How Secret-sharing can Defeat Terrorist Fraud

Gildas Avoine
Université catholique de Louvain
ICTEAM Institute
B-1348, Louvain la Neuve, Belgium
gildas.avoine@uclouvain.be

Cédric Lauradoux
Université de Lyon, INRIA, INSA-Lyon, CITI,
F-69621, Villeurbanne, France
cedric.lauradoux@inria.fr

Benjamin Martin
Université catholique de Louvain
ICTEAM Institute
B-1348, Louvain la Neuve, Belgium
benjamin.martin@uclouvain.be

ABSTRACT

Terrorist fraud is a relay attack against distance bounding protocols where the prover conspires with an adversary to misrepresent the distance between himself and the verifier. In ideal situations, the adversary does not gain any knowledge about the prover's long-term secret. This makes designing a distance bounding protocol resistant to a such fraud tricky: the secrets of an honest prover must be protected, while those of a dishonest one should be disclosed as an incentive not to cheat.

In this paper, we demonstrate that using a secret-sharing scheme, possibly based on threshold cryptography, is well suited for thwarting terrorist fraud. Although such an idea has been around since the work of Bussard and Bagga, this is the first time that secret-sharing and terrorist fraud have been systematically studied altogether. We prove that secret sharing can counter terrorist fraud, and we detail a method that can be applied directly to most existing distance bounding protocols. We illustrate our method on the protocol of Hancke and Kuhn, yielding two variants: the threshold distance bounding (TDB) protocol and the thrifty threshold distance bounding (TTDB) protocol. We define the adversarial strategies that attempt to gain some knowledge on the prover's long-term secret, evaluate the amount of information disclosed, and determine the adversary's success probability.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*

General Terms

Security, design, theory.

Keywords

Authentication, distance-bounding, terrorist fraud, mafia fraud, secret sharing, threshold cryptography.

1. INTRODUCTION

Man-in-the-middle attacks (MITM) are effective against a wide range of protocols.

Two variants are particularly relevant when considering authentication protocols that use distance-bounding for checking proximity: *mafia fraud* and *terrorist fraud* [1].

In mafia fraud, the adversary, Eve, attempts to impersonate a legitimate prover, Alice, to a legitimate verifier, Bob, using MITM. Terrorist fraud, also known as *rental fraud* in [13], is a variant of mafia fraud where Alice helps Eve to impersonate herself to Bob. Many scenarios may justify this singular attack, e.g., Alice pays Eve for getting a perfect alibi during a crime. However, terrorist fraud can be quite hazardous for Alice if Eve is able to (re)impersonate her afterward: Eve could commit crimes pretending to be Alice. Hence, Alice allows terrorist fraud only if Eve is able to achieve a *one-time impersonation.* More precisely, we assume that Alice does not get involved in terrorist fraud if Eve may gain some advantage for future attacks.

Since the seminal works of Desmedt *et al.* on these frauds [5, 13], most of the literature [3, 8, 14, 17, 21, 22, 27, 34, 36, 37] has been dedicated to mafia fraud. Concerning terrorist fraud, the first proposal was made by Bussard and Bagga [10, 11]. Their work was later extended by Reid, Nieto, Tang, and Senadji [30]. Recently, new protocols emerged to defeat terrorist fraud [22, 37]. All these protocols use secret-sharing, even if not explicitly stated by the authors.

This paper attempts to shed more light on terrorist fraud. Our contribution is an examination of secret sharing, and more precisely, (n, k) threshold cryptography. Classical authentication [19] and most distance bounding protocols [8, 17] fail to resist terrorist fraud because all the material needed for the authentication can be supplied to the adversary, as the long-term secret key cannot be retrieved from this material. This problem can be discarded by using a threshold scheme: the authentication material consists in the n shares of an (n, k) threshold scheme. If Alice exposes any combination of k shares to Eve, the long-term secret leaks. Therefore, Eve can only obtained $k - 1$ shares from Alice.

In the following sections, we provide a method based on secret sharing that enforces security against terrorist fraud. To illustrate our method, we suggest two protocols based on Hancke and Kuhn's protocol [17], which are: *threshold distance-bounding* (TDB) and *thrifty threshold distance-bounding* (TTDB). In TDB, Alice uses a different (n, k) threshold scheme in each protocol round to answer Bob. TTDB is more thrifty than TDB in the sense that the same system of shares is used q times ($2 \leq q < k-1$) instead of only once. The success probability of impersonation in mafia fraud and in terrorist fraud are given for both protocols. Particular attention is paid to key recovery attacks which fall into two categories: birthday paradox and divide-and-conquer. The first attack is prevented by choosing carefully the size of the values exchanged at the initialization of the protocol. The second attack depends on the capability of the adversary to observe the protocol success. For this purpose, we define three different classes of adversaries. Against the weakest adversary, TDB with $k = 2$ and TTDB with $k = 3$, $q = 2$ provide the best security level. For stronger adversaries, TDB with $k = 3$ must be considered. For TTDB, the same system of shares cannot be used more than $q = \frac{k-1}{2}$ times.

Our contribution is fourfold: (1) We introduce a method based on (n, k) threshold cryptography that enforces the security of distance bounding protocols against terrorist fraud. (2) We illustrate our method by applying it to Hancke-Kuhn's protocol, yielding two variants of this protocol. (3) We refine the adversary model, introducing three classes of adversaries: BD-ADV, RES-ADV, and RD-ADV. (4) We provide a comprehensive and accurate analysis of key recovery strategies, including bounds on the disclosed information and on the number of shares needed to maintain system security.

The rest of the paper is organized as follows: Section 2 reviews distance bounding protocols, mafia and terrorist frauds, and defines the adversary capabilities. The protocols TDB and TTDB are described in Section 3. Section 4 and 5 analyze the security of our protocols against different adversaries and strategies, including impersonation and key recovery attacks. Section 4 computes the