Survey of Distance Bounding Protocols and Threats

Agnès Brelurut[1], David Gerault[1], and Pascal Lafourcade[1] [⋆]

University Clermont Auvergne, LIMOS, France

Abstract. NFC and RFID are technologies that are more and more present in our life. These technologies allow a tag to communicate without contact with a reader. In wireless communication an intruder can always listen and forward a signal, so he can mount a so-called *worm hole* attack. In the last decades, several Distance Bounding (DB) protocols have been introduced to avoid such attacks. In this context, there exist several threat models: Terrorist Fraud, Mafia Fraud, Distance Fraud etc. We first show the links between the existing threat models. Then we list more than forty DB protocols and give the bounds of the best known attacks for different threat models. In some cases, we explain how we are able to improve existing attacks. Then, we present some advices to the designers of the DB protocols and to the intruders to mount some attacks.

Keywords: Distance Bounding, Threat Models, Mafia Fraud, Terrorist Fraud, Distance Fraud, RFID, NFC, Relay Attack, Collusion Fraud.

1 Introduction

Nowadays, Radio Frequency IDentification (RFID) and Near Field Communication (NFC) technologies and more generally the wireless technologies are increasingly developped. They are commonly used in payments, access-control applications and even in many electronic passports [23]. The main purpose of these technologies is to allow a *reader* (or *verifier*) to communicate wirelessly with *tags* (or *provers*) implanted into objects. In this context, an intruder can simply mount relay attacks just by forwarding some signal and then fake the reader by using the signal of a tag that can be far-away. To avoid such attacks, Distance Bounding (DB) protocols were introduced by Brands and Chaum in 1993 [14]. They are a countermeasure against relay attacks since they measure the round-trip delays during a rapid phase of challenge-response. DB protocols check that provers are close to the verifier in the trusted zone. In the literature, there exist several threat models according to the power and the aim of the intruder:

Distance Fraud [14]: a far-away malicious prover tries to convince the verifier that they are close, while the prover is not in the trusted zone. A practical example is house arrest, where a convict wearing an electronic bracelet is forbidden to leave a given area. If he can mount a distance fraud against the electronic surveillance device, then he can pretend he is within allowed area even though he is far away.

[⋆] This research was conducted with the support of the “Digital trust” Chair from the University of Auvergne Foundation.

Mafia Fraud (MF) [18]: an adversary between a far-away honest prover and a verifier tries to get advantage of his position to authenticate the prover close to the verifier. The adversary can simply relay the message, but he may also modify the messages involved or create new messages (to the prover or to the verifier). To illustrate this attack, imagine a waiting line in which an attacker would relay the signal between the payment card of a custommer who is at the end of the line and the payment terminal. This allows him to make someone else pay for him.

Terrorist Fraud (TF) [18]: a far-away malicious prover, helped by an adversary, tries to convince the verifier that they are close. In fact, the adversary is close to the verifier and the prover gives information to the adversary, but the adversary cannot impersonate the prover during a further protocol execution. An example is someone wanting to let a friend open his locker once, but not willing to allow him to do it later. In other words, he is willing to provide help only if this help can not be used by the friend to authenticate again in the future.

Impersonation Fraud (IF) [5]: an adversary tries to impersonate the prover to the verifier. The aim can be for instance to make someone else blamed of one's bad actions.

Distance Hijacking (DH) [17]: a far-away prover takes advantage of some honest, active provers (to which one is close) to make the verifier grants privileges for the far-away prover. It can be used to forge an alibi.

Contributions: We first explain the relations between the different threats by distinguishing, on one hand, the threats where the prover is dishonest and, on the other hand, the threats where the prover is honest. Then, we present a survey of DB protocols and for each one we give the success probabilities of the best know attacks. For several protocols we were able to improve some attacks. Our list of protocol contains 42 protocols from 1993 up to 2015. We also present more than 24 attack improvements. Finally, we compile the main attack strategies discovered over the years and list some advices to the designer of DB protocols.

Related Work: Distance Bounding was introduced by Brands and Chaum in 1993 [14] to combat relay attacks. They also introduce the attack that we call *distance fraud*: their protocol prevents to the response bits which are sent out too soon. Before the existence of distance bounding protocols, in 1988, Desmedt identifies the *terrorist fraud* and *mafia fraud* [18]. Then, Avoine & Tchamkerten in 2009 [5] study the *impersonation fraud*. In 2012, Cremers *et al.* find an attack that they called *distance hijacking* in [17]. At the same time, some of them lay the groundwork for formally modelling DB protocols [3, 19].

Avoine *et al.* proposed a formal framework for cryptanalyzing the DB protocols in [3]. In particular, they defined the adversary strategies for *mafia* and *terrorist fraud* which they called *no-ask*, *pre-ask* and *post-ask* strategy. Two years later, Dürholz *et al.* proposed in [19] the first computational formal framework for providing properties of DB protocols that are based on shared symmetric keys. They give rigorous cryptographic security models for *mafia, terrorist,* and *distance fraud*. The BMV model proposed by Boureanu, Mitrokotsa and Vaudenay in [9] generalizes the previous fraud into three group of threats: *distance fraud*, *Man-In-the-Middle* and *collusion fraud*. By their definitions, the *distance fraud* includes *distance hijacking* and the previous *distance fraud*,

the *Man-In-the-Middle* contains the *mafia fraud* and the *impersonation fraud*, and, the *collusion fraud* extends the notion of *terrorist fraud*. But they do not establish relations between these three different threat models.
Some papers [26, 22, 12, 11, 10, 13] compare from four to fourteen protocols to their success probabilities for *distance fraud, mafia fraud, terrorist fraud* and/or *impersonation fraud* attack. Distance Bounding was studied in the context of RFID but also in ad-hoc networks as in the survey proposed by Meghdadi *et al.* [40].

Outline : In Section 2, we show the relationship between different threat models. Then in Section 3, we list existing DB protocols and the success probability of attacks against them. Finally before concluding, we give some advices for designing DB protocols and also strategies to mount some attacks in Section 4.

2 Relations between Threats for DB Protocols

In [9], Boureanu *et al.* propose a framework, denoted BMV model, that generalizes the definitions of the previously enumerated common frauds into three possible threats: **distance fraud, man-in-the-middle** and **collusion fraud**.

We present the formal definitions given in [9], then we show how these defintions cover usual threats models and prove some relations between some of these notions.

2.1 Threat Models of [9]

The BMV model [9] offers formal definition about DB protocols and their three threats. In the following, provers are denoted by P, verifiers by V, the adversary by $\mathcal{A}$, and P^* denotes dishonest provers. Provers do not have output, and verifiers have one bit output Out_V, where $Out_V = 1$ denotes acceptance and $Out_V = 0$ denotes rejection.

Definition 1 (Distance-Bounding Protocols [9]). *A* Distance Bounding (DB) protocol *is defined by a tuple* $(Gen, P, V, \mathbb{B})$*, where:*

1. *Gen is randomised, key-generation algorithm such that $Gen(1^s; r_k) \mapsto (x, y)$, where r_k are random coins of Gen and s is a security parameter;*
2. *$P(x; r_P)$ is a ppt. ITM [1] running the algorithm of the prover with input x and random input r_P;*
3. *$V(y; r_V)$ is a pp. ITM running the algorithm of the verifier with input y, and random input r_V;*
4. *$\mathbb{B}$ is a distance-bound.*

They must be such that the following two properties hold, where we denote by $d(loc_V, loc_P)$ the distance between the localisation of V and P:

- ***Termination:*** *$(\forall s)(\forall \mathbf{R})(\forall r_k, r_V)$ if $(., y) \to Gen(1^s; r_k)$ and $\mathbf{R} \leftrightarrow V(y; r_V)$ model the execution, it is the case that V halts in $Poly(s)$ computational steps, where $\mathbf{R}$ is any set of (unbounded) algorithms;*

[1] ppt. ITM is for polynomial probabilistic time Interactive Turing Machine

- *p-completeness:*$(\forall s)(\forall loc_V, loc_P$ *such that* $d(loc_V, loc_P) \leq \mathbb{B})$ *we have:*

$$\underset{r_k, r_P, r_V}{Pr}\left[Out_V = 1 : \begin{array}{c}(x,y) \leftarrow Gen(1^s; r_k)\\ P(x, r_P) \leftrightarrow V(y; r_V)\end{array}\right] \geq p$$

Throughout, "$Pr_r[\text{event: experiment}]$*" denotes the probability that an event takes place after the experiment has happened, taken on the set of random coins* r *underlying the experiment. The random variable associated to the event is defined via the experiment leading to the description of a random variable*

This model implicitly assumes *concurrency* involving participants that do not share the secret inputs amongst them. In the rest of the paper, $\alpha, \beta, \gamma, \gamma' \in [0; 1]$ and the $View$ of a participant on an experiment is the collection of all its initial inputs (including coins) and his incoming messages.

