Secure & Lightweight Distance-Bounding

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

[1] EPFL
CH-1015 Lausanne, Switzerland
http://lasec.epfl.ch
[2] University of Applied Sciences of Western Switzerland (HES-SO)
CH-1227 Geneva, Switzerland
katerina.mitrokotsa@hesge.ch

Abstract. Distance-bounding is a practical solution aiming to prevent relay attacks. The main challenge when designing such protocols is maintaining their inexpensive cryptographic nature, whilst being able to protect against as many, if not all, of the classical threats posed in their context. Moreover, in distance-bounding, some subtle security shortcomings related to the PRF (pseudorandom function) assumption and ingenious attack techniques based on observing verifiers' outputs have recently been put forward. Also, the recent terrorist-fraud by Hancke somehow recalls once more the need to account for noisy communications in the security analysis of distance-bounding. In this paper, we attempt to incorporate the lessons taught by these new developments in our distance-bounding protocol design. The result is a new class of protocols, with increasing levels of security, accommodating the latest advances[3]; at the same time, we preserve the lightweight nature of the design throughout the whole class.

1 Introduction

In [9], Brands and Chaum introduced the notion of distance-bounding (DB) protocols. The aim is to have a prover demonstrate his proximity to a verifier, and authenticate himself to this verifier. The proof of proximity can be an efficient deterrent against relay attacks [15]. DB protocols [23,25,34,37] generally consist of an *initialisation phase* (where the parties establish some short-term secret) and a *distance-bounding* phase. This latter phase is time-critical. It imposes very fast computation, typically of less than a single clock cycle per round, and the verifier measures the time-of-flight of the messages exchanged. This is how the verifier ascertains a distance-bound between him and the prover.

In the literature covering such protocols, the threat-model comprises three well-established types of attacks. The first is *distance-fraud* (DF), in which a prover tries to convince the verifier that he is closer than he really is. In the second type, *mafia-fraud* (MF), an adversary communicates with both a prover and a verifier which are far apart, and the adversary tries to convince the verifier that the prover would be close enough to be granted privileges. Finally, in a *terrorist-fraud* (TF), an adversary is getting the

[3] An earlier version of this line of work was presented in [7]. Also, some preliminaries and adjacent topics made the subject of an invited talk [8].

necessary and sufficient help from a coerced, far-away prover in order to pass the protocol only during this corrupted run, but not in a later, coercion-free session. Generalisations of these frauds have also been described. In [13], Cremers *et al.* describe distance-hijacking as a mixture between distance-fraud and terrorist-fraud: one dishonest, far-away prover exploits several honest provers to gain privileges. Impersonation-fraud is presented in [16]; as its name suggests, one dishonest prover tries to impersonate an honest one.

The first DB protocols were not secure against terrorist-fraud [9,21,29,35]. Then, to name but a few, Bussard and Bagga [10], Hancke and Kuhn [21], Munilla and Peinado [29], Kim and Avoine [24], Reid *et al.* [34] proposed schemes addressing terrorist-fraud protection or mafia-fraud protection, or a better suitability to practice, etc. For instance, in [3], the TDB protocol by Avoine *et al.* addresses specifically the protection against terrorist-fraud, using threshold secret sharing schemes. Nonetheless, many attacks [1,27,28,31,30] onto DB protocols [25,35,37,33] continue to be published. To this end, Kim *et al.* stated [24] that there is no DB protocol, which can resist well against all three classical frauds and has only one-bit challenges/responses per iteration in the distance-bounding phase.

Recently, the first attempts to formalise DB have emerged. In [2], Avoine *et al.* give a semi-formal model for distance-bounding. Dürholz *et al.* [17] follow, with a more precise formalisation in which the expression TF appears possibly too strong (i.e., many protocols that are intuitively TF-resistant are shown insecure against TF in this model). At the same time, [6,4] expose some essential shortcomings of the DB design and of the security claims related to it (i.e., [6] exposes building blocks for DB, like pseudorandom functions (PRFs), that lead to DF and generalised MF attacks; [4] shows that public-key mechanisms may fail to provide TF).

In this paper, we attempt to take notice of all these recent developments: e.g., we strengthen the way PRFs are used in DB, we reinforce and take forward the manner in which secret sharing schemes can be employed to build TF-resistant DB protocols and, finally, we attempt to combine it all harmoniously in such a way that we obtain robust, yet lightweight DB.

2 Summary of DB Security: Status & Results

At this early stage, in Table 1 below, we present the security status of several, existing DB protocols, and announce that of two of our DB protocols to be presented herein. Namely, please notice our $\mathbf{SKI_{pro}}$ and $\mathbf{SKI_{lite}}$ protocols and their security guarantees by comparison to the other DB protocols in this table.

In this table, we assumed channels that are *not noisy*, though further in this paper we extend the analysis on our protocols to the case of noisy channels as well. Let us briefly explain some details from the table. Let n be the number of DB rounds, and $\nu < n$. Let t be the number of possible values for a challenge, i.e., classically $t = 2$. In the case of terrorist-fraud, we supposed along standard lines two facts: 1. for $n - \nu$ DB rounds, the adversary has got all responses, irrespective of the value of challenges; 2. for the other ν DB rounds, for each such round, the adversary knows the answers for $t - 1$ (out of t) values possible for a challenge.

Table 1. Probability of success of the best (known) attacks onto DB

Protocol	Success Probability				
	Key-Length	Distance-Fraud	Mafia-fraud	Terrorist-Fraud	MIM
Brands & Chaum [9]	n	$(1/2)^n$ cnf [19]	$(1/2)^n$ cnf [25]	1 cnf [25]	$(1/2)^n$
Bussard & Bagga [11]	n	1 cnf [4]	$(1/2)^n$	1 cnf [4]	$(1/2)^n$
Čapkun *et al.* (SECTOR) [12]	n	$(1/2)^n$ cnf [19]	$(1/2)^n$ cnf [25]	1 cnf [25]	$(1/2)^n$
Hancke & Kuhn [21]	n	$(3/4)^n$ cnf [19]	$(3/4)^n$ cnf [25]	1 cnf [25]	$(3/4)^n$
Reid *et al.* [34]	n	$(3/4)^n$ cnf [19]	$(3/4)^n$ cnf [26]	$(3/4)^\nu$ cnf [25]	$(3/4)^n$ or 1 cnf [4]
Singelée & Preneel [35]	n	$(1/2)^n$ cnf [19]	$(1/2)^n$ cnf [25]	1 cnf [25]	$(1/2)^n$
Tu & Piramuthu [37]	n	$(3/4)^n$ cnf [30]	$(9/16)^n$ cnf [30]	$(3/4)^\nu$ cnf [30]	1 cnf [25]
Munilla & Peinado [29]	n	$(3/4)^n$ cnf [19]	$(3/5)^n$ cnf [19]	1 cnf [19]	$(3/5)^n$
Swiss-Knife [25]	n	$(3/4)^n$ cnf [25]	$(1/2)^n$ cnf [25]	$(3/4)^\nu$ cnf [25]	$(1/2)^n$
Kim & Avoine [24]	n	$(7/8)^n$ cnf [19]	$(1/2)^n$ cnf [19]	1 cnf [19]	$(1/2)^n$
Nikov & Vauclair [32] *	$\bar{k}$	$1/k$ cnf [25]	$(1/2)^n$ cnf [25]	1 cnf [25]	$(1/2)^n$
Avoine *et al.* [3]	n	$(3/4)^n$	$(2/3)^n$	$(2/3)^\nu$	$(2/3)^n$
SKI$_{\text{pro}}$	ℓ	$(3/4)^n$	$(2/3)^n$	$(2/3)^\nu$	$(2/3)^n$
SKI$_{\text{lite}}$	ℓ	$(3/4)^n$	$(3/4)^n$	1	$(3/4)^n$

* In this case, $\bar{k}$ and k are additional parameters; this protocol requires heavy computations. The parameter ν is explained in the paragraph above.

