).

Distance Fraud Attack on an Instantiation of the TDB Protocol. Let g be a PRF from $\{0,1\}^{2m}$ to itself. Let us consider the PRF f constructed from Theorem 3 based on g and the following elements. Let $T^g(x) = g(x)$. Let D be the set of (N_P, N_V) pairs. Let $\sigma(s, N_P, N_V) = s \| s$, $\mathsf{correctPad}(s) = \{s \| N_V; N_V \in \{0,1\}^m\}$, and $\mathsf{extract}^{g_s(.)}(N_P, N_V) = N_P$. We have

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

By Theorem 3, f is a PRF. Consider an instantiation of the TDB protocol, where the response matrix is $\mathcal{R}_1$ above and the PRF f is being used.

In this instance of the TDB protocol, it is obvious that a legitimate, far-away but dishonest prover could easily perform a distance-fraud attack. He just needs to choose N_P to be equal to s (as shown above). Then, the $\mathcal{R}_1$ matrix has all its rows equal to s. So for any challenge c_i the response will be the i-th bit of the secret key s. This sort of fixed responses can be sent before receiving the

challenge. Thus, he can defeat the distance-bound. The extension to n, k greater than 3 is trivial: in the trapdoor case of f_s, one repeats s for $n-1$ times and then considers the case where n is odd and even separately.

So, if a PRF exists, then we can exhibit instances of TDB which are insecure against DF! The PRF assumption is not enough for the security of the TDB protocols against DF.

Man-in-the-Middle Attack on an Instantiation of the TDB Protocol. Consider again the instantiation of the TDB protocol with $n = k = 3$, $G = F_2$ and with $\mathcal{R}_1$ being the response matrix. Let the shared key be denoted by s. Let g be a PRF mapping $\{0,1\}^{\frac{m}{2}}$ to itself and from $\{0,1\}^{2m}$ to $\{0,1\}^{\frac{3m}{2}}$. We assume that the least significant half of s is hard-core for g. We define $T^{g_s(\cdot)}(N_P, N_V) = (\alpha, \beta, \gamma, \beta \oplus g_s(\alpha))$ where $g_s(N_P, N_V) = (\alpha, \beta, \gamma)$. Let us consider the following elements: $\sigma(s, N_P, N_V) = s \| s$, $\mathsf{correctPad}(s) = \{N_P \| \overline{\alpha} \| (g_s(\overline{\alpha}) \oplus \mathsf{lsb}_{\frac{m}{2}}(s)) \, ; \, \overline{\alpha} \in \{0,1\}^{\frac{m}{2}}\}$, and $\mathsf{extract}^{g_s(\cdot)}(N_P, \overline{\alpha}, \overline{\beta}) = \overline{\beta} \oplus g_s(\overline{\alpha})$. Let f be constructed from Theorem 3 based on g as below. By Theorem 3, f is a PRF.

$$f_s(N_P, N_V) = \begin{cases} (\alpha, \beta, \gamma, \beta \oplus g_s(\alpha)), & \text{if } N_V \text{ is not of the form } \overline{\alpha} \| (g_s(\overline{\alpha}) \oplus \mathsf{lsb}_{\frac{m}{2}}(s)) \\ & \text{and } (\alpha, \beta, \gamma) = g_s(N_P, N_V) \\ s \| s, & \text{if } N_V = \overline{\alpha} \| (g_s(\overline{\alpha}) \oplus \mathsf{lsb}_{\frac{m}{2}}(s)), \text{ for some } \overline{\alpha} \end{cases}$$

In the notations of the TDB protocol, $r_1 = (\alpha, \beta)$ and $r_2 = (\gamma, \beta \oplus g_s(\alpha))$.

We are now going to explain the attack. The attacker has the goal of recovering s from the prover, so that he can later impersonate this prover as he pleases. To do so, the attacker impersonates first the verifier to the prover. He sends an arbitrary N_V, so the prover calculates the generic subsecret vectors $(\alpha', \beta', \gamma', \psi')$ as some $(\alpha, \beta, \gamma, \beta \oplus g_s(\alpha))$. Then the adversary sends many challenges equal to 1, $c_i = 1$, e.g., for $i \in \{1, \ldots, \frac{m}{2}\}$. In this way, he gets the first half of the first subsecret-vector $r_1{=}(\alpha, \beta)$, i.e., he obtains α,. Then, the adversary sends the prover many challenges equal to 3, some $c_i = 3$. By the secret sharing scheme used, the responses to the latest challenges are equal to $r_1 + r_2 + s = (\alpha, \beta) \oplus (\gamma, \beta \oplus g_s(\alpha)) \oplus s = (\alpha \oplus \gamma, g_s(\alpha)) \oplus s$. So, from this approach, the attacker gets $g_s(\alpha) \oplus \mathsf{lsb}_{\frac{m}{2}}(s)$. Finally, he can now form $N'_V = \alpha \| (g_s(\alpha) \oplus \mathsf{lsb}_{\frac{m}{2}}(s))$. The second step of the attack (in a new hijacked session in which the attacker is again impersonating the verifier to the honest prover) consists in the attacker to employ his knowledge gained as above to choose N_V equal to N'_V. By then injecting any challenges to the prover, the attacker will know (due to the built-in PRF) that the responses of the prover will be the bits of s. Like this, he will learn the whole of the secret key and he will be subsequently able to impersonate this prover in any circumstance.