Distance Fraud (DF) [9]: it corresponds to the classical notion, but concurrent runs with many participants are additionally considered, *i.e.*, it includes other possible provers (with other secrets) and verifiers. Consequently, this generalized distance fraud also includes distance hijacking.

Definition 2 **(α-resistance to DF [9]).** *A protocol is* α*-resistant to DF if:* $\forall s, \forall P^*, \forall loc_V$ *such that* $d(loc_V, loc_{P^*}) > \mathbb{B}$*, and* $\forall r_k$*, we have:*

$$\underset{r_V}{Pr}\left[Out_V = 1 : \begin{array}{c}(x,y) \leftarrow Gen(1^s; r_k)\\ P^*(x) \leftrightarrow V(y; r_V)\end{array}\right] \leq \alpha$$

where P^* *is any (unbounded) dishonest prover. In a concurrent setting, a polynomially bounded number of honest* $P(x')$ *and* $V(y')$ *close to* $V(y)$ *with independent* (x', y') *are implicitely allowed.*

In others words, a protocol is α-resistance to DF if a far-away prover cannot be authenticated by a verifier with probability more than α.

Man-In-the-Middle (MiM) [9]: this formalization considers an adversary that works in two phases. During a *learning phase*, this adversary interacts with many honest provers and verifiers. Then, the *attack phase* implies a far-away honest prover of given ID and possibly many other honest provers and other verifiers. The goal of the adversary is to make the verifier accept in a session with ID. Clearly, this generalizes the mafia fraud and includes impersonation fraud.

Definition 3 **(β-resistance to MiM [9]).** *A protocol is* β*-resistant to MiM attack if:* $\forall s, \forall m, l, z$ *polynomially bounded,* $\forall \mathcal{A}_1, \mathcal{A}_2$ *polynomially bounded, for all locations such that* $d(loc_{P_j}, loc_V) > \mathbb{B}$*, where* $j \in \{m+1, ..., l\}$ *we have:*

$$Pr\left[Out_V = 1 : \begin{array}{c}(x,y) \leftarrow Gen(1^s)\\ P_1(x), ..., P_m(x) \leftrightarrow \mathcal{A}_1 \leftrightarrow V_1(y), ..., V_z(y)\\ P_{m+1}(x), ..., P_l(x) \leftrightarrow \mathcal{A}_2(View_{\mathcal{A}_1}) \leftrightarrow V(y)\end{array}\right] \leq \beta$$

over all random coins, where $View_{\mathcal{A}_1}$ *is the final view of* $\mathcal{A}_1$*. In a concurrent setting, a polynomially bounded number of* $P(x')$*,* $P^*(x')$ *and* $V(y')$ *with independent* (x', y')*, is implicitely allowed anywhere.*

A protocol is β-resistant to MiM, if the probability that an adversary authenticates a far-away prover to a verifier is at most β even if the adversary has access to information of a first session run between provers close to verifiers.

Definition 3 separates a *learning phase* (with the adversarial behaviour $\mathcal{A}_1$) from an *attack phase* (with the adversarial behaviour $\mathcal{A}_2$). This definition models a practical setting where an attacker would have cloned several tags (provers) and would make them interact with several readers (verifiers) with which they are registered. From such a multi-party communication, the attacker can get potentially more benefits, in a shorter period of time. To increase his gain, the attacker can set up the learning phase as he pleases (otherwise the learning phase is not obligatory). So, the attacker can place prover-tags close to verifier-readers, even if being an active adversary between two neighbouring P and V is technically more challenging than interfering between two far-away parties.

Collusion Fraud (CF) [9]: this fraud considers a far-away prover holding a secret x who helps an adversary to make the verifier accept. This might be in the presence of many other honest participants. However, there should be no man-in-the-middle attack based on this malicious prover, i.e., the adversary should not extract any advantage from this prover to run (later) a man-in-the-middle attack.

Definition 4 **((γ, γ')-resistance to CF [9]).** *A protocol is (γ, γ')-resistant to CF if: $\forall s, \forall P^*, \forall loc_{V_0}$ such that $d(loc_{V_0}, loc_{P^*}) > \mathbb{B}$) and $\forall \mathcal{A}^{CF}$ppt. such that:*

$$Pr\left[Out_{V_0} = 1 : \begin{array}{l} (x, y) \leftarrow Gen(1^s) \\ P^*(x) \leftrightarrow \mathcal{A}^{CF} \leftrightarrow V_0(y) \end{array}\right] \geq \gamma$$

over all random coins, there exists a (kind of) [2] MiM attack $m, l, \mathcal{A}_1, \mathcal{A}_2, P_i, P_j, V_{i'}$ using P and P^* in the learning phase, such that:

$$Pr\left[Out_V = 1 : \begin{array}{l} (x, y) \leftarrow Gen(1^s) \\ P_1^{(*)}(x), ..., P_m^{(*)} \leftrightarrow \mathcal{A}_1 \leftrightarrow V_1(y), ..., V_z(y) \\ P_{m+1}(x), ..., P_l(x) \leftrightarrow \mathcal{A}_2(View_{\mathcal{A}_1}) \leftrightarrow V(y) \end{array}\right] \geq \gamma'$$

where P^* is any (unbounded) dishonest prover and $P^{(*)}$ runs either P or P^*. Following the MiM requirements, $d(loc_{P_j}, loc_V) > \mathbb{B}$, for all $j \in \{m+1, ..., l\}$. In a concurrent setting, a polynomially bounded number of $P(x')$, $P^*(x')$ and $V(y')$ with independent (x', y') is implicitely allowed, but no honest participant close to V_0.

In others words, a protocol is (γ, γ')-resistant to CF, if when an adversary manages to authenticate a far-away prover to a verifier with probability at least γ, then there exists a further MiM attack 2, where an adversary manages to authenticate a far-away prover to the verifier with probability at least γ'.

2.2 Relationship between Different Threat Models

We prove some relations between some of these properties. In the rest of this section, for a given protocol, $X \rightarrow Y$ denotes that if the property X is satisfied then Y is also

[2] Def 3 defines MiM attack as using a honest $P(x)$. Here, the definition use $P^*(x)$.

satisfied, which is equivalent to say that if there exists an attack on the property Y then there exists an attack on the property X. For a given protocol, we also denote by $X \dashrightarrow Y$ the fact that if there exists an attack on the property Y without sending the secret x then there exists an attack on the property X. For classical threat models, we explain how they can be defined using this formal model.

Theorem 1 (DF → DH [9]). *If a protocol is α-resistant to DF then it is also α-resistant to DH.*

Proof. Distance Hijacking is included in the definition of resistance of Definition 2. An experiment in which a dishonest far-away prover P^* may use several provers to get authenticated as one, honest P that is close to the verifier is clearly included in the concurrent setting.

$$\begin{array}{l} (x, y) \leftarrow Gen(1^s) \\ P^*(x) \leftrightarrow P_1(x'), .., P_n(x') \leftrightarrow V(y) \end{array}$$

□

Theorem 2 (MiM → MF [9] and MiM → IF [9]). *If a protocol is β-resistant to MiM, then it is β-resistant to MF and β-resistant to IF.*

Proof. In Definition 3, the classical notion of mafia fraud corresponds to $m = z = 0$ and $l = 1$. The experiment performing this attack can be described as follows:

$$\begin{array}{l} (x, y) \leftarrow Gen(1^s) \\ P(x) \leftrightarrow \mathcal{A} \leftrightarrow V(y) \end{array}$$

In Definition 3, the classical notion of impersonation corresponds to $l = m$, *i.e.*, there is no prover in the attack phase. The experiment corresponding is:

$$\begin{array}{l} (x, y) \leftarrow Gen(1^s) \\ P_1(x), ..., P_m(x) \leftrightarrow \mathcal{A}_1 \leftrightarrow V_1(y), ..., V_z(y) \\ \mathcal{A}_2(View_{\mathcal{A}_1}) \leftrightarrow V(y) \end{array}$$

□

Theorem 3 (CF → TF [9]). *If a protocol is (γ, γ')-resistant to CF, then it is (γ, γ')-resistant to TF.*

Proof. The notion of terrorist fraud is connected with the definition 4: it is sufficient to take $m = z = 1$, $l = 2$ and $\mathcal{A}_1$ just runs $\mathcal{A}^{\mathrm{CF}}$ in the learning phase, *i.e.*, $\mathcal{A}^{\mathrm{CF}}$ gets information to directly impersonate the prover. We can model this fraud by:

$$\begin{array}{l} (x, y) \leftarrow Gen(1^s) \\ P^*(x) \leftrightarrow \mathcal{A}_1 \leftrightarrow V(y) \\ P(x) \leftrightarrow \mathcal{A}_2(View_{\mathcal{A}_1}) \leftrightarrow V(y) \end{array}$$

□

Theorem 4 (TF $\dashrightarrow$ DF). *If a protocol is not α-resistant to DF, then there exists an attack of kind TF which succeed with probability at least α.*

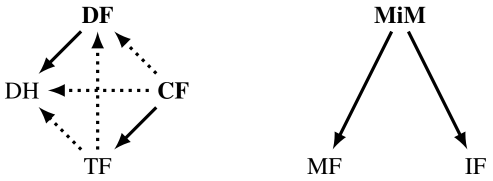

Fig. 1. Relations between different Threats Models.