From the table, we can already notice some similarities between the protocol in [21], by Hancke and Kuhn, and the simplest version of the **SKI** protocols to be introduced herein, namely **SKI$_{\text{lite}}$** . Also, we can see a certain closeness between the Avoine *et al.* protocol [3] and a stronger version of our protocols, i.e., **SKI$_{\text{pro}}$** . In that sense, what this line of work brings as a novelty is a more precise design of the protocols (i.e., there are design differences between the **SKI** protocols and its similar counterparts in the table). Our design is driven by very recent exhibited DB attack-techniques and classical frauds [20,4,6]. We also propose a more in-depth security analysis due to the same recent threats and a more attentive look into the DB security in noisy communications[4].

As the reader will see in our design choices presented in Section 5.3 and in the attacks we present in Section 5, we get our attack bounds (as per Table 1) by enforcing certain requirements on our DB building blocks. We hereby mention some of these enforcements: 1. the use of the PRF instance in the initialisation phase is masked, i.e., we use $f_x(\cdot) \oplus M$ for a randomly looking M, instead of just employing $f_x(\cdot)$; 2. the DB response-function is such that it uses the secret x in a way that it does not conflict with x keying $f_x(\cdot)$ in the initialisation phase; 3. a linear transformation is chosen at the initialisation phase to be applied on the secret x, before we use x in the response-function. These are the sort of design-amendments imposed by the recent, aforementioned attacks [20,4,6]. In fact, the very new attack-technique in [20] is not taken into account in Table 1. With our **SKI$_{\text{pro}}$** protocol, we resist the TF by Hancke.

Structure. The rest of this paper is structured as follows. In Section 3, we reiterate what are the settings in which DB communications are taking place. In Section 4, we express the DB security requirements in these communication settings; to do so, we follow the well-established understandings of the classical frauds in the existing literature, and we also offer some generalisations. In Section 5, we introduce our protocol-schema, called **SKI**, explain most of its design, and present two instantiations of it, i.e., **SKI$_{\text{pro}}$** and

[4] E.g., a recent attack-technique [20] by Hancke, described on page 10, reiterates the importance of considering noise in DB, bit-based computations.

$\mathbf{SKI_{lite}}$. We then argue that these protocols protect against the frauds as they were described in Section 4. In Section 6, we conclude. In an appendix, we present other instantiations of our protocol schema (i.e., $\mathbf{SKI_{shamir}}$ and $\mathbf{SKI_4}$), varying in their security strength, but all remaining lightweight.

3 DB Communication

In what follows, we present the main, very straightforward guidelines of the settings in which DB protocols are considered to run. (The underlying communication and the threat model could actually be properly formalised, e.g., as an interactive system [18]. This is not our purpose herein, and it will be left for an extended version of this paper.)

DB protocols are run in natural communication settings :

- there is a generally accepted notion of time, e.g., there is a time-unit;
- a notion of measurable/quantifiable location and distance;
- the timed communication obeys the laws of physics.

All participants (provers, verifiers, adversaries) comply to the following:

- have the correct means/algorithms to run their part (e.g., an RFID tag, a reader, both, etc.);
- are fixed at some location;
- send messages with a destination through a broadcast, non-authenticated, asynchronous channel.

Furthermore, honest participants read messages that are intended for them, when these messages reach them. An attacker can change the destination of a message, aiming it to himself and can create his own messages and inject them into the communication. In the distance-bounding phase, the noise of the channel cannot be corrected by honest parties (i.e., the adversary may have extra technology to do so, but the honest parties cannot do so within the limits of time imposed).

NOTE: It is clear in this model that an adversary can do very limited man-in-the-middle attacks. If a verifier sends a message and expects a fast response back, this deters a man-in-the-middle (MiM) adversary to send the message further to a prover and await for the prover's response to convey to the verifier, i.e., as such responses would arrive to the verifier too late. In the same way, an adversary can get very limited, *online* help even from a coerced, but far-away prover.

4 DB: Protocols & Requirements

In line with the previous section, we present these requirements using natural language. (As before, it is worth mentioning that in a formal model for DB, these could be expressed, e.g., in the style of completeness/soundness requirements on interactive systems [18], with thresholds on the success/failure probabilities of different events or sequences of events. This is left for an extended version of this paper tackling formalisation and provable security aspects.)

4.1 Distance-Bounding Protocols

In general, let the provers be denoted by P and the verifiers by V. Let $\mathcal{A}$ denote the adversary and P^* designate dishonest provers. We assume that verifiers end the DB protocols by outputting one bit b denoting acceptance, i.e. $b = 1$, or rejection, i.e., $b = 0$. (I.e., this is in line with practice, where a LED turning green or red on an access point denotes granted or denied access, respectively). In the generalised MF presented in [4], it is this sort of return channel that facilitates the attacks (i.e., logically, intruders learn more information by looking also at whether the run was successful or not.). We proceed with the definition of a DB protocol.

Definition 1. ***Distance-Bounding Protocols.*** *A* distance-bounding (DB) protocol *is defined by an acceptable distance-bound, a prover P and a verifier V, each running probabilistic, efficient*[5] *algorithms, both sharing a long-term key x such that the following happens:*

- *the verifier's algorithm efficiently terminates on* any *interaction*[6]*;*
- *if the prover P is within the acceptable distance-bound from the verifier V, then the verifier V terminates successfully (almost always*[7]*).*

DB can take place in concurrent settings as well, i.e., there are several provers and several verifiers, sharing secrets in a pairwise manner, all running the same DB protocol in parallel. We can also think of the scenario where one prover and one verifier run the same protocol several times, in a sequential fashion. In the description of the security requirements to follow, we will also consider such multi-party extensions of the definition above.

4.2 Distance-Bounding Requirements

Let $\alpha, \beta, \gamma, \gamma' \in [0,1]$ be some variables (depending on some parameters, e.g., on the number of rounds in the distance-bounding phase), or let $\alpha, \beta, \gamma, \gamma' \in [0,1]$ be some fixed constants (e.g., pre-established security-tolerance limits). The security requirements of DB protocols are described below, and they depend on the values of these $\alpha, \beta, \gamma, \gamma'$.

Definition 2. ***α-resistance to distance-fraud****: We say that a DB protocol is* α-resistant to distance-fraud *if any far-away, dishonest prover P^* which is running the protocol with a verifier V, on their shared secret, cannot make the verifier accept (i.e., output* 1*) with a probability greater than α (taken over the random choices made by V).*

[5] In theory, "efficient" denotes polynomial in some security parameters. In practice, one should be able to see clearly that these algorithms are computationally inexpensive.