Again, according to Theorem 3, the resulting family of functions $(f_s)_{s \in G^m}$ is a PRF. The attack exhibited does therefore disprove the claims of MiM security in [1] based solely on the PRF assumption.

4.2 DFKO Protocol (Enhancement of the Kim-Avoine Protocol)

The protocol proposed by Dürholz *et al.* [6] is based on the protocol proposed by Kim and Avoine [11]. More precisely, the protocol proposed by Kim and Avoine [11] is claimed to be mafia and distance-fraud resistant. Dürholz *et al.* [6] have modified it in order to safeguard against impersonation attacks. Its sketch is depicted in Fig. 5. This protocol is again discriminated into two phases: the *initialisation phase* and the *distance-bounding phase*. During the *initialisation phase* both the prover P and the verifier V select a random nonce N_P and a random nonce N_V correspondingly. Then, both of them compute the output of a PRF instance f_x that takes as input the two random nonces that they have exchanged, i.e.,: $I\|C\|D\|v^0\|v^1 := f_x(N_V\|N_P)$, where x is shared secret/key. The prover P also sends part of this output (i.e., I) to the verifier and V verifies that this is the correct value for I.

During the *distance-bounding phase* the verifier V sends the bit-challenges R_i to the prover that are either generated randomly or using the value D (i.e., a part of the output of $f_x(N_P, N_V)$ generated in the *initialisation phase*). If the challenges are random, then the responses are also random otherwise the responses depend on the value v^0 (i.e., a part of the output of $f_x(N_P, N_V)$ generated in the *initialisation phase*). For more details regarding the Dürholz *et al.* protocol, we refer the readers to [6].

Based on Theorem 3, we now give an attack on an instantiation of DFKO.

Distance Fraud Attack on the DFKO Protocol. Let g be a PRF with parameters as the one needed in the DFKO protocol. Let us consider the PRF f constructed from Theorem 3 based on g and the following elements. Let $T^g(x) = g(x)$. Let $\sigma(x, N_P, N_V) = I\|0\cdots0\|D\|v^0\|v^1$ where $g_x(N_P, N_V)$ parses into $I\| \cdot \|D\|v^0\|v^1$, $\mathsf{correctPad}(x) = \{x\|N_V; N_V \text{ nonce}\}$, and $\mathsf{extract}^{g_x(\cdot)}(N_P, N_V) = N_P$. We have

$$f_x(N_P, N_V) = \begin{cases} I\|0\ldots0\|D\|v^0\|v^1, & \text{if } N_P = x \text{ for } g_x(x, N_V) = I\|C\|D\|v^0\|v^1 \\ g_x(N_P, N_V) & \text{otherwise,} \end{cases}$$

By Theorem 3, f is a PRF.

Consider an instantiation of the DFKO protocol, where the PRF f is being used. In this instance of the DFKO protocol, if a prover is dishonest and picks N_P to be equal to x, then the response of the prover P will always be a bit of D. Thus, this dishonest prover would be able to know in advance all the responses to every possible challenge. Thus, when being in fact far-away from V, he would perform successfully a distance-fraud attack. This obviously contradicts the security against distance-fraud attacks that is claimed by Dürholz *et al.* in Theorem 2 of [6], based solely on the PRF assumption and described only by a sketch-proof in the appendix of [6].