Proof. We assume that there exists an attack A of type DF such that: Pr$[A \text{ succeed}] = \alpha$ $\Leftrightarrow$ Pr $\left[Out_V = 1 : \begin{array}{c} (x,y) \leftarrow Gen(1^s; r_k) \\ P^*(x) \leftrightarrow V(y; r_V) \end{array} \right] = \alpha$, where α is not negligible. Then we can elaborate an attack of type TF, only if P^* does not transmit his secret in clear, *i.e.*:

$$\text{Pr} \left[Out_{V_0} = 1 : \begin{array}{c} (x,y) \leftarrow Gen(1^s) \\ P^*(x) \leftrightarrow \mathcal{A}^{\text{TF}} \leftrightarrow V_0(y) \end{array} \right] \geq \alpha$$

with $\mathcal{A}^{\text{TF}}$ who simply relays messages and P^* plays the same role as before. But if P^* and $\mathcal{A}^{\text{TF}}$ cooperate, it may be that the attack has a better success probability. □

In Figure 1, we summarize all these relations. On the left there are attacks where the prover is far away and dishonnest, and on the right there are the attacks where the prover is close to the verifier. The arrow from TF (or CF) to DH is obtained by transitivity as well as the arrow from CF to DF. Once an attack is discovered against a property, it is easily to extend it to other properties using these results.

3 Survey

Our aim is to list the utmost number of protocols[3] in order to understand their special features. Table 1 references the success probability of the best known attacks in the literature. The color red highlights the improvements we discovered for some protocols. We do not consider DH threat model in our study since only few papers study this property [17] and mounting such attacks is difficult. We do not recall the description of the protocols for obvious reasons, but for each protocol where we propose one improvement we use the same notations as the ones used in the original paper. However, each DB protocol usually follows this form: one initialization phase where the participant generally share some secret data (often denoted x) or public information often denoted by N_V and N_P respectively for the verifier and the prover. Some encryption, hash function or *Pseudo-Random Function* (PRF) are also often used. Then a fast challenge response phase where usually a sequence of n challenges c_i are sent by the verifier to the prover who answers by some responses denoted r_i. Then the last phase of verification or authentication usually consists in opening a commitment or verifying a common shared data called *transcript*. We present in Figure 2 the general structure of a DB protocol. In the rest of this section, we explain the new attacks we discovered on some protocols, then the improvements we did for protocols that are using *Pseudo-Random Function* (PRF) in a non appropriated way.

[3] Most of the papers are avaible at `http://www.avoine.net/rfid/`

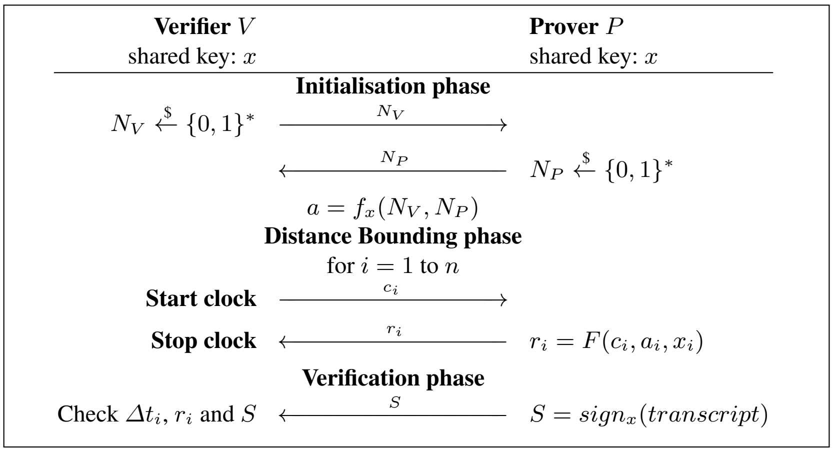

Fig. 2. The general structure of DB protocol.

3.1 Improvements of attacks

For [5, 49, 7, 50, 56, 34, 38, 41, 31, 51, 20], we mount new TF attacks. These attacks all follow the same scheme: the adversary is close to the verifier whereas the prover is far away. During the initialisation phase, the adversary simply relays the messages from the prover to the verifier and the messages from the verifier to the prover. After this phase, the prover computes responses, with the help of his key (the adversary does not have access to this key and the prover does not want him to obtain it). Then the prover sends his responses to his partner (the adversary), so that the adversary can answer to the verifier's challenges in the fast challenge-response phase. The prover sends all the results of these computations because he knows that the adversary cannot recover the shared secret key even if he has access to all results (except for the protocols [28, 31] where the adversary receives $n - v$ bits of the secret key). For the last phase, the adversary sends all he receives from the verifier to the prover and so the prover can close the session by sending his signature of the transcript. Using the result about the relation between the threat models we immediately deduce attacks for CF.

We can also see in Table 1 that the column IF is mainly filled with $\left(\frac{1}{2}\right)^s$ (where s is the size of the key). This correspond to the exhaustive research on the key, which is the simplest attack. Most of this column is in red is due to the fact that this threat model is not considered in many papers.

As we can see in Table 2, many of the listed protocols use a PRF [29, 47, 33, 42, 44, 5, 37, 36, 7, 45, 50, 4, 56, 28, 34, 55, 38, 41, 31, 12, 25, 51, 20–22]. It is possible to mount some attacks if the PRF used follows a certain form. This kind of attacks is first introduced in [8]. By using the idea of [8], we improve some attacks on DF on [42, 36, 7, 50, 34, 38, 51, 22, 20, 21] and one on MiM on [56]. We detail only one attack on DF and the one on MiM, but for the other protocols we give the PRF construction to mount a successful attack. To succeed during a DF attack, it is necessary for the prover to send his response before receiving the verifier's challenge during the fast challenge-response phase. But, when the PRF's output is split into several parts to precompute responses, using a special PRF the dishonest prover is able to make the response independant of the challenge recieved. We show in Figure 3 this kind of attack. The PRF f is based on the other PRF g as follows, where z is a special value known by P also called trapdoor:

$$f_x(N_V, N_P) = \begin{cases} a||a \text{ if } N_P = z \\ g_x(N_V, N_P) \text{ otherwise} \end{cases}$$

So this attack succeeds with probability 1.

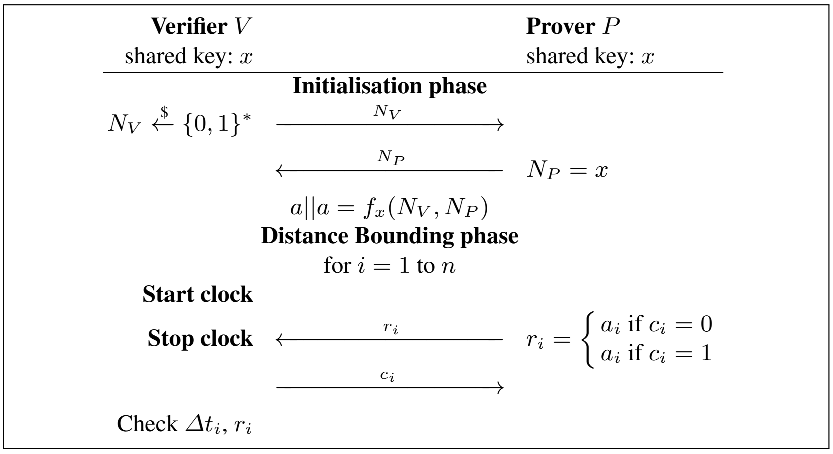

Fig. 3. *Distance fraud* attack against DB protocol using PRF.

Munilla & Peinado [42] - DF: let g be a PRF. Let us consider the PRF $hash$ is constructed as follows: $hash(K, N_a, N_b) = \begin{cases} P||v||v & \text{if } N_b = z \\ g(K, N_a, N_b) & \text{otherwise} \end{cases}$
Consider an instantiation of the Munilla & Peinado protocol where $hash$ is used. In this instance a far-away malicious prover P^* could easily perform a distance fraud attack. By picking N_b equal to z, , he can send the same response r_i regardless of the verifier's challenge c_i . Then, in agreement with this protocol scheme [42] (void challenge-response if $P_i = 1$) if $P_i = 0$ for any challenge c_i the response is the i-th bit of v. If $P_i = 1$, P^* waits a delay before sending the next response r_{i+1}. Thus, if the malicious prover applies this strategy he can defeat the distance-bound.