[6] Even if the prover is dishonest, after a finite number of steps, a reader either accepts or rejects.

[7] In theory, "almost always" would entail some overwhelming probability in a security parameter. In practice, it means that there are some exceptional circumstances where the verifier would reject correct transcripts. I.e., in extremely noisy channels (which occur very rarely) the verifier would be bound to reject messages that originated correctly from the prover.

As we said before, depending on the security desired, one may take α to be negligible in a security parameter (e.g., $c^{\Theta(n)}$, where c is a constant in $(0,1)$ and n is the number of rounds and/or the key-length) or, simply a fixed value in $(0,1)$.

If we consider a multi-party setting (e.g., taking several runs, with far-away and close-by provers), then the DF-resistance as defined above captures the notion of distance hijacking in [13], i.e., an experiment in which a dishonest far-away prover P^* may use several other provers to get authenticated as if he was close to the verifier. The DF-resistance we assess in Section 5.5 can be extended to account for such a multi-party setting.

We move now to the resistance to mafia-fraud.

Definition 3. β***-resistance to MF****: We say that a DB protocol is* β-resistant to mafia-fraud *if an adversary $\mathcal{A}$ interfering up to his powers within the interaction between a far-away, honest prover P and verifier V on their shared secret cannot make the verifier accept (i.e., output* 1*) with a probability greater than* β *(taken over the random choices made by P,V and $\mathcal{A}$).*

Of course, this definition of MF-resistance can be cast in a multi-party setting as well and it can also be generalised to a stronger MiM attack. For instance, in a multi-party setting, we consider that during a learning phase, the attacker $\mathcal{A}$ interacts, in parallel, with several provers and several verifiers and then —in an attack phase— $\mathcal{A}$ tries to win in a run in front of a verifier, which is far-away from several provers. (In a practical setting, it is as if an attacker would have cloned several tags and would make them interact with several readers with which they are registered. From such a multi-party communication, he can get potentially more benefits, faster.) In our security assessment in Section 5.5, the arguments can be easily extended to such a concurrent setting.

Definition 4. (γ,γ')***-resistance to TF****: We say that a DB protocol is* (γ,γ')-resistant to terrorist-fraud *if for any far-away, coerced prover P^*, it is the case that, below,* (1) *implies* (2)

— (1). *an adversary $\mathcal{A}$ interfering up to his powers with an interaction between P^* and verifier V on their shared secret, where this interaction is successful with probability at least γ (over the random choices of V and $\mathcal{A}$),*

— (2). *$\mathcal{A}$ can later succeed on his own to make the verifier accept in a new protocol run with a probability greater than γ' (taken over the new random choices made by V and $\mathcal{A}$).*

This definition of TF-resistance can also be presented in a multi-party setting and generalised to a stronger threat. For instance, one first thing to imagine is a coercion-phase followed by a multi-party MF, i.e., a MiM phase as we mentioned after Definition 3. In fact, our assessment of TF-resistance made in Section 5.5 can be extended easily to such an enhanced threat.

5 The SKI Distance-Bounding Protocols

In the first part of this section, we present our protocols. In the second, we explain our design. In the third, we assess their resistance to frauds, upon the definitions and discussions in Section 3 and Section 4.

5.1 Protocols' Descriptions

We now propose a schema of DB protocols denoted ***SKI*** and presented in Figure 1, i.e., we use "schema" to denote that, at this stage, we leave under-determined the choice of the exact primitives to be used inside. Later in the section, by suggesting different instantiations of these primitives, we obtain a class of DB protocols, with varying levels of resistance to DB attacks. Nonetheless, from the weakest to the strongest of them, these protocols are lightweight.

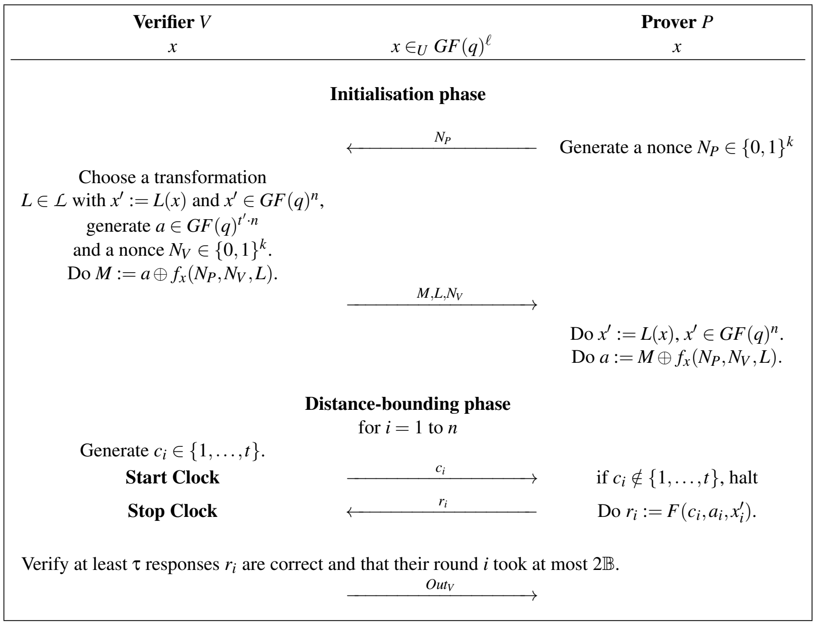
Verifier V
x
$x \in_U GF(q)^\ell$
Prover P
x
Initialisation phase
N_P
Generate a nonce $N_P \in \{0,1\}^k$
Choose a transformation
$L \in \mathcal{L}$ with $x' := L(x)$ and $x' \in GF(q)^n$,
generate $a \in GF(q)^{t' \cdot n}$
and a nonce $N_V \in \{0,1\}^k$.
Do $M := a \oplus f_x(N_P, N_V, L)$.
M, L, N_V
Do $x' := L(x)$, $x' \in GF(q)^n$.
Do $a := M \oplus f_x(N_P, N_V, L)$.
Distance-bounding phase
for $i = 1$ to n
Generate $c_i \in \{1, \ldots, t\}$.
Start Clock
c_i
if $c_i \notin \{1, \ldots, t\}$, halt
Stop Clock
r_i
Do $r_i := F(c_i, a_i, x'_i)$.
Verify at least τ responses r_i are correct and that their round i took at most $2\mathbb{B}$.
Out_V

Fig. 1. The **SKI** schema of Distance-Bounding Protocols

Let s be a security parameter. The secret key x is a vector of ℓ elements over $GF(q)$, with $\ell \in \Omega(s)$, with q a constant giving the power of prime so that we work over $GF(q)$, the finite field with q elements. In some of the concrete examples to follow, we employ $q = 2$, i.e., we work over bits[8]. The **SKI** protocols are built using a PRF, denoted $(f_x)_{x \in GF(q)^\ell}$.

The prover selects a nonce N_P of k bits and sends it over to the verifier, for $k \in \Omega(s)$.