4.3 Hancke and Kuhn's Protocol

Hancke and Kuhn's protocol [9] is again separated conceptually into two phases: an *initialisation phase* and a *distance-bounding phase*. In the *initialisation phase*

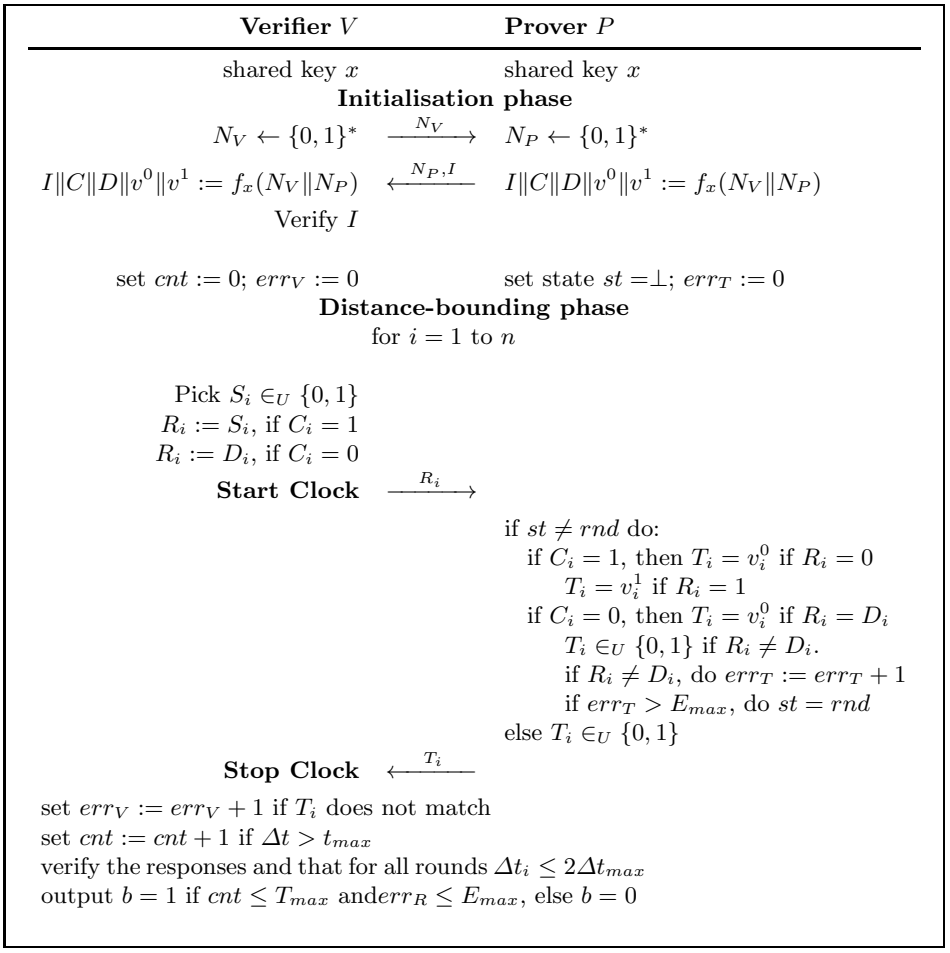

Fig. 5. Protocol Dürholz *et al.* [6] (Enhanced Kim-Avoine Protocol)

the verifier V and the prover P exchange nonces N_V and N_P correspondingly, then both of them compute the output of a PRF instance f_x that takes as input the two nonces, i.e., $v^0 \| v^1 := f_x(N_P, N_V)$, where x is the shared secret/key. During the *distance-bounding phase* the verifier V selects a random bit-challenge c_i, where $i \in \{1, \ldots, n\}$ and the prover P responds with a bit r_i that has the following form: $r_i := \begin{cases} v_i^0, & \text{if } c_i = 0 \\ v_i^1, & \text{if } c_i = 1. \end{cases}$
After the end of the distance-bounding phase the verifier checks all the received responses and if the response times Δt_i satisfy the condition: $\Delta t_i \leq 2\Delta t_{max}$. The protocol is depicted in Fig. 6.