Kim & Avoine [36] - DF: let g be a PRF. The PRF h to perform the DF attack is as follows: $h(K, N_a, N_b) = \begin{cases} T||D||v||v & \text{if } N_b = z \\ g(K, N_a, N_b) & \text{otherwise} \end{cases}$

Benfarah *et al.* [7] - DF: let g be a PRF. We use the PRF f as follows:
$f(k, N_P, N_V) = \begin{cases} S^V||S^P||R||R \text{ if } N_P = z \\ g(k, N_P, N_V) \text{ otherwise} \end{cases}$

Poulidor [50] - DF: let g be a PRF. We use the following PRF:
$PRF(x, N_P, N_V) = \begin{cases} 1^{2n}||H^{2n} & \text{if } N_P = z \\ g(x, N_P, N_V) & \text{otherwise} \end{cases}$ where 1^{2n} denotes the number constituted of $2n$ bits equal to 1 and H^{2n} denotes a number of $2n$ random bits.

Kardas *et al.* [34] - DF : we use two PRF f_{K_i} and f_{L_i} based on an other PRF g:

$$T = f_{K_i}(r_P, r_V) = \begin{cases} x & \text{if } r_P = z \\ g_{K_i}(r_P, r_V) & \text{otherwise} \end{cases} \text{ and } f_{L_i}(T) = \begin{cases} v_1||v||v \text{ if } T = x \\ g_{L_i}(T) \ \text{ otherwise} \end{cases}$$

Lee *et al.* [38] - DF: let g be a PRF. The PRF f based on g is as follows:

$$f(K, N_V, N_P) = \begin{cases} d||d||v||v & \text{if } N_P = z \\ g(K, N_V, N_P) & \text{otherwise} \end{cases}$$

TMA [51] - DF: let g be a PRF. We consider the following PRF based on g:

$PRF(x, N_P, N_V) = \begin{cases} 0^n||R||R & \text{if } N_P = z \\ g(x, N_P, N_V) & \text{otherwise} \end{cases}$ where 0^n denotes the number constituted of n bits equal to 0

Baghernejad *et al.* [22] - DF: let g be a PRF. We use the PRF F as follows:

$$F(K, N_R, N_T) = \begin{cases} D||a||a & \text{if } N_T = z \\ g(K, N_R, N_T) & \text{otherwise} \end{cases}$$

EBT [20] - DF: let g be a PRF. We use the PRF h as follows:

$$h(x, N_V, N_P) = \begin{cases} d||d||d||d||v||v \text{ if } N_P = z \\ g(x, N_V, N_P) \ \text{ otherwise} \end{cases}$$

Falahati *et al.* [21] - DF: let g be a PRF. We use the PRF F as follows:

$$F(K, N_U, N_V) = \begin{cases} \text{D a tree with same bit at each level} & \text{if } N_U = z \\ g(K, N_U, N_V) & \text{otherwise} \end{cases}$$

$$F(D, K, C_1, C_2, \ldots, C_i) = \begin{cases} \text{a tree with same bit at} & \text{if D is a tree with} \\ \text{each level} & \text{same bit at each level} \\ g(D, K, C_1, C_2, \ldots, C_i) & \text{otherwise} \end{cases}$$

A MiM attack is successfull when the Man-in-the-Middle adversary recovers the prover's key, or manages to make the verifier accept the authentication of the prover even if the prover is far-away.

Yum *et al.* [56] - MiM: Consider an instantiation of the Yum *et al.* protocol where $\tau = 1$ and K denotes the shared key. As $\tau = 1$, we have $D = f_\tau(K, N_V, N_P) = 0^n$. So for each round the prover waits a verifier's challenge. Let g be the PRF used to compute $v = g(K, N_V, N_P)$. Let PRF be a PRF and h the PRF based on PRF as follows:

$$Z = h(K, N_V, N_P, C, R) = \begin{cases} K & \text{if } C = 0^{\frac{n}{2}}||v^{\frac{n}{2}} \\ PRF(K, N_V, N_P, C, R) & \text{otherwise} \end{cases}$$

$0^{\frac{n}{2}}$ is the number composed of $\frac{n}{2}$ bits equal to 0 and $v^{\frac{n}{2}} = \mathbf{msb}_{\frac{n}{2}}(f(K, N_V, N_P))$, where $\mathbf{msb}(x)$ denotes the most significant bits of x. The adversary aims to recover the key from the prover, in order to impersonate the prover at any moment. The attacker impersonates the verifier to the prover, he sends an arbitrary N_V to the prover and receives the prover's nonce N_P. The prover computes v. Then the rapid phase of challenge-response begins: for the $\frac{n}{2}$ first rounds, the adversary sends $c_i = 0$, and so he receives the $\frac{n}{2}$ most significant bits ($\mathbf{msb}$) of v. For the other half of rounds, the adversary sends one by one the received bits. Then, the prover creates Z and $C = c_1||..||c_n$ such that $C = 0^{\frac{n}{2}}||v^{\frac{n}{2}}$, so in fact by sending Z, the prover sends to the adversary his own secret key K. The attacker is able to impersonate the prover for further executions of the protocol.

3.2 Comparison of DB protocols

Our survey highlights some points: First, very few protocols are strong against all frauds, only nine protocols insure the security against all kinds of attacks. They are in bold in Tables 1 and 2, and are the following: KZP (2008) [33], Hitomi (2010) [45], NUS (2011) [28], SKI$_{\text{pro}}$ (2013) [9], FO (2013) [25], DB1 (2014) [12], DB2 (2014) [12], ProProx (2014) [53] and VSSDB (2014) [26]. The security level for *impersonation fraud* are the same for all these protocols and it is the best security level, *i.e.*, it is equivalent at the security against brute force. Proprox [53] has the best security level against *distance fraud, mafia fraud* and *terrorist fraud* and he is also the most secure against all frauds.

The graph of dependency of protocols, presented in Figure 4, shows the descendants of some protocols and reveals six families including two large ones (one composed by the descendants of Brands & Chaum and Hancke & Kuhn, the other one by the descendants of Swiss-Knife and SKI$_{\text{pro}}$).

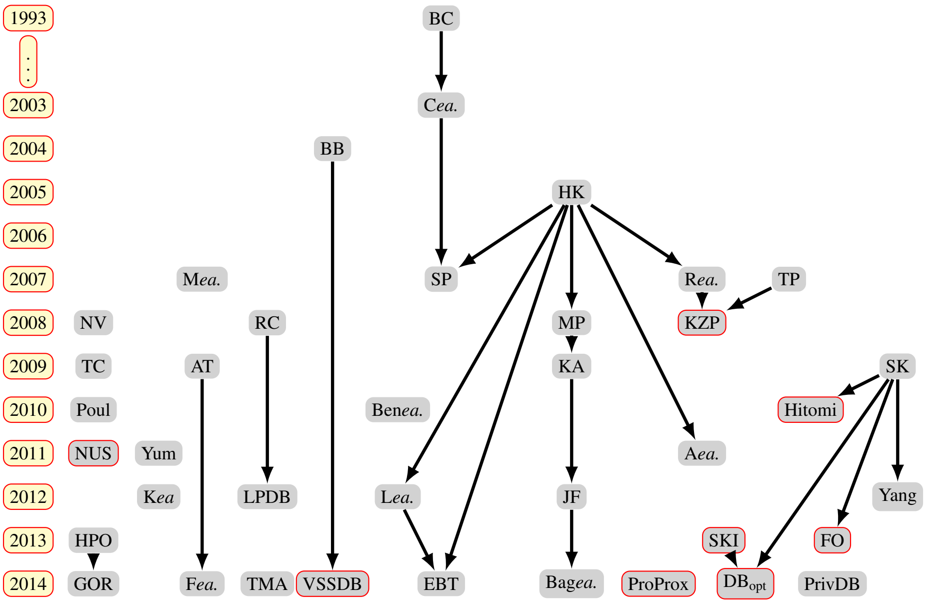

Fig. 4. Graph of dependency of protocols.

4 Tool Box

For each fraud, we describe the different strategies known to increase the chances for a successful attack.

4.1 Attack Strategies

We list all types of attacks that we have found in our survey. For each strategy we also give a simple example.

Distance Fraud (DF): We identified several techniques to mount DF attacks:

- Protocols using two possible responses (one for $c_i = 1$ and the other for $c_i = 0$): the prover computes the two possible responses in advance and examines if they are equal or not. If they are, then the prover responds correctly. Else, he responds randomly.

Example: We consider a protocol which, during the fast phase, uses two independent values: v^0 and v^1. At each round i, the prover responds either v_i^0 if $c_i = 0$ or v_i^1 if $c_i = 1$, v_i^0 and v_i^1 denote respectively the i^th^ bit of v^0 and v^1. So at each round i, the prover has a $\frac{1}{2}$ chance of having $v_i^0 = v_i^1$, and so he responds correctly regardless of the verifier's challenge c_i, or $v_i^0 \neq v_i^1$, and so the prover sends a random response before receiving the challenge c_i to counter the distance bounding. To summmarize, the probability that the prover responds correctly at each round is $\frac{1}{2} \times 1 + \frac{1}{2} \times \frac{1}{2} = \frac{3}{4}$.

- Protocols using the value of the challenge recieved to compute the response: the prover has a $\frac{1}{2}$ chance to respond correctly.