The verifier V first selects a nonce N_V also of k bits. Then, he picks a random linear transformation L from a set $\mathcal{L}$, set that is specified by the **SKI** protocol instance (as we

[8] Irrespective of working over bits or not, we consider that the practicality of today's cheap RFID/NFC cards goes anyway beyond one-bit responses [36]. Moreover, pre-computation tables can be used to render online computation very efficient.

will concretely see later). The parties compute $x' = L(x)$. We consider that the vector x' obtained out of x through L has length n, with $n \in \Omega(s)$.

The distance-bounding phase will have n rounds with challenges taking t possible values, for a constant t. Another constant we use is t'. To give an anticipative intuition, t' is such that $t' \leq t$ In our main proposal, we use $t = 3$ and $t' = 2$, i.e., we keep the lightweight character.

The verifier finally picks a masking-vector M with $M \in GF(q)^{t' \cdot n}$. Further, the element $a = (a_1, \ldots, a_n)$ is established by V and it is sent encrypted into M as follows: $M = a \oplus f_x(N_P, N_V, L)$, with $M \in GF(q)^{t' \cdot n}$. (As we can see, **SKI** employs $f_x(N_P, N_V, L)$ as illustrated, with $f_x(N_P, N_V, L) \in GF(q)^{t' \cdot n}$.)

So, $c = (c_1, \ldots, c_n)$ is the challenge-vector with $c_i \in \{1, \ldots, t\}$, $r_i := F(c_i, a_i, x'_i)$ is the i-th response to the i-th challenge c_i, with $i \in \{1, \ldots, n\}$, $r_i \in GF(q)$ and F as specified below. In our concrete proposals, we use $t = 3$, or $t = 2$ for the lighter version.

By $\mathbb{B}$, we denote the maximal accepted time-of-flight of one challenge/response between P and V. Assume that messages travel uniformly with a speed of one space-unit/time-unit. Then, as usual, $\mathbb{B}$ is also the distance-bound acceptable between P and V.

The protocol ends with a message Out_V denoting the output of the verifier (i.e., the success/failure of the protocol). We tolerate communication noise. Thus, a successful run is that where at least τ out of n DB responses are correct and have been delivered within the time-bound $\mathbb{B}$. Later in the paper, it will be implied what bound on τ we need (in function of n and the probability of the communication noise) such that legitimate runs are not overruled, yet malicious runs are not validated.

As we anticipated already, all the variables and functions in **SKI** will be instantiated with small values and lightweight mathematical objects.

NOTE: To address noisy time-critical communications, we introduce the probability p_{noise} of one response being erroneous (à la [21]). The probability that at least τ responses out of n are of the correct kind is clearly given by:

$$B(n, \tau, 1 - p_{noise}) = \sum_{i=\tau}^{n} \binom{n}{i} (1 - p_{noise})^i p_{noise}^{n-i}$$

It is natural to choose τ (and other parameters) such that we operate with correct DB protocols, cnf. with Definition 1. I.e., the protocol is complete: honest communications succeed with high probability. Let us assess this. So, let $\varepsilon > 0$. If we force τ such that $\tau \leq (1 - p_{\mathsf{noise}} - \varepsilon)n$, then it implies $B(n, \tau, 1 - p_{noise}) \geq 1 - e^{-2\varepsilon^2 n}$ (due to the Hoeffding bound [22]), i.e., it implies the verifier accepting honest communications with a very good probability as n grows. Also, in practice, we may use a constant p_{noise} (i.e., hard-coded in the protocol implementation). This also entails employing τ as some parameter which is linear in terms of n (in order to have negligible probabilities of failure in honest executions).

A detailed analysis on optimising the selection of τ is provided in [14].

5.2 Towards Specific Building Blocks

We now continue with the instantiations of some of objects in our **SKI** schema. Our choices of them will be explained shortly.

The response-function F. In the main body of the paper, we consider one generic such response-function F in which the i-th response ($1 \leq i \leq n$) is produced as follows:

$$\mathbf{F_{xor}}(c_i, a_i, x'_i) = x'_i 1_{c_i=t} + (a_i)_1 1_{c_i \in \{t,1\}} + \ldots + (a_i)_{t'} 1_{c_i \in \{t,t'\}}$$

where $c_i \in \{1, \ldots, t\}$, $x'_i \in GF(q)$, $q \geq 2$, $(a_i)_j \in GF(q)$, $j \in \{1, \ldots, t'\}$, and 1_R is 1 if R is true and 0 otherwise.

In the appendix, we will present other possibilities for the response-function F.

The set of transformations $\mathcal{L}$. We can consider several sets $\mathcal{L}$ of transformations to be used in the PRF-instance of **SKI**'s initialisation phase. (Such a set is formally referred to as a *leakage scheme* and it is thuswise defined in [5].)

We consider $\mathcal{L}_{\mathsf{bit}}$ consisting of all L_μ transforms, where L_μ is defined using a vector $\mu \in GF(q)^\ell$ by

$$L_\mu(x) = (\mu \cdot x, \ldots, \mu \cdot x)$$

I.e., all coordinates of the vector $L_\mu(x)$ are set to the scalar product between μ and x.

We could consider other suitable[9] instances of $\mathcal{L}$, but this may entail a number of DB rounds greater than n (because of the noise involved). Or, if no noise is to be considered, we could employ $\mathcal{L} = \mathcal{L}_{\mathsf{classic}}$, i.e., the set containing a single function L which is the identity function. For the purpose of this paper (i.e., lightweight DB protocols), we restrict ourselves to using $\mathcal{L}_{\mathsf{bit}}$ as per the above. Also, if we do not view TF-resistance, then we can leave the $\mathcal{L}$ set empty.

5.3 SKI: Design Choices

Using a mask M. We chose to use a mask M, indirectly decided by V, due to the fact that just using $f_x(\ldots, N_P, \ldots)$ to calculate a can lead to DF attacks [6]. To mount such an attack, a corrupt P^* basically chooses a trapdoor N_P to bias the output distribution of $f_x(\ldots, N_P, \ldots)$. By using the mask M, we prevent such a P^* from reaching his goal.

The PRF f & the Response-Function F. Already note the $\mathbf{F_{xor}}$ is in fact carrying on from the TDB protocol [3], i.e., using secret-sharing ideas to protect against TF. Also, it preserves the lightweight trend.

Moreover, in **SKI**, the chosen f and F have to meet the following requirement. They are such that it is indistinguishable when L_μ is applied to the secret key x and when it is applied to another randomly selected $\bar{x} \in GF(q)^\ell$, even if we are given access to the other messages in the protocol, i.e., N_P, N_V and some results related to $f_x(N_p, N_V, L)$ and $L(\bar{x})$ as per the protocol, or even if we choose them adaptively as an adversary may do. This security-enforcement also has an impact on an additional property of the PRF f (i.e., on how its keys are used outside its calls). This design choice is motivated by the attacks in [6], where a trapdoor choice of N_P or N_V together with x being used in L_μ could lead generalised MF attacks.

The F-functions that we take (see the previous section) enjoy other properties that help attain security in front of DB frauds. E.g., similarly to [3], the F-functions are such

[9] "Suitable" denotes here compliant with deterring the TF in [20].

that knowing the complete table of the response-function F for a given c_i leaks x'_i, yet knowing only up to t' entries challenge-response in this table discloses no information about x'_i. Please refer to [5] for details.