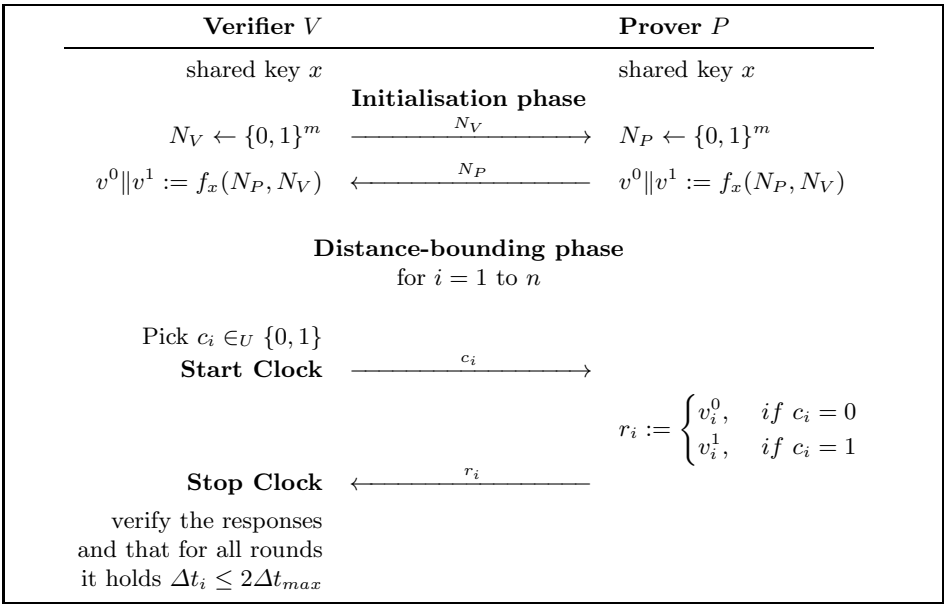

Fig. 6. Protocol proposed by Hancke and Kuhn's [9]

Based on the construction and significance of Theorem 3, we construct the following attack on an instantiation of Hancke and Kuhn's protocol.

Distance Fraud Attack on Hancke and Kuhn's Protocol. Let g be a PRF with parameters as the one needed in the protocol. Let us consider the PRF f constructed from Theorem 3 based on g and the following elements. Let $T^g(x) = g(x)$. Let D be the set of (N_P, N_V) pairs. Let $\sigma(x, N_P, N_V) = v \| v$ where v is an arbitrary constant, $\mathsf{correctPad}(x) = \{x \| N_V; N_V \in \{0,1\}^m\}$, and $\mathsf{extract}^{g_x(.)}(N_P, N_V) = N_P$. We have

$$f_x(N_P, N_V) = \begin{cases} v \| v, & \text{if } N_P = x \\ g_x(N_P, N_V), & \text{otherwise} \end{cases}$$

By Theorem 3, f is a PRF.

Consider an instantiation of the Hancke and Kuhn where the PRF f is being used. In this instance of the Hancke and Kuhn protocol, if a prover is dishonest and picks N_P to be equal to x, then the responses r_i will always be the same for all challenges c_i where $i \in \{1, \ldots, n\}$. Thus, such a legitimate, far-away but dishonest prover can successfully perform a distance-fraud attack by sending the responses before receiving the challenges. This obviously contradicts the security against distance-fraud attacks that was claimed in Theorem 3.2 of [7] solely based on the PRF assumption and that claims of [9].

4.4 Avoine and Tchamkerten's Protocol

This protocol from [2], presented in Fig. 7, is again divided into two phases an *initialisation* and a *distance-bounding* base. The prover P and the verifier V share a common secret x and they have agreed on some parameters m and n. In the *initialisation phase* which is not time critical, the verifier V selects a random nonce N_V and transmits it to the prover P. The prover P also selects a random nonce N_P and transmits it to V. Then, they compute the output of a PRF instance f_x on the input given by the two nonces N_P and N_V, i.e., $v^0 \| v^1 := f_x(N_P, N_V)$. The output of this computation has length at least $m + 2^{n+1} - 2$. We denote the first m bits of this output by v^0 and the rest of the bits by v^1. Then, the prover P sends to the verifier V the value v^0 for verification purposes.

Verifier V		**Prover** P
shared key x		shared key x
	Initialisation phase	
$N_V \leftarrow \{0,1\}^m$	$\xrightarrow{N_V}$	$N_P \leftarrow \{0,1\}^m$
$v^0 \| v^1 := f_x(N_P, N_V)$	$\xleftarrow{v^0, N_P}$	$v^0 \| v^1 := f_x(N_P, N_V)$
abort if v^0 is incorrect		where $\lvert v^0 \rvert = m$ and $\lvert v^1 \rvert = 2^{n+1} - 2$
	Distance-bounding phase for $i = 1$ to n	
Pick $c_i \in_U \{0,1\}$		
Start Clock	$\xrightarrow{c_i}$	
		$r_i = Node(c_1, \ldots, c_i)$
Stop Clock	$\xleftarrow{r_i}$	
verify the responses and that for all rounds it holds $\Delta t_i \leq 2\Delta t_{max}$		

Fig. 7. Protocol proposed by Avoine and Tchamkerten [2]

For the *distance-bounding phase* using v^1, the prover P and the verifier V label a full binary tree of depth n. The left and right edges of the tree are

labelled with 0 and 1 respectively, while each node of the tree (except of the root node) is assigned a value of a particular bit of v^1 in an one-to-one fashion (starting from the lowest level nodes and moving from left to right, moving up the tree after assigning all the nodes in the current level). The *distance-bounding* phase has n challenge-response exchanges/rounds. At each round i the challenge-response delay Δt_i is measured. The verifier V chooses a random bit c_i, initialises the clock to zero and transmits c_i to P. Then, P answers with the value $r_i = Node(c_1, \ldots, c_i)$. This function gives the value of the label of node in the tree which we would reach from the root by taking the path labelled $c_1, c_2, \ldots, c_i$ on the edges.