Example: We take the Brands and Chaum protocol [14], in which during the initialisation phase the prover picks a bit-string m. During the challenge-response phase, at each round i the prover computes his response with m_i, which denotes the i^th^ bit of m, and the value of the challenge c_i: $r_i = m_i \oplus c_i$. In this context, the prover cannot predict any response, so he sends his random response before receiving the verifier's challenge to counter the distance bounding. The probability to have the correct answer is $\frac{1}{2}$ at each round.

Mafia Fraud (MF): We list different strategies to perform a MF attack:

- *Protocols that do not use signature of the transcript:* the adversary can pre-ask the prover with a random challenge in order to obtain some responses beforehand. Either he guessed the challenge properly and is able to respond upon recieving the verifier's challenge, or he didn't, and he has to make a random guess to answer the challenge.

Example: We consider the following protocol with 3 rounds: during the initialisation phase, the prover and the verifier compute two values as responses for the next phase, let $v^0 = 1||0||1$ and $v^1 = 0||1||1$ be them. The fast phase runs as follows: the verifier sends a challenge c_i. After receiving the challenge, the prover sends v_i^0 if $c_i = 0$ and v_i^1 if $c_i = 1$ where v_i^0 and v_i^1 denote respectively the i^th^ bit of v^0 and v^1. To authenticate the prover, the verifier checks the response and the round-trip delay time. An adversary between the prover and the verifier, mounts a MF attack as follows. Before the fast phase, he sends to the prover the following challenges: 1, 0, 1. So he receives these responses: 0, 0, 1. Now, he waits for the challenges from the verifier, we suppose they are: 0, 0, 1. For the first round, the adversary does not know the value of v_1^0 so he responds randomly, but for the others challenges he knows the value of the response. Here, the probability of the adversary responds correctly is $\frac{1}{2}$ (the probability to guess correctly the first response), instead of $\left(\frac{1}{2}\right)^3$ (the probability to guess all responses) if he does not ask the prover before.

- *Protocols using signature of the transcript:* The adversary cannot pre-ask the prover in order to have some guaranteed responses, because any wrong guess would change the transcript. Since he is not able to forge a valid signature on a different transcript, he must not alter it.

Example: We consider a protocol in which during the verification phase, the prover sends a signature of R and C, respectively the concatenation of the responses and the challenges transmitted during the fast phase. Because of that, the adversary must not alter the transcript. He can either try to guess the challenge and use a pre ask strategy, or the response by using a post ask strategy. In the latter, he forwards the legitimate challenge to the prover after answering it. We note that both strategies are equivalent in this case, since in both scenarios, the adversary has $\left(\frac{1}{2}\right)^n$ to guess either the challenge or the response for the n successive rounds.

Impersonation Fraud (IF): We enumerate the ways to impersonate the prover:

- We suppose the prover is close to the verifier. The adversary can play several MiM to recover all key bits.

Example: We consider a protocol where during the first phase, verifier and prover compute two values v and k where v is the result of a function (often a PRF) and k is $k = v \oplus x$. During the challenge response phase, the prover sends v_i if $c_i = 0$ and k_i if $c_i = 1$. An adversary A can impersonate a prover by recovering the secret x during a *man-in-the-middle*. To learn a bit x_i of that key, A can, during the fast bit exchange, toggle the value of challenge bit c_i when it is transmitted from the verifier to prover and leave all other messages unmodified. The attacker then observes the verifier's reaction. As a matter of fact, if the verifier accepts the prover, it means that the prover's answer r_i was nevertheless correct, and thus that $v_i = k_i$. As $k_i = v_i \oplus x_i$, A concludes that $x_i = 0$. Similarly, if the verifier refuses the prover, the adversary concludes that $x_i = 1$.

- The adversary can always use the exhaustive research to find the key.

Example: For all protocols, the adversary can always make an assumption on the prover's key and try it by running the protocol with the verifier.

Terrorist Fraud (TF): We show two techniques to increase the chance of success of a TF attack:

- In some protocols, the far-away prover can send his partner A a table that contains all $c_i \mapsto r_i$, without revealing his secret key, so A can respond correctly during the fast phase but he cannot impersonate the prover during an other session.

Example: We consider a protocol which, during the fast phase, uses two independent values: v^0 and v^1. They are obtained from a PRF and do not give any information about the secret key. At each round i, the prover responds either v_i^0 if $c_i = 0$ or v_i^1 if $c_i = 1$, where v_i^0 and v_i^1 denote respectively the i$^{\text{th}}$ bit of v^0 and v^1. After computing v^0 and v^1, the far away prover can send all values to his partner, allowing him to successfully run the fast phase with the verifier.i

- In some cases, the far-away prover cannot send his partner the same table as above, because giving both possible answers for all rounds would leak his secret. Then A would be able to impersonate the prover during an other session. To avoid this, and nevertheless be authenticated to the verifier, the prover sends a table where some entries are correctly computed and others are fully random. Like this, his partner performs the rapid phase of challenge-response with the verifier in some cases and he cannot recover all bits of the secret key. This strategy is particularly efficient for protocols that assume a noisy communication channel and allow some errors.

Example: We consider a protocol which during the first phase, verifier and prover compute two values v and k where v is the result of a function and k is $k = v \oplus x$. During the challenge response phase, the prover sends v_i if $c_i = 0$ and k_i if $c_i = 1$. With this protocol, the prover cannot send his partner the complete values of k and v because the partner would recover the key x by computing $x = k \oplus v$. But the prover can send v', which is v where some bits are replaced by random ones. He can similarly generate k' This way, the partner cannot recover all bits of the key but he can respond correctly to a large number of the verifier's challenges, and so increases the chance to the prover to be authenticated by the verifier.

PRF: Protocols using PRF to pre-compute response for the fast phase are often exposed to a DF attack, and protocols using PRF to compute the signature of the *transcript* are often exposed to a MiM attack which permits to the adversary to impersonate the prover.

Example: In Section 3.1, we propose such attacks based on PRF construction and show how several protocols using PRF are not protected from DF attack or MiM attack.

4.2 Design

Through our readings, we compile some protocol's features in the Table 2 and we could see that protocol's particularities prevent some attacks described above. We present these features, as guidelines for the construction of secure protocols.

Transcript: We note that the presence of the (signed) *transcript* in the verification phase prevents Mafia Fraud attacks. Indeed, it prevents the adversary from using a pre ask strategy to improve his success probability since the verifier aborts the protocol if the challenges do not correspond to the adversary's challenges. So, all the protocols [14–16, 48, 37, 45, 56, 34, 25, 30, 27, 12, 26] that use the signature of the *transcript*, have a success probability to MF at $\left(\frac{1}{t}\right)^n$ where t denotes the number of possible values for a challenge. Except the FO protocol [25], because it uses two modes of execution: one verifies the transcript and the other not.

PRF Output: From the moment where the output of the PRF is cut into several parts like in [29, 42, 5, 36, 7, 50, 4, 34, 55, 38, 51, 20–22], it is possible to mount an attack using PRF construction (see Section 3.1) and so an DF attack can be successful. All protocols cited before bear the consequences of this risk. Then it is better to avoid splitting the result of a PRF in order to avoid such kind of attacks like in [47, 33, 44, 37, 45, 56, 28, 12, 25].

Specifics Responses:
- Some protocols [14, 56] use two complementary values to respond to the verifier's challenge to prevent DF attack, in other words for a challenge equal to 0 the verifier waits the value $r_i = v_i$ and for the challenge equal to 1 the verifier waits the value $r_i = \bar{v_i}$. Then a dishonest prover cannot predict responses for all rounds.
- Protocols [16, 48, 5, 50, 51, 21] use all challenges received to compute the response r_i. Because of this, an adversary between the verifier and the prover has a lower chance to guess the string of challenges (and so to have correct repsonses).

5 Conclusion

We first used the model proposed in [9] to review how classical threat models are covered by this general framework. Then we explicited some relations between these notions. Our main contribution is the list of existing attacks for 42 DB protocols of the literature. For 17 of them we were able to improve the best known attacks. Finally from this experience we could extract the common features to the most secure protocols, and compile them into a tool box that can be used to design safe protocols.

In the future, we would like to improve the classification of all these protocols and possibly extract the best features of each to build new, more robust ones. Another extension is to study DH property for all these protocols, which is a recent threat model and for which it is not always obvious to mount some attacks. In the same vein, the anonimity and privacy properties are more and more studied in new protocols, hence it would be interesting to include them in future work.