The Set $\mathcal{L}$ of Transforms. The idea of the set $\mathcal{L}$ is that, when leaking some noisy versions of $L(x)$ for some chosen $L \in \mathcal{L}$, the adversary can reconstruct x without noise.

We introduced this transformation in order to protect against a TF observed by Hancke [20]. In this attack, a malicious prover could select a noise-vector e of Hamming weight $n - \tau$ and provide a slightly modified, but full table of all $c_i \mapsto F(c_i, a_i, x_i)$ functions. The modification in the table is as follows: if $e_i = 1$, then the output of $F(v, a_i, x_i)$ is flipped, where $v \in \{1, \ldots, t\}$ is one, randomly chosen value of the possible values for the i-th challenge c_i. Assuming that the adversary has a powerful device which can answer to V without noise, then this adversary passes with probability $\gamma = 1$. Then, an adversary –out of this full table– can reconstruct $x + e$. Then, x cannot be recovered efficiently by the adversary (whilst P^* substantially helped the adversary towards passing the protocol).

Recall that –in our protocols– the "master-secret" that f is applied upon for one value of c_i is not necessarily the shared key x, but instead it is x' with $x' = L(x)$, where L is the transformation chosen by V in the initialisation phase of the protocol and discussed above. As we said, this offers better protection against new types of threats: by introducing $L(x)$ instead of x inside F, then in Hancke's attack, the adversary can get to learn $L(x) + e$.

Imagine now a dishonest prover as above choosing a noise-vector e of Hamming weight (at most) t, with e possibly depending on x and a transformation L chosen in the current run initialisation phase. If $\mathcal{L}_{bit}$ is used as in our protocols, then in n rounds of the attack as per the above, an attacker $\mathcal{A}$ deduces $\mu \cdot x$, for all obtained μ in the round-transformations $L = L_\mu (L \in \mathcal{L}_{bit})$. The attacker does so by computing the majority of the vector x' that he learns[10] out of the responses. These values $\mu \cdot x$ can be correct with a high probability, if $t = HW(e)$ is not too close to n, i.e., t is at least less than $\frac{n}{2}$. (HW denotes the Hamming weight.) Then, $\mathcal{A}$ can solve a linear system to get x. Hence, leaking x makes this attack *not* a valid terrorist-fraud (since the dishonest prover helped $\mathcal{A}$ pass the protocol, but he also leaked x to this attacker). I.e., our protocol instance with $\mathcal{L}_{bit}$ resists the attack by Hancke [20].

5.4 The Main Instances of SKI

We now propose the most interesting instances of **SKI**: the first one protecting against all threats presented in Section 3 (and rendering the TF scenario by Hancke [20] hard to mount for some parameters, if not infeasible), and a second one, much more lightweight, not offering TF-resistance, but only DF- and MF-resistance. Of course, the spectrum of the class **SKI** is much larger, and we will touch upon that in our appendix.

- $\mathbf{SKI_{pro}}$: defined by the response-function $\mathbf{F_{xor}}$ above, with $q = 2$, $t = 3$, $t' = 2$, i.e., $F(c_i, a_i, x'_i) = (a_i)_{c_i}$ for $c_i \in \{1,2\}$ and $F(3, a_i, x'_i) = x'_i + (a_i)_1 + (a_i)_2$, with $(a_i)_1, (a_i)_2, x_i \in GF(2)$, and $\mathcal{L} = \mathcal{L}_{\mathsf{bit}}$;

[10] We presume that if you know the full table of the response for a given c_i, then this leaks x'. Our F functions are like that.

- **$\mathbf{SKI_{lite}}$** : defined by a variant of response-function $\mathbf{F_{xor}}$ above (not depending on x'_i), with $q = 2$, $t = t' = 2$, i.e., $F(c_i, a_i, x'_i) = (a_i)_{c_i}$ for $c_i \in \{1,2\}$, with $(a_i)_1, (a_i)_2 \in GF(2)$, and $\mathcal{L} = \emptyset$. (In **$\mathbf{SKI_{lite}}$** , $\mathcal{L} = \emptyset$ since in the response-function F there is no x' used, as TF-resistance is not envisaged by this instance.)

Note, once more, that both protocols are very inexpensive computationally.

5.5 Security Analysis

In this section, we simply describe the best-known attack strategies for mounting DB frauds onto **SKI**. We report the security analysis for $\mathbf{SKI_{pro}}$, made in a symbolic form (i.e, on variables t, q, on the function F, etc). The analysis for $\mathbf{SKI_{lite}}$ is omitted, as it follows exactly the same principles (where eventually just the numerical values for t, q, or the expression of F would change). In an extended version of this work [5], we will give the formal proofs showing that these attacks are indeed the best attainable attacks against **SKI**, i.e., their probabilities of success can be shown to be the actual provable security bounds.

DF-resistance for $\mathbf{SKI_{pro}}$. Intuitively, to defeat DF-resistance, the dishonest, far-away prover P^* has to anticipate the challenge before it reaches him, to compute the response-function F with the challenge as one of the arguments, and to do so as early as possible. Then, he needs to send the resulting response pre-emptively. So, in real terms, this P^* is computing the preimage of a map $c_i \mapsto F(c_i, a_i, x'_i)$ and he gets more successful at mounting this fraud as this computable preimage gets larger. (Note that the size of this computable preimage depends on some random choice, i.e., on the value selected for a_i).

We recall that our response-function for $\mathbf{SKI_{pro}}$, taken on the i-th DB round, is as follows: $F(c_i, a_i, x'_i) = (a_i)_{c_i}$ for $c_i \in \{1,2\}$ and $F(3, a_i, x'_i) = x'_i + (a_i)_1 + (a_i)_2$. (Let us assume a fixed transformation L that gives x': e.g., as for $\mathbf{SKI_{pro}}$, one L_μ chosen in such a protocol round; this does not affect the rest).

So, the best case for P^* to invert $c_i \mapsto F(c_i, a_i, x'_i)$, i.e., to get the right answer on an "anticipated" challenge, is when $(a_i)_1 = (a_i)_2 = x_i$. In this case, he would know the answer, no matter which of the 3 values c_i actually takes, i.e., the preimage of the map $c_i \mapsto F(c_i, a_i, x'_i)$ has size 3. Over the choice of a_i, this would happen with probability $\frac{1}{4}$. If you look further at the response-function, you will note that it is impossible to invert the map onto one specific value of c_i, i.e., to make the aforementioned preimage have size 1. As the preimage can only have size 1, 2 or 3, then the prover narrowing his correct answers over a space of 2 values for the challenges can happen with probability $1 - \frac{1}{4}$.

So, the expected value of the size of this pre-image over the choices of a_i (i.e., the expected number of values for the challenge c_i that the prover could anticipate the answer for) is $(2 \times \frac{3}{4} + 3 \times \frac{1}{4}) = \frac{9}{4}$.

Remember that in $\mathbf{SKI_{pro}}$, in total, we have $t = 3$ values for any challenge. So, given x' fixed, each iteration has a probability to succeed equal of $\frac{9}{4} \times \frac{1}{3} = \frac{3}{4}$.