Based on the construction and significance of Theorem 3, we construct the following attack on an instantiation of the Avoine and Tchamkerten protocol.

Distance Fraud Attack on the Avoine and Tchamkerten Protocol. Let g be a PRF with parameters as the one needed in the protocol. Let us consider the PRF f constructed from Theorem 3 based on g and the following elements. Let $T^g(x) = g(x)$. Let D be the set of (N_P, N_V) pairs. Let $\sigma(x, N_P, N_V) = v^0 \| T$ where v^0 and T are constant defined below, $\mathsf{correctPad}(x) = \{x \| N_V; N_V \in \{0,1\}^m\}$, and $\mathsf{extract}^{g_x(.)}(N_P, N_V) = N_P$. We have

$$f_x(N_P, N_V) = \begin{cases} v^0 \| T, & \text{if } N_P = x \\ g_x(N_P, N_V), & \text{otherwise} \end{cases}$$

The constant v^0 is an arbitrary admissible value for v^0. The constant T denotes the full binary tree of length n where all the paths constructed by reading-out the labels of the nodes are equal. A tree with such a property is, e.g., the one where all nodes in the same level have the same label (either all 0 or all 1 per one level). By Theorem 3, f is a PRF.

Consider an instantiation of the Avoine and Tchamkerten protocol, where the PRF f is being used. In that case a dishonest, far-away prover which forges N_P to x will always give the correct response without the need to wait for the challenge to arrive. Thus, he would be able to respond earlier and perform successfully a distance-fraud attack. This obviously contradicts the security against distance-fraud attacks that was claimed in Theorem 3.3 of [7], page 11.

4.5 Reid's *et al.* Protocol

In the Reid *et al.* [14] protocol (depicted in Fig. 8, the prover and the verifier that share a secret key x. During the initialisation phase both of them generate random nonces N_P and N_V and exchange them, as well as exchanging their identities. Then both of them generate a session key k as $k := f_x(ID_P \| ID_V \| N_V \| N_P)$ and encrypt the shared key x with the session key k, i.e., $e := \mathcal{E}_k(x)$, where f_x is a PRF instance. One can view k as an ephemeral key. Based on Theorem 3.4 in [7], the assumption needed for the security of this protocol is that $\mathcal{E}$ should be a IND-CPA secure, symmetric encryption. For instance, we can use one-time pad $\mathcal{E}_k(x) = x \oplus k$.

The distance-bounding phase contains n rounds. At each round i the challenge-response delay Δt_i is measured, where $i \in \{1, \ldots, n\}$. The verifier chooses a random challenge c_i and the prover responds with r_i such that:

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

After the end of the distance-bounding phase the verifier checks the responses and verifies that all response times are below a pre-defined threshold.

Based on the construction and significance of Theorem 3, we construct the following attack on an instantiation of Reid *et al.*'s protocol.

Distance-Fraud Attack on Reid's* et al. *Protocol. Let g be a PRF with parameters as the one needed in the protocol. Let us consider the PRF f constructed from Theorem 3 based on g and the following elements. Let $T^g(x) = g(x)$. Let D be the set of $(ID_V \| ID_P \| N_P, N_V)$ tuples. Let $\sigma(x, ID_V, ID_P, N_V, N_P) = x$, $\mathsf{correctPad}(x) = \{ID_V \| ID_P \| N_V \| x\}$, and $\mathsf{extract}^{g_x(\cdot)}(ID_V, ID_P, N_V, N_P) = N_P$. We have

$$f_x(ID_V \| ID_P \| N_V \| N_P) = \begin{cases} x, & \text{if } N_P = x \\ g_x(ID_V \| ID_P \| N_V \| N_P) & \text{otherwise,} \end{cases}$$

By Theorem 3, f is a PRF.