References

1. M. R. S. Abyaneh. Security analysis of two distance-bounding protocols. *CoRR*, abs/1107.3047, 2011.
2. J. Aumasson, A. Mitrokotsa, and P. Peris-Lopez. A note on a privacy-preserving distance-bounding protocol. In *ICICS 2011*, pages 78–92, 2011.
3. G. Avoine, M. A. Bingöl, S. Kardas, C. Lauradoux, and B. Martin. A formal framework for cryptanalyzing RFID distance bounding protocols. *IACR Cryptology ePrint Archive*, 2009:543, 2009.
4. G. Avoine, C. Lauradoux, and B. Martin. How secret-sharing can defeat terrorist fraud. In *Wisec'11*, pages 145–156. ACM, 2011.
5. G. Avoine and A. Tchamkerten. An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement. In *ISC'09*, 2009.
6. A. Bay, I. Boureanu, A. Mitrokotsa, I. Spulber, and S. Vaudenay. The bussard-bagga and other distance bounding protocols under man-in-the-middle attacks. In *Inscrypt*, 2012.
7. A. Benfarah, B. Miscopein, J. Gorce, C. Lauradoux, and B. Roux. Distance bounding protocols on TH-UWB radios. In *GLOBECOM*, pages 1–6, 2010.
8. I. Boureanu, A. Mitrokotsa, and S. Vaudenay. On the pseudorandom function assumption in (secure) distance-bounding protocols - prf-ness alone does not stop the frauds! In *LATINCRYPT*, 2012.
9. I. Boureanu, A. Mitrokotsa, and S. Vaudenay. Practical & provably secure distance-bounding. *IACR Cryptology ePrint Archive*, 2013:465, 2013.
10. I. Boureanu, A. Mitrokotsa, and S. Vaudenay. Secure and lightweight distance-bounding. In *LightSec 2013*, pages 97–113, 2013.
11. I. Boureanu, A. Mitrokotsa, and S. Vaudenay. Towards secure distance bounding. In *FSE 2013*, pages 55–67, 2013.
12. I. Boureanu and S. Vaudenay. Optimal proximity proofs. In *Inscrypt*, pages 170–190, 2014.
13. I. Boureanu and S. Vaudenay. Challenges in distance bounding. *IEEE Security & Privacy*, 13(1):41–48, 2015.
14. S. Brands and D. Chaum. Distance-bounding protocols (extended abstract). In *EUROCRYPT93, LNCS 765*, pages 344–359. Springer-Verlag, 1993.
15. L. Bussard and W. Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. In *IFIP SEC 2005*, 2005.

16. S. Capkun, L. Buttyn, and J.-P. Hubaux. Sector: secure tracking of node encounters in multi-hop wireless networks. In *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pages 21–32, 2003.
17. C. Cremers, K. B. Rasmussen, B. Schmidt, and S. Capkun. Distance hijacking attacks on distance bounding protocols. In *IEEE S & P*, 2012.
18. Y. Desmedt. Major security problems with the "unforgeable" (feige-)fiat-shamir proofs of identity and how to overcome them. In *Securicom'88*, pages 147–159, 1988.
19. U. Dürholz, M. Fischlin, M. Kasper, and C. Onete. A formal approach to distance-bounding RFID protocols. In *Information Security, 14th International Conference, ISC 2011, Xi'an, China, October 26-29, 2011. Proceedings*, pages 47–62, 2011.
20. R. Entezari, H. Bahramgiri, and M. Tajamolian. A mafia and distance fraud high-resistance rfid distance bounding protocol. In *ISCISC*, pages 67–72, 2014.
21. A. Falahati and H. Jannati. All-or-nothing approach to protect a distance bounding protocol against terrorist fraud attack for low-cost devices. *Electronic Commerce Research*, 15(1):75–95, 2015.
22. M. S. Fatemeh Baghernejad, Nasour Bagheri. Security analysis of the distance bounding protocol proposed by jannati and falahati. *Electrical and Computer Engineering Innovations*, 2(2):85–92, 2014.
23. K. Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley & Sons, Inc., New York, NY, USA, 2 edition, 2003.
24. M. Fischlin and C. Onete. Provably secure distance-bounding: an analysis of prominent protocols. *IACR Cryptology ePrint Archive*, 2012:128, 2012.
25. M. Fischlin and C. Onete. Terrorism in distance bounding: Modeling terrorist-fraud resistance. In *ACNS'13*, pages 414–431, 2013.
26. S. Gambs, M.-O. Killijian, C. Lauradoux, C. Onete, M. Roy, and M. Traoré. VSSDB: A Verifiable Secret-Sharing and Distance-Bounding protocol. In *BalkanCryptSec'14*, 2014.
27. S. Gambs, C. Onete, and J. Robert. Prover anonymous and deniable distance-bounding authentication. *IACR Cryptology ePrint Archive*, 2014:114, 2014.
28. A. O. Gürel, A. Arslan, and M. Akgün. Non-uniform stepping approach to rfid distance bounding problem. In *DPM'10/SETOP'10*, pages 64–78, 2011.
29. G. P. Hancke and M. G. Kuhn. An rfid distance bounding protocol. In *SECURECOMM '05*, pages 67–73, Washington, DC, USA, 2005. IEEE Computer Society.
30. J. Hermans, R. Peeters, and C. Onete. Efficient, secure, private distance bounding without key updates. In *WISEC'13*, pages 207–218, 2013.
31. A. F. Hoda Jannati. Mutual implementation of predefined and random challenges over RFID distance bounding protocol. *ISCISC*, pages 43–47, 2012.
32. Y. ju Tu and S. Piramuthu. Rfid distance bounding protocols. In *In First International EURASIP Workshop on RFID Technology*, 2007.
33. G. Kapoor, W. Zhou, and S. Piramuthu. Distance bounding protocol for multiple RFID tag authentication. In *2008 IEEE/IPIP EUC 2008*, pages 115–120, 2008.
34. S. Kardaş, M. S. Kiraz, M. A. Bingöl, and H. Demirci. A novel rfid distance bounding protocol based on physically unclonable functions. In *RFIDSec'11*, pages 78–93, 2012.
35. C. H. Kim. Security analysis of YKHL distance bounding protocol with adjustable false acceptance rate. *IEEE Communications Letters*, 15(10):1078–1080, 2011.
36. C. H. Kim and G. Avoine. Rfid distance bounding protocol with mixed challenges to prevent relay attacks. In *CANS '09*, pages 119–133, 2009.
37. C. H. Kim, G. Avoine, F. Koeune, F. Standaert, and O. Pereira. The swiss-knife RFID distance bounding protocol. In *ICISC*, pages 98–115, 2008.
38. S. Lee, J. S. Kim, S. J. Hong, and J. Kim. Distance bounding with delayed responses. *IEEE Communications Letters*, 16(9):1478–1481, 2012.

39. C. Meadows, R. Poovendran, D. Pavlovic, L. Chang, and P. F. Syverson. Distance bounding protocols: Authentication logic analysis and collusion attacks. In *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, pages 279–298, 2007.
40. M. Meghdadi, S. Ozdemir, and I. Gler. A survey of wormhole-based attacks and their countermeasures in wireless sensor networks. *IETE Technical Review*, 28(2):89–102, 2011.
41. A. Mitrokotsa, C. Onete, and S. Vaudenay. Mafia fraud attack against the rč distance-bounding protocol. In *RFID-TA 2012*, pages 74–79, 2012.
42. J. Munilla and A. Peinado. Distance bounding protocols for rfid enhanced by using void-challenges and analysis in noisy channels. *Wirel. Commun. Mob. Comput.*, 8(9):1227–1232, Nov. 2008.
43. J. Munilla and A. Peinado. Security analysis of tu and piramuthu's protocol. In *NTMS 2008*, pages 1–5, 2008.
44. V. Nikov and M. Vauclair. Yet another secure distance-bounding protocol. In *SECRYPT 2008*, pages 218–221, 2008.
45. P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and J. C. A. van der Lubbe. Shedding some light on RFID distance bounding protocols and terrorist attacks. *CoRR*, abs/0906.4618, 2009.
46. K. B. Rasmussen and S. Capkun. Location privacy of distance bounding protocols. In *CCS 2008*, pages 149–160, 2008.
47. J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji. Detecting relay attacks with timing-based protocols. In *ASIACCS '07*, pages 204–213. ACM, 2007.
48. D. Singele and B. Preneel. Distance bounding in noisy environments. In *Security and Privacy in Ad-hoc and Sensor Networks*, pages 101–115. Springer Berlin Heidelberg, 2007.
49. N. O. Tippenhauer and S. Capkun. Id-based secure distance bounding and localization. In *ESORICS 2009*, pages 621–636, 2009.
50. R. Trujillo-Rasua, B. Martin, and G. Avoine. The poulidor distance-bounding protocol. In *RFIDSec'10*, pages 239–257, 2010.
51. R. Trujillo-Rasua, B. Martin, and G. Avoine. Distance-bounding facing both mafia and distance frauds: Technical report. *CoRR*, abs/1405.5704, 2014.
52. S. Vaudenay. On modeling terrorist fraud. In *ProvSec 2013*, pages 1–20, 2013.
53. S. Vaudenay. Proof of proximity of knowledge. *IACR ePrint Archive*, 2014:695, 2014.
54. S. Vaudenay. Private and secure public-key distance bounding – application to NFC payment. In *Financial Cryptography and Data Security 2015*, January 2015.
55. A. Yang, Y. Zhuang, and D. S. Wong. An efficient single-slow-phase mutually authenticated rfid distance bounding protocol with tag privacy. In *ICICS*, pages 285–292, 2012.
56. D. H. Yum, J. S. Kim, S. J. Hong, and P. J. Lee. Distance bounding protocol with adjustable false acceptance rate. *IEEE Communications Letters*, 15(4):434–436, 2011.