We note that there is no other mechanism that this prover P^* could pull through. For instance, since it is the verifier who chooses a and M, the distribution of the a_i's is uniform, i.e., not influenced by a possible trapdoor choice of N_P from P^*. (This excludes the DF attacks in [6].)

So, we have just given the description of the best (mathematical) strategy of P^* to mount a DF, which passes with a probability of $(\frac{3}{4})^n$, if no noise is considered. (This is as reported on our initial table, Table 1, on page 3.).

If, in turn, we do consider noise, then the overall success probability is going to be $B(n,\tau,\frac{3}{4}) = \sum_{i=\tau}^{n}\binom{n}{i}(\frac{3}{4})^i(\frac{1}{4})^{n-i}$. Then, for $\tau > (\frac{3}{4}+\varepsilon)n$, we have $B(n,\tau,\frac{3}{4})$ less than $e^{-2\varepsilon^2 n}$, for some $\varepsilon > 0$ (by the Hoeffding bound [22]).

□

MF-resistance for **SKI_{pro}** . Assume a mafia-fraud attacker $\mathcal{A}$ taking part, up to his capabilities, in an interaction between P and V.

Assuming that the attacker does not learn anything conclusive during the initialisation phase (about x or how to respond in the DB phase)[11], the probability of him succeeding in this fraud rests only on giving (by chance) the correct answers to the challenges sent by V (before P does so); I.e., the probability of $\mathcal{A}$ of succeeding in this MF is given by

$$p = \prod_{1\le i\le n} \Pr_{c_i\in\{1,\dots,t\}}[r_i \text{ being correct for } c_i | \, c_i \text{ is sent by } V].$$

Getting r_i correct for c_i can be attained in two distinct ways: 1. in the event $e1$ of guessing $c_i' = c_i$ and sending it beforehand to such a P_j and getting the correct response r_i, or 2. in the event $e2$ of simply guessing the correct answer r_i (for a challenge $c_i' \neq c_i$).

So the probability of success in one round is $\Pr[e1]+\Pr[e2] = \frac{1}{t}+\frac{t-1}{t}\times\frac{1}{q}$. In $\mathbf{SKI_{pro}}$, $t = 3$ and $q = 2$, so we get a concrete, overall p of $(\frac{2}{3})^n$, if no noise is considered within the communications. (This is as reported on our initial table, Table 1, on page 3.).

If we consider the noise of the channels, then we get $p = B(n,\tau,\Pr[e1]+\Pr[e2]) = B(n,\tau,\frac{2}{3})$. Then, for $\tau > (\frac{2}{3}+\varepsilon)n$, we have $B(n,\tau,\frac{2}{3})$ less than $e^{-2\varepsilon^2 n}$, for some $\varepsilon > 0$ (by the Hoeffding bound [22]).

□

[11] In fact, we can argue that this is the case in the **SKI** protocols. With high probability, there is no collision between the nonces N_P and N_V (if the space of the nonces is large enough, e.g., $2^{\Omega(n)}$). So, the output of the PRF instance f_x is not biased by the choices of these values. So, $\mathcal{A}$ learns nothing from this. Also, in $\mathbf{SKI_{pro}}$, it is not x that is used in F directly, but $L(x)$ is. Moreover, as we stated in the description of the design, the chosen F for $\mathbf{SKI_{pro}}$ is such that it is indistinguishable when it is applied to the secret key x and when it is applied to another randomly selected $x' \in GF(q)^\ell$, even if we are given access to the other messages in the protocol, i.e., N_P, N_V, or we choose them adaptively as $\mathcal{A}$ may do. (We leave the complete formalism and proof of this for an extended version of this paper [5].) So, due to this indistinguishability, it is as if the shared secret key x were not used outside the f-keying procedure. Given the above and the standard PRF assumption, it means that $\mathcal{A}$ seeing the output of f_x equates to him seeing the output of a real-random function, i.e., for $\mathcal{A}$, it is as if a were chosen at random. So, no "good" strategy comes out from the observed protocol transcript. So, there is no better strategy but what we say in the analysis above.

TF-resistance for **SKI$_{\mathbf{pro}}$** . We split the discussion in two analyses: I. for noiseless communication; II. for noisy communications.

I. Let us assume first that there are noiseless conditions.

As it is traditional in TF analysis, let us assume that the dishonest prover P^* gives away information to help $\mathcal{A}$. Namely, suppose that: 1. for $n-\nu$ DB rounds, the adversary has got all responses, irrespective of the value of challenges; 2. for the other ν DB rounds, for each such round, the adversary knows the answers for $t-1$ (out of t) possible values for a challenge. Then, his best chances to succeed is to get from V the challenges that he knows how to answer to (in the ν "decisive" rounds), i.e., chances of $(\frac{t-1}{t})^{\nu}$.

(This translates in the case of noiseless conditions to the $(\frac{2}{3})^{\nu}$ bound, reported on our initial table, Table 1, on page 3, for **SKI$_{\mathbf{pro}}$** .)

II. Let us assume now that the communications are noisy.

For concreteness, let us assume that the threshold of noise acceptance is τ out of n.

As per the strong, new attack by Hancke [20], assume that the dishonest prover P^* chose a noise bit-vector e with $HW(e) = \frac{n}{2}$. Further assume that e deterministically depends on x and L, i.e., $e = g(x,L)$ for some function g (i.e., P^* does not choose e adaptively based on the transformation L and on x). Then, assume that this P^* leaked the response table with noise e. I.e., for each i with $e_i = 0$, P^* leaked the full $c \mapsto F(c,a_i,x'_i)$ table; for each i with $e_i = 1$, P^* leaked the table except that for some random c^*_i, for which the response-value $F(c^*_i,a_i,x'_i)$ was flipped. Clearly, the leakage property[12] of F makes sure that $\mathcal{A}$ learns L and $L(x)+g(x,L)$. Due to the structure of L, the latter is a vector of Hamming weight $\frac{n}{2}$. If g has some good property, this is indistinguishable from L and $L(x)+g(y,L)$ for x and y independent. But $g(y,L)$ perfectly randomizes $L(x)$. So, it does not help to give any information about x to $\mathcal{A}$. Without knowing x, $\mathcal{A}$ cannot predict any response in another session[13] with new nonces (to compute the a vector), so $\mathcal{A}$ has no advantage to succeed in the protocol. Therefore, for any strategy, γ' is negligible.

Concretely, the probability that P^* manages to help $\mathcal{A}$ succeed in the protocol during the terrorist-fraud is the probability that at least τ rounds give a correct answer. Clearly, the $\frac{n}{2}$ rounds for which $e_i = 0$ will be correct for sure. The others are correct with probability $\frac{t-1}{t}$. So, we have $\gamma = B(\frac{n}{2}, \tau-\frac{n}{2}, \frac{t-1}{t})$.

For $\tau - \frac{n}{2} > (\frac{t-1}{t}+\varepsilon) \times \frac{n}{2}$ and some positive ε, we have γ is lower than $e^{-2\varepsilon^2\frac{n}{2}}$ (due again to the Hoeffding bound [22]). That is, for the latter, we need $\tau > \frac{5}{6}n+\varepsilon\frac{n}{2}$ (since our t is 3). This simply comes down to taking τ slightly bigger than $\frac{5}{6}n$.