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

Fig. 8. Protocol proposed by Reid *et al.* [14]

Consider an instantiation of the Reid's *et al.* protocol where the PRF f is being used. Also, we assume that the following encryption function instance is employed:

$$\mathcal{E}_k^{new}(x) = \begin{cases} \mathcal{E}_k(x), & \text{if } k \neq x \text{ and } \mathcal{E}_k(x) \neq x \\ x, & \text{if } k = x \\ \mathcal{E}_k(k), & \text{if } \mathcal{E}_k(x) = x \text{ and } k \neq x. \end{cases}$$

Similarly to Theorem 3, we can show that if $\mathcal{E}$ is an IND-CPA secure symmetric encryption, so is $\mathcal{E}^{new}$.

If a far-away dishonest prover indeed chooses N_P to be equal to x, then the responses r_i will always be equal to x_i (for all $i \in \{1, \ldots, n\}$). Thus, a dishonest, far-away prover can perform a successful distance-fraud attack and claim that he is nearer to the verifier than he really is. This obviously contradicts the security against distance-fraud attacks given in Theorem 3.4, in page 13 of [7], solely based on the PRF assumption and Theorem 1, in page 17 of [14].

Another weak PRF leading to a distance-bounding attack is provided in the next example.

Man-in-the-Middle Attack on Reid's* et al.*'s Protocol. We first construct a PRF producing some unforgeable outputs. To this end, we start with a PRF g such that $f_x(u, v) = g_x(u, v) \| g_x(g_x(u, v))$ has parameters as the one needed in the protocol. We define a predicate $V_x(a, b)$ which is true if and only if $g_x(a) = b$. Clearly, $V_x(f_x(u, v))$ holds for all x, u, v. It is easy to see that f is a PRF. Next, we consider the encryption function $\mathcal{E}$ defined by

$$\mathcal{E}_k(y) = \begin{cases} y, & \text{if } V_y(k) \text{ or } V_y(k \oplus y) \\ k \oplus y, & \text{otherwise} \end{cases}$$

We can show that, for k random (an unknown to the adversary), it is hard to forge y such that $V_y(k)$ or $V_y(k \oplus y)$ hold. So, $\mathcal{E}$ if IND-CPA.

Consider now an instantiation of Reid's *et al.* protocol, where f and $\mathcal{E}$ are as constructed. In this instantiation of the protocol, the encryption is such that $\mathcal{E}_k(x) = x$ for all choices of the nonces. The attacker impersonates the verifier to the prover. First, he starts a session in which he inflicts $N_V = 0$. So, in this session, he sends many challenges equal to 0. Like this, he retrieves $\mathcal{E}_k(x)$ from the responses, which is the secret key x.

Note that by changing the encryption so that $\mathcal{E}_k(x) = k$, we can make a distance fraud attack.

4.6 The Swiss-Knife Protocol

In the Swiss-Knife protocol [12] (depicted in Fig. 9), the prover and the verifier share a secret key x. During the initialisation phase both of them generate random nonces N_P and N_V correspondingly and exchange them. Furthermore, both of them generate a session key a as: $a := f_x(cte, N_P)$, where cte denotes a constant and two values Z^0 and Z^1 such that: $\begin{cases} Z^0 := a \\ Z^1 := a \oplus x \end{cases}$