Year	Protocol	Success Probability					
		DF	**MiM**	MF	IF	**CF**	TF
1993	[14]^n	$\left(\frac{1}{2}\right)^n$ [28]	$\left(\frac{1}{2}\right)^n$ [37]	$\left(\frac{1}{2}\right)^n$ [37]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	1 [37]	1 [37]
2003	[16]^n	$\left(\frac{1}{2}\right)^n$ [28]	$\left(\frac{1}{2}\right)^n$ [10]	$\left(\frac{1}{2}\right)^n$ [37]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	1 [37]	1 [37]
2004	[15]^n	1 [6]	$\left(\frac{1}{2}\right)^n$ [10]	$\left(\frac{1}{2}\right)^n$ [10]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	1 [6]	1 [6]
2005	[29]^n	$\left(\frac{3}{4}\right)^n$ [28] to 1 [8]	$\left(\frac{3}{4}\right)^n$ [37]	$\left(\frac{3}{4}\right)^n$ [37]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	1 [37]	1 [37]
2007	[47]^n	$\left(\frac{3}{4}\right)^n$ [28] to 1 [8]	$\left(\frac{3}{4}\right)^n$ [37] to 1 [8]	$\left(\frac{3}{4}\right)^n$ [37]	1 [37]	$\left(\frac{3}{4}\right)^v$ [37]	$\left(\frac{3}{4}\right)^v$ [37]
	[48]^n	$\left(\frac{1}{2}\right)^k$ [28]	$\left(\frac{1}{2}\right)^n$ [37]	$\left(\frac{1}{2}\right)^n$ [37]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	1 [37]	1 [37]
	[32]^n	$\left(\frac{3}{4}\right)^n$ [43]	$\left(\frac{9}{16}\right)^n$ [43] to 1 [37]	$\left(\frac{9}{16}\right)^n$ [43]	1 [37]	$\left(\frac{3}{4}\right)^v$ [43]	$\left(\frac{3}{4}\right)^v$ [43]
	[39]^s	$\left(\frac{1}{2}\right)^{\lvert N_V\rvert}$ [39]	$\left(\frac{1}{2}\right)^{\lvert N_V\rvert}$ [39]	$\left(\frac{1}{2}\right)^{\lvert N_V\rvert}$ [39]	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	1 [39]	1 [39]
2008	[46]^s	$\left(\frac{1}{2}\right)^n$ [46]	P_1 [41]	P_1 [41]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	1 [2]	1 [2]
	[33]$^{\mathbf{s}}$	$\boldsymbol{\left(\frac{3}{4}\right)^n}$ **[33]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[33]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[33]**	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	$\boldsymbol{\left(\frac{3}{4}\right)^v}$ **[33]**	$\boldsymbol{\left(\frac{3}{4}\right)^v}$ **[33]**
	[42]^n	$\left(\frac{3}{4}\right)^n$ [28] to **1**	$\left(\frac{3}{5}\right)^n$ [42]	$\left(\frac{3}{5}\right)^n$ [42]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	1 [28]	1 [28]
	[44]^n	$1/k$ [44]	$\left(\frac{1}{2}\right)^n$ [44]	$\left(\frac{1}{2}\right)^n$ [44]	$\left(\frac{1}{2}\right)^n$ [44]	1 [37]	1 [37]
2009	[5]^n	$\left(\frac{3}{4}\right)^n$ to 1 [8]	$\left(\frac{1}{2}\right)^n \times \left(\frac{n}{2}+1\right)$ [5]	$\left(\frac{1}{2}\right)^n \times \left(\frac{n}{2}+1\right)$ [5]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	**1**	**1**
	[37]^n	$\left(\frac{3}{4}\right)^n$ [37]	$\left(\frac{1}{2}\right)^n$ [37] to 1 [8]	$\left(\frac{1}{2}\right)^n$ [37]	1 [8]	$\left(\frac{3}{4}\right)^v$ [37]	$\left(\frac{3}{4}\right)^v$ [37]
	[36]^n	$\left(\frac{7}{8}\right)^n$ [36] to **1**	$\left(\frac{1}{2}\right)^n$ [36]	$\left(\frac{1}{2}\right)^n$ [36]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	1 [28]	1 [28]
	[49]^s	$\left(\frac{1}{2}\right)^n$ [49]	$\left(\frac{1}{2}\right)^n$ [49]	$\left(\frac{1}{2}\right)^n$ [49]	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	**1**	**1**
2010	[7]^s	$\boldsymbol{\left(\frac{3}{4}\right)^n}$ to **1**	$\left(\frac{4x^2-1}{4x^3}\right)^n$ [7]	$\left(\frac{4x^2-1}{4x^3}\right)^n$ [7]	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	**1**	**1**
	[7]^s	$\boldsymbol{\left(\frac{3}{4}\right)^n}$ to **1**	$\left(z\left(\frac{5}{2}-2z\right)\right)^n$ [7]	$\left(z\left(\frac{5}{2}-2z\right)\right)^n$ [7]	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	**1**	**1**
	[45]$^{\mathbf{n}}$	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[45]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[45]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[45]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	$\boldsymbol{\left(\frac{3}{4}\right)^v}$ **[45]**	$\boldsymbol{\left(\frac{3}{4}\right)^v}$ **[45]**
	[50]^n	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ to **1**	P_2 [50]	P_2 [50]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	**1**	**1**
2011	[4]^n	$\left(\frac{3}{4}\right)^n$ [4] to 1 [8]	$\left(\frac{2}{3}\right)^n$ [4] to 1 [8]	$\left(\frac{2}{3}\right)^n$ [4]	1 [8]	$\left(\frac{5}{6}\right)^v$ [4]	$\left(\frac{5}{6}\right)^v$ [4]
	[56]^n	$\left(\frac{1}{2}\right)^{\tau n}$ [35]	$\left(\frac{1}{2}\right)^n$ [35] to **1**	$\left(\frac{1}{2}\right)^n$ [35]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ to **1**	**1**	**1**
	[28]$^{\mathbf{n}}$	$\boldsymbol{\left(\frac{3}{4}\right)^n}$ **[1]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[28]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[28]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[28]**	$\boldsymbol{\left(\frac{3}{4}\right)^v}$	$\boldsymbol{\left(\frac{3}{4}\right)^v}$
2012	[34]^{2s}	$\boldsymbol{\left(\frac{3}{4}\right)^n}$ to **1**	$\left(\frac{1}{2}\right)^n$ [34]	$\left(\frac{1}{2}\right)^n$ [34]	$\boldsymbol{\left(\frac{1}{2}\right)^{2s}}$	**1**	**1**
	[55]^n	$\left(\frac{3}{4}\right)^n$ [55] to 1 [24]	$\left(\frac{3}{4}\right)^n$ [55]	$\left(\frac{3}{4}\right)^n$ [55]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	1 [24]	1 [24]
	[38]^n	$\boldsymbol{\left(\frac{3}{4k}\right)^n}$ to **1**	$\left(\frac{2k+1}{4k}\right)^n$ [38]	$\left(\frac{2k+1}{4k}\right)^n$ [38]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	**1**	**1**
	[41]^s	$\left(\frac{1}{2}\right)^n$ [41]	$\left(\frac{1}{2}\right)^n$ [41]	$\left(\frac{1}{2}\right)^n$ [41]	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	**1**	**1**
	[31]^n	$\left(\frac{3}{4}\right)^n$ [31]	P_3 [31] to 1 [22]	P_3 [31]	1 [22]	$\boldsymbol{\left(\frac{3}{4}\right)^v}$	$\boldsymbol{\left(\frac{3}{4}\right)^v}$
2013	**[9]$^{\mathbf{s}}$**	$\boldsymbol{\left(\frac{3}{4}\right)^n}$ **[9]**	$\boldsymbol{\left(\frac{2}{3}\right)^n}$ **[9]**	$\boldsymbol{\left(\frac{2}{3}\right)^n}$ **[9]**	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	$\boldsymbol{\left(\frac{5}{6}\right)^v}$ **[11]**	$\boldsymbol{\left(\frac{5}{6}\right)^v}$ **[11]**
	[25]$^{\mathbf{2s}}$	$\boldsymbol{\left(\frac{3}{4}\right)^n}$ **[52]**	$\boldsymbol{\left(\frac{3}{4}\right)^n}$ **[52]**	$\boldsymbol{\left(\frac{3}{4}\right)^n}$ **[52]**	$\boldsymbol{\left(\frac{1}{2}\right)^{2s}}$	$\boldsymbol{\left(\frac{3}{4}\right)^v}$ **[52]**	$\boldsymbol{\left(\frac{3}{4}\right)^v}$ **[52]**
	[30]^s	$\left(\frac{3}{4}\right)^n$ [30]	$\left(\frac{1}{2}\right)^n$ [30]	$\left(\frac{1}{2}\right)^n$ [30]	$\left(\frac{1}{2}\right)^s$ [30]	1 [30]	1 [30]
2014	[27]^s	$\left(\frac{3}{4}\right)^n$ [27]	$\left(\frac{1}{2}\right)^n$ [27]	$\left(\frac{1}{2}\right)^n$ [27]	$\left(\frac{1}{2}\right)^{\lvert\mathbb{G}\rvert}$ [27]	1 [27]	1 [27]
	[12]$^{\mathbf{s}}$	$\boldsymbol{\left(\frac{1}{t}\right)^n}$ **[12]**	$\boldsymbol{\left(\frac{1}{t}\right)^n}$ **[12]**	$\boldsymbol{\left(\frac{1}{t}\right)^n}$ **[12]**	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	$\boldsymbol{\left(\frac{t-1}{t}\right)^v}$ **[12]**	$\boldsymbol{\left(\frac{t-1}{t}\right)^v}$ **[12]**
	[12]$^{\mathbf{s}}$	$\boldsymbol{\left(\frac{1}{\sqrt{2}}\right)^n}$ **[12]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[12]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[12]**	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	$\boldsymbol{\left(\frac{1}{\sqrt{2}}\right)^v}$ **[12]**	$\boldsymbol{\left(\frac{1}{\sqrt{2}}\right)^v}$ **[12]**
	[12]^s	$\left(\frac{1}{t}\right)^n$ [12]	$\left(\frac{1}{t}\right)^n$ [12]	$\left(\frac{1}{t}\right)^n$ [12]	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	1 [12]	1 [12]
	[53]$^{\mathbf{s}}$	$\boldsymbol{\left(\frac{1}{\sqrt{2}}\right)^{ns}}$ **[53]**	$\boldsymbol{\left(\frac{1}{2}\right)^{ns}}$ **[53]**	$\boldsymbol{\left(\frac{1}{2}\right)^{ns}}$ **[53]**	$\boldsymbol{\left(\frac{1}{2}\right)^s}$ **[53]**	$\boldsymbol{\left(\frac{1}{\sqrt{2}}\right)^{ns}}$ **[53]**	$\boldsymbol{\left(\frac{1}{\sqrt{2}}\right)^{ns}}$ **[53]**
	[51]^s	P_4 [51]	P_5 [51]	P_5 [51]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	**1**	**1**
	[26]$^{\mathbf{s}}$	$\boldsymbol{\left(\frac{3}{4}\right)^n}$ **[26]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[26]**	$\boldsymbol{\left(\frac{1}{2}\right)^n}$ **[26]**	$\boldsymbol{\left(\frac{1}{2}\right)^{2s}}$ **[26]**	$\boldsymbol{\left(\frac{3}{4}\right)^v}$ **[26]**	$\boldsymbol{\left(\frac{3}{4}\right)^v}$ **[26]**
	[20]^s	$\left(\frac{k+2}{4k}\right)^n$ [20] to **1**	$\left(\frac{k+2}{4k}\right)^n$ [20]	$\left(\frac{k+2}{4k}\right)^n$ [20]	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	**1**	**1**
	[21]^n	$\left(1+\frac{n}{2}\right)\left(\frac{1}{2}\right)^n$ [21] to **1**	$\left(1+\frac{n}{2}\right)\left(\frac{1}{2}\right)^n$ [21]	$\left(1+\frac{n}{2}\right)\left(\frac{1}{2}\right)^n$ [21]	$\boldsymbol{\left(\frac{1}{2}\right)^n}$	P_6 [21]	P_6 [21]
	[22]^n	$\left(\frac{3}{4}\right)^n$ [22] to **1**	P_3 [22]	P_3 [22]	$\left(\frac{1}{2}\right)^n$ [22]	1 [22]	1 [22]
2015	[54]^s	$\left(\frac{3}{4}\right)^n$ [54]	$\left(\frac{3}{4}\right)^n$ [54]	$\left(\frac{3}{4}\right)^n$ [54]	$\boldsymbol{\left(\frac{1}{2}\right)^s}$	1 [54]	1 [54]