□

As we mentioned above, this analysis extends almost identically to **SKI$_{\mathbf{lite}}$** , i.e., up

[12] This holds as we assume that the response-function F is such that knowing the complete table of the response-function F for a given c_i leaks x'_i.

[13] As we saw in the proof for the MIM attack, there is no other advantage that this attacker can get on his own, in such runs.

to some changes in values. As we saw, the attack bounds are obtained provided that the design-blocks inside **SKI** (i.e., the PRF f, the response-function F, their inter-play within the rest of the design, the transformations $\mathcal{L}$, etc.) meet some requirements (e.g., f and F are such that within the protocol-exchanges it does not show the fact that F uses x inside, $\mathcal{L}$ contains linear transformations, M masks $f_x(\ldots)$, etc.). All these can be formalised further and then all these attack strategies can be transformed into proofs of provable security bounds for the whole **SKI** class, i.e., all conditioned by and parametrised in F, $\mathcal{L}$, f, t, t', q, ℓ, n, τ.

6 Conclusions

We note again the similar best-attack bounds stated in Table 1 for the protocol in [21], by Hancke and Kuhn, and our simplest version of **SKI**, namely $\mathbf{SKI_{lite}}$. The Swiss-Knife protocol [25] and the Avoine *et al.* [3] also seems to enjoy good security bounds for DF and MF, but they do not protect against the new TF attack by Hancke [20].

Moreover, it was shown in [6] that the conditions on the underlying primitives need to be strengthened for DF and generalised MF security to be indeed attained. This type of attacks as in [6] can be bypassed in the **SKI** protocols. I.e., when the PRF instance is used, it is masked with the randomly looking value M, computed on the right side of the protocol avoiding the DF susceptibility shown in [6]. In our best MF description for $\mathbf{SKI_{pro}}$, we explained the idea of choosing an f and an F that together with the protocol transcript make F look as if it is not using x; in our further work, we will give the formal details of how such a PRF f needs to come together with the response-function F to attain formally the avoidance of the generalised MF exposed in [6]. We remind that we also introduced a transform on x to be used inside the response-function F in order to deter (if not avoid) the recent TF attacks by Hancke [20]. So, by all this and beyond Table 1, we conclude that very compelling security—in accordance to the recent developments in DB—is now provided by (at least one of) the **SKI** protocols.

References

1. J.-P. Aumasson, A. Mitrokotsa, and P. Peris-Lopez. A Note on a Privacy-Preserving Distance-Bounding Protocol. In *Proceedings of the 13th International Conference on Information and Communications Security (ICICS 2011)*, LNCS, pages 78–92, Beijing, China, November 2011.
2. G. Avoine, M. Bingöl, S. Kardas, C. Lauradoux, and B. Martin. A Framework for Analyzing RFID Distance Bounding Protocols. *Journal of Computer Security*, 19(2):289–317, 2011.
3. G. Avoine, C. Lauradoux, and B. Martin. How Secret-sharing can Defeat Terrorist Fraud. In *Proceedings of the 4th ACM Conference on Wireless Network Security – WiSec'11*, Hamburg, Germany, June 2011. ACM Press.
4. A. Bay, I. C. Boureanu, A. Mitrokotsa, I.-D. Spulber, and S. Vaudenay. The Bussard-Bagga and Other Distance-Bounding Protocols under Attacks. In M. Kutyłowski and M. Yung, editors, *Proceeding of the 88th China International Conference on Information Security and Cryptology (Inscrypt 2012)*, volume LNCS, pages 371–391, Heidelberg, 2012. Springer,.
5. I. Boureanu, A. Mitrokotsa, and S. Vaudenay. Provably Secure Authenticated Distance-Bounding. submitted.

6. I. Boureanu, A. Mitrokotsa, and S. Vaudenay. On the Pseudorandom Function Assumption in (Secure) Distance-Bounding Protocols. In A. Hevia and G. Neven, editors, *Progress in Cryptology – LATINCRYPT 2012*, Lecture Notes in Computer Science, pages 100–120. Springer, 2012.
7. I. Boureanu, A. Mitrokotsa, and S. Vaudenay. On secure distance bounding (extended abstract), 2013. Talk by Ioana Boureanu.
8. I. Boureanu, A. Mitrokotsa, and S. Vaudenay. Towards secure distance bounding. In *the 20th anniversary annual Fast Software Encryption (FSE 2013)*, 2013, to appear. Invited Talk by Serge Vaudenay.
9. S. Brands and D. Chaum. Distance-Bounding Protocols (Extended Abstract). In *EUROCRYPT*, pages 344–359, 1993.
10. L. Bussard and W. Bagga. Distance-Bounding Proof of Knowledge Protocols to Avoid Terrorist Fraud Attacks. Technical Report RR-04-109, EURECOM, May 2004.
11. L. Bussard and W. Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. In *Security and Privacy in the Age of Ubiquitous Computing, IFIP TC11 20th International Conference on Information Security (SEC 2005), May 30 - June 1, 2005, Chiba, Japan*, pages 223–238. Springer, 2005.
12. S. Čapkun, L. Buttyán, and J.-P. Hubaux. SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. In *ACM Workshop on Security of Ad Hoc and Sensor Networks - SASN*, pages 21 – 32. ACM, 2003.
13. C. Cremers, K. B. Rasmussen, and S. Čapkun. Distance hijacking attacks on distance bounding protocols. In *IEEE Symposium on Security and Privacy*, pages 113–127, 2012.
14. C. Dimitrakakis, A. Mitrokotsa, and S. Vaudenay. Expected Loss Bounds for Authentication in Constrained Channels. In *Proceedings of INFOCOM 2012*, pages 478–85, Orlando, FL, USA, March 2012. IEEE press.
15. S. Drimer and S. J. Murdoch. Keep your enemies close: distance bounding against smartcard relay attacks. In *Proceedings of 16th USENIX Security Symposium*, pages 7:1–7:16, Berkeley, CA, USA, 2007. USENIX Association.
16. U. Dürholz, M. Fischlin, M. Kasper, and C. Onete. A Formal Approach to Distance Bounding RFID Protocols. In *Proceedings of the* 14^{th} *Information Security Conference ISC 2011*, LNCS, pages 47–62. SPRINGER, 2011.
17. M. Fischlin and C. Onete. Subtle kinks in distance-bounding: an analysis of prominent protocols. In *WISEC*, pages 195–206, 2013.
18. O. Goldreich. *Foundations of Cryptography: Volume 1*. Cambridge University Press, New York, NY, USA, 2006.
19. A. O. Gürel, A. Arslan, and M. Akgün. Non-uniform Stepping Approach to RFID Distance Bounding Problem. In *Proceedings of the 5th international Workshop on Data Privacy management, and 3rd International Conference on Autonomous Spontaneous Security*, DPM'10/SETOP'10, pages 64–78, Berlin, Heidelberg, 2011. Springer-Verlag.
20. G. P. Hancke. Distance-bounding for rfid: Effectiveness of 'terrorist fraud' in the presence of bit errors. In *RFID-TA*, pages 91–96, 2012.
21. G. P. Hancke and M. G. Kuhn. An RFID Distance Bounding Protocol. In *SECURECOMM*, pages 67–73. ACM, 2005.
22. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):pp. 13–30, 1963.
23. G. Kapoor, W. Zhou, and S. Piramuthu. Distance Bounding Protocol for Multiple RFID Tag Authentication. In C.-Z. Xu and M. Guo, editors, *Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing - Volume 02 – EUC'08*, pages 115–120, Shanghai, China, December 2008. IEEE, IEEE Computer Society.