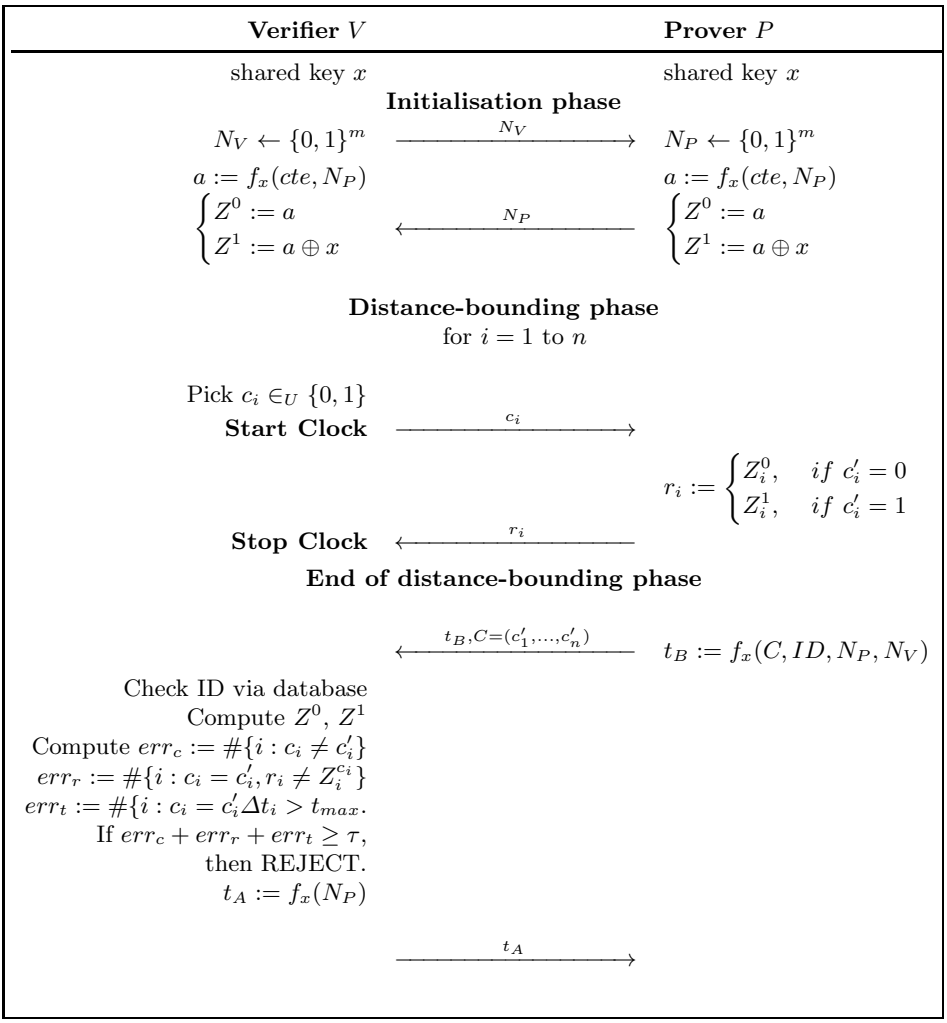

Fig. 9. Swiss-Knife protocol [12]

In the distance bounding phase which is repeated n times the verifier selects a random challenge c_i where $i \in \{1 \dots n\}$ and the prover responds with r_i such that:

$$r_i := \begin{cases} Z_i^0, & if\ c_i' = 0 \\ Z_i^1, & if\ c_i' = 1, \end{cases}$$

where c_i' is the challenge that the prover actually received in the i-th round, i.e., c_i' will be c_i itself, if the transmission was correct, or c_i' will be $\overline{c_i}$, if c_i was perturbed by noise. After the end of the distance bounding phase the prover transmits a message t_B such that: $t_B := f_x(C, ID, N_P, N_V)$ where $C = c_1', \dots, c_n'$.

Based on the construction and significance of Theorem 3, we construct the following attack on an instantiation of the Swiss-Knife protocol.

Man-in-the-Middle Attack on the Swiss-Knife Protocol. Let g be a PRF with parameters as the one needed in the protocol such that the function truncating the the leading half is hard-core. Let us consider the PRF f constructed from Theorem 3 based on g and the following elements. Let $T^g(x) = g(x)$. Let D be the set of (C, ID, N_P, N_V) tuples. Let $\sigma(x, C, ID, N_P, N_V) = x$, $\mathsf{correctPad}(x) = \{C \| ID \| N_P \| N_V; C = 1^{\frac{m}{2}} \| \mathsf{msb}_{\frac{m}{2}}(g_x(cte, N_P) \oplus x)\}$, and $\mathsf{extract}^{g_x(.)}(C, ID, N_P, N_V) = \mathsf{lsb}_{\frac{m}{2}}(C) \oplus \mathsf{msb}_{\frac{m}{2}}(g_x(cte, N_P))$. (Note that $\mathsf{extract}$ only recovers the leading half of x so it is not exactly compatible with the assumptions of Theorem 3.) We have

$$f_x(cte, N_P) = g_x(cte, N_P),$$
$$f_x(C, ID, N_P, N_V) = \begin{cases} x, & \text{if } C = 1^{\frac{m}{2}} \| \mathsf{msb}_{\frac{m}{2}}(g_x(cte, N_P) \oplus x) \\ g_x(C, ID, N_P, N_V), & \text{otherwise,} \end{cases}$$
$$f_x(N_P) = g_x(N_P).$$

By Theorem 3, f is a PRF. Indeed, since $\mathsf{extract}$ only recovers half of x, we need another trick in the proof of Theorem 3 to show that $\Pr[F]$ is negligible.

Consider an instantiation of the Swiss-Knife protocol where the PRF f is being used. Then, an adversary can extract the key and conduct successfully an impersonation attack. Namely, he can query c_i=1 for $i \in \{1, \dots, \frac{n}{2}\}$ and $c_i = r_{i-\frac{n}{2}}$ for $i = \{\frac{n}{2} + 1, \dots, n\}$. Given the error-tolerance of the protocol, we can presume that the adversary is powerful enough to make the communication noiseless and thus the prover will respond to this very challenges, and, due to the shape of f_x, the adversary will learn x out of his strategy.