Table 1. Summary of the success probability of attacks, where v is such that : for $n-v$ rounds, the adversary knows all responses independently of the challenge's value, and for the v other rounds, for each of them the adversary knows $t-1$ responses on t values possible for a challenge. For [12], $t \geq 3$. For [48], the last $(n-k)$ bits depend on the first k bits. The number of rounds is always n (expect [39, 46, 44, 41] where there is only one round) and the exponent after the citation is the size of the key. Where $P_1 = \frac{1}{n(|stream_V|-n)(|stream_P|-n)}$, $P_2 = \sum_{t=1}^{n} \frac{1}{2^t}(\prod_{i=t}^{n} \max(\Pr(r_1 = c_i'|c_t \neq \tilde{c_t}), ..., \Pr((r_n = c_i'|c_t \neq \tilde{c_t}))) + \frac{1}{2^n}$, $P_3 = \left(\frac{3}{4}\right)^n \times \left(\frac{1}{2}\right)^n + \left(\frac{1}{2}\right)^{n-1} \times \left(1 - \left(\frac{3}{4}\right)^n\right)$, $P_4 = \frac{1}{4}Pr(D_{n-1}) + \frac{1}{2^n} + \frac{1}{8}\sum_{j=1}^{i-1} Pr(D_j)\frac{1}{2^{i-j}}$, $P_5 = \left(\frac{1}{2}\right)^n + Pr(M_n|C_n \widetilde{\neq C}_{n-1})\left(1 - \left(\frac{1}{2}\right)^n\right)$ and $P_6 = \left(\frac{1}{2}\right)^{n-L}\left(\frac{n}{L} + \left(\frac{1}{2}\right)^L\left(1 - \frac{n}{L}\right)\right)$.

Protocol	PRF	$Sign$(transcript)	based on
Brands and Chaum (1993) [14]	✗	✓	
Capkun *et al.* (2003) [16]	✗	✓	BC [14]
Bussard and Bagga (2004) [15]	✗	✓	
Hancke and Kuhn (2005) [29]	✓2	✗	
Ried *et al.* (2007) [47]	✓1	✗	HK [29]
Singele & Preneel (2007) [48]	✗	✓	*Cea.* [16] and HK [29]
Tu & Piramuthu (2007) [32]	✗	✗	
Meadows *et al.* (2007) [39]	✗	✗	
RČ (2008) [46]	✗	✗	
KZP (2008) [33]	**✓1**	✗	***Rea.* [47] and TP [32]**
Munilla & Peinado (2008) [42]	✓3	✗	HK [29]
Nikov & Vauclair (2008) [44]	✓1	✗	
Avoine & Tchamkerten (2009) [5]	✓2	✗	
Swiss-Knife (2009) [37]	✓1	✓	
Kim & Avoine (2009) [36]	✓4	✗	MP [42]
Tippenhauer & Capkun (2009) [49]	✗	✗	
Benfarah *et al.* A (2010) [7]	✓4	✗	
Benfarah *et al.* B (2010) [7]	✓4	✗	
Hitomi (2010) [45]	**✓1**	✓	**SK [37]**
Poulidor (2010) [50]	✓2	✗	
Avoine *et al.* (2011) [4]	✓n-1	✗	HK [29]
Yum *et al.* (2011) [56]	✓1	✓	
NUS (2011) [28]	**✓1**	✗	
Kardas *et al.* (2012) [34]	✓3	✓	
Yang *et al.* (2012) [55]	✓3	✗	SK [37]
Lee *et al.* (2012) [38]	✓4	✗	HK [29]
LPDB (2012) [41]	✓2	✗	RČ [46]
Jannati & Falahati (2012) [31]	✓2	✗	KA [36]
SKI$_{\mathbf{pro}}$ (2013) [9]	**✓1**	✗	
Fischlin & Onete (2013) [25]	**✓1**	✓	**SK [37]**
HPO (2013) [30]	✗	✓	
GOR (2014) [27]	✗	✓	HPO [30]
DB1 (2014) [12]	**✓1**	✓	**SKI [9] and SK [37]**
DB2 (2014) [12]	**✓1**	✓	**SKI [9] and SK [37]**
DB3 (2014) [12]	✓1	✓	SKI [9] and SK [37]
ProProx (2014) [53]	✗	✗	
TMA (2014) [51]	✓3	✗	
VSSDB (2014) [26]	✗	✓	**BB [15]**
EBT (2014) [20]	✓6	✗	HK [29] and *Lea.* [38]
Falahati *et al.* (2014) [21]	✓1	✗	AT [5]
Baghernejad *et al.* (2014) [22]	✓3	✗	JF [31]
PrivDB (2015) [54]	✗	✗	

Table 2. Features of the protocols, where ✓denotes the presence of the feature and ✗denotes the absence of the feature. The number next to ✓represent the number of cutting of the PRF's output during the initialisation phase.