24. C. H. Kim and G. Avoine. RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks. In *Proceedings of the 8th International Conference on Cryptology and Networks Security (CANS 2009)*, volume 5888 of *LNCS*, pages 119–131. SPRINGER, 2009.
25. C. H. Kim, G. Avoine, F. Koeune, F. Standaert, and O. Pereira. The Swiss-Knife RFID Distance Bounding Protocol. In *International Conference on Information Security and Cryptology – ICISC*, Lecture Notes in Computer Science. Springer-Verlag, December 2008.
26. A. Mitrokotsa, C. Dimitrakakis, P. Peris-Lopez, and J. C. Hernandez-Castro. Reid et al.'s Distance Bounding Protocol and Mafia Fraud Attacks over Noisy Channels. *IEEE Communications Letters*, 14(2):121–123, February 2010.
27. A. Mitrokotsa, C. Onete, and S. Vaudenay. Mafia Fraud Attack against the RČ Distance-Bounding Protocol. In *Proceedings of the 2012 IEEE RFID Technology and Applications (IEEE RFID T-A)*, pages 74–79, Nice, France, November 2012. IEEE Press.
28. A. Mitrokotsa, P. Peris-Lopez, C. Dimitrakakis, and S. Vaudenay. On selecting the nonce length in distance-bounding protocols. *The Computer Journal*, 2013.
29. J. Munilla and A. Peinado. Distance Bounding Protocols for RFID Enhanced by Using Void-challenges and Analysis in Noisy Channels. *Wireless Communications and Mobile Computing*, 8:1227–1232, November 2008.
30. J. Munilla and A. Peinado. Security Analysis of Tu and Piramuthu's Protocol. In *New Technologies, Mobility and Security – NTMS'08*, pages 1–5, Tangier, Morocco, November 2008. IEEE Computer Society.
31. J. Munilla and A. Peinado. Attacks on a Distance Bounding Protocol. *Computer Communications*, 33:884–889, 2010.
32. V. Nikov and M. Vauclair. Yet Another Secure Distance-Bounding Protocol. In *Proceedings of the Conference on Security and Cryptography (SECRYPT 2008)*, pages 218–221, July 2008.
33. K. B. Rasmussen and S. Čapkun. Location Privacy of Distance Bounding. In *Proceedings of the Annual Conference on Computer and Communications Security (CCS)*, pages 149–160. ACM, 2008.
34. J. Reid, J. M. Gonzalez Nieto, T. Tang, and B. Senadji. Detecting Relay Attacks with Timing-based Protocols. In *ASIACCS '07: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security*, pages 204–213. ACM, 2007.
35. D. Singelée and B. Preneel. Distance Bounding in Noisy Environments. In *Proceedings of the European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS)*, volume 4572 of *LNCS*, pages 101 –115. Springer-Verlag, 2007.
36. B. Toiruul, K. O. Lee, and J. M. Kim. SLAP - A Secure but Light Authentication Protocol for RFID Based on Modular Exponentiation. In *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 29–34, November 2007.
37. Y.-J. Tu and S. Piramuthu. RFID Distance Bounding Protocols. In *Proceedings of the First International EURASIP Workshop on RFID Technology*, 2007.

A Other Instances of SKI

In this section, we present two more instances of **SKI**: one with even stronger security guarantees than what we have seen so far and another placing its assurances in between $\mathbf{SKI_{pro}}$ and $\mathbf{SKI_{lite}}$. For that, we first provide a new response-function.

Other instances of the response-function F. For the strongest version of **SKI**, we recommend the following response function.

$$\mathbf{F_{shamir}}(c_i, a_i, x'_i) = x'_i + (a_i)_1 \overline{c}_i + (a_i)_2 \overline{c}_i^{\,2} + \ldots + (a_i)_{t-1} \overline{c}_i^{\,t-1}$$

where $x_i' \in GF(q)$, $q \geq 4$, $c_i \in \{1,\ldots,t\}$ is mapped to $\bar{c}_i \in GF(q)^*$ by an arbitrary injective mapping, $(a_i)_j \in GF(q), j \in \{1,\ldots,t-1\}$;

It is obvious, given the expression of $\mathbf{F_{shamir}}$ (i.e., with x' inside it) that it is meant to protect against classical TF. If $x' = x$, then it may not protect against the newest TF scenario by Hancke [20].

*Other instances of **SKI**.* We present two more instances of **SKI** in descending order of their security strength:

- $\mathbf{SKI_{shamir}}$: defined by the response-function $\mathbf{F_{shamir}}$ above, with $q = 4$, $t = 3$, $t' = 2$, i.e., $F(c_i, a_i, x_i') = x_i' + (a_i)_1\bar{c}_i + (a_i)_2\bar{c}_i{}^2$, with $x_i, (a_i)_1, (a_i)_2 \in GF(4)$ and $\bar{c}_i \in GF(4)^*$; $\mathcal{L} = \mathcal{L}_{\mathsf{bit}}$
- $\mathbf{SKI_4}$: defined by the response-function $\mathbf{F_{xor}}$ above, with $q = 2$, $t = 4$, $t' = 3$, i.e., $F(c_i, a_i, x_i') = (a_i)_{c_i}$ for $c_i \in \{1,2,3\}$ and $F(4, a_i, x_i) = x_i' + (a_i)_1 + (a_i)_2 + (a_i)_3$, with $(a_i)_1, (a_i)_2, (a_i)_3, x_i \in GF(2)$; $\mathcal{L} = \mathcal{L}_{\mathsf{bit}}$.

As we can see, $\mathbf{SKI_{shamir}}$ is more secure even than $\mathbf{SKI_{pro}}$, with a response-function F, based on Shamir's secret-sharing scheme. In fact, recalling our DF-resistance analysis, this response-function F is more powerful when it comes to inverting the map $c_i \mapsto F(\cdot, a_i, x_i)$. Hence, the DF-resistance of this instance is better than the one of $\mathbf{SKI_{pro}}$. The opposite can be said about $\mathbf{SKI_4}$, by comparison to $\mathbf{SKI_{pro}}$.

Security of these instances. Doing an analysis similar to the one we did for $\mathbf{SKI_{pro}}$ in Section 5.5, and we consider noisy conditions, we can state the following bounds for the best mounted DF and MF attacks against these new instances:

	$\mathbf{SKI_{shamir}}$	$\mathbf{SKI_4}$
DF	$\alpha = B(n, \tau, \frac{5}{8})$	$\alpha = B(n, \tau, \frac{3}{4})$
MF	$\beta = B(n, \tau, \frac{1}{2})$	$\beta = B(n, \tau, \frac{5}{8})$

If, in turn, we do not consider noisy conditions then we get the following probabilities for the best-known TF attacks against these instances of **SKI**:

	$\mathbf{SKI_{shamir}}$	$\mathbf{SKI_4}$
TF	$(\frac{2}{3})^\nu$	$(\frac{3}{4})^\nu$