The attack exhibited does therefore disprove the claim of MiM security in Theorem 3.5 in [7] based solely on the PRF assumption and the achievement of authentication claimed in [12]. Moreover, it appears [12] that Swiss-Knife was aimed to resist MiM attacks.

5 Conclusions

In this paper, we gave two constructions of PRFs with trapdoors by PRF programming, assuming that PRFs exist. These constructions respectively prompt

to distance-frauds and MiM attacks in DB protocol. In fact, we presented such attacks on important DB protocols, thus disproving different security claims or proofs that appeared in the literature. The latter claims were relying on the PRF assumption for families of function used inside these DB protocol. Our results show that such an assumption is then not enough for the security of DB protocols.

As future work, we will prove how to restore security by additional tricks. Distance fraud security can be achieved by key-masking, i.e., by using $f_x(\cdot) \oplus M$ for a random M instead of $f_x(\cdot)$. MiM security can be restored by introducing an extra security notion to the PRF, so that using $f_x(\cdot) \oplus x$ is still safe.

Acknowledgements. The authors acknowledge the support of the Marie Curie IEF project "PPIDR: Privacy-Preserving Intrusion Detection and Response in Wireless Communications", grant number: 252323, and of the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), under the Swiss National Science Foundation.

References

1. Avoine, G., Lauradoux, C., Martin, B.: How Secret-sharing can Defeat Terrorist Fraud. In: Proceedings of the 4th ACM Conference on Wireless Network Security – WiSec 2011, Hamburg, Germany. ACM, ACM Press (June 2011)
2. Avoine, G., Tchamkerten, A.: An Efficient Distance Bounding RFID Authentication Protocol: Balancing False-Acceptance Rate and Memory Requirement. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C.A. (eds.) ISC 2009. LNCS, vol. 5735, pp. 250–261. Springer, Heidelberg (2009)
3. Brands, S., Chaum, D.: Distance Bounding Protocols (Extended Abstract). In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 344–359. Springer, Heidelberg (1994)
4. Cremers, C., Rasmussen, K.B., Čapkun, S.: Distance hijacking attacks on distance bounding protocols. Cryptology ePrint Archive, Report 2011/129 (2011), `http://eprint.iacr.org/`
5. Drimer, S., Murdoch, S.J.: Keep your enemies close: distance bounding against smartcard relay attacks. In: Proceedings of the 16th USENIX Security Symposium on USENIX Security Symposium, pp. 7:1–7:16. USENIX Association, Berkeley (2007)
6. Dürholz, U., Fischlin, M., Kasper, M., Onete, C.: A Formal Approach to Distance-Bounding RFID Protocols. In: Lai, X., Zhou, J., Li, H. (eds.) ISC 2011. LNCS, vol. 7001, pp. 47–62. Springer, Heidelberg (2011)
7. Fischlin, M., Onete, C.: Provably secure distance-bounding: an analysis of prominent protocols. Cryptology ePrint Archive, Report 2012/128 (2012)
8. Ford. Safe and Secure *SecuriCode*$^{\text{TM}}$ Keyless Entry (2011), `http://www.ford.com/technology/`
9. Hancke, G.P., Kuhn, M.G.: An RFID Distance Bounding Protocol. In: Proceedings of SECURECOMM, pp. 67–73 (2005)
10. Kapoor, G., Zhou, W., Piramuthu, S.: Distance Bounding Protocol for Multiple RFID Tag Authentication. In: Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, EUC 2008, vol. 02, pp. 115–120. IEEE, IEEE Computer Society, Shanghai, China (2008)

11. Kim, C.H., Avoine, G.: RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 119–133. Springer, Heidelberg (2009)
12. Kim, C.H., Avoine, G., Koeune, F., Standaert, F.-X., Pereira, O.: The Swiss-Knife RFID Distance Bounding Protocol. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 98–115. Springer, Heidelberg (2009)
13. Nielsen, J.B.: A Threshold Pseudorandom Function Construction and Its Applications. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 401–416. Springer, Heidelberg (2002)
14. Reid, J., Gonzalez Nieto, J.M., Tang, T., Senadji, B.: Detecting Relay Attacks with Timing-based Protocols. In: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, ASIACCS 2007, pp. 204–213. ACM, Singapore (March 2007)
15. Shoup, V.: Sequences of Games: a Tool for Taming Complexity in Security Proofs (2006) (manuscript)
16. Tu, Y.-J., Piramuthu, S.: RFID Distance Bounding Protocols. In: Proceedings of the First International EURASIP Workshop on RFID Technology (2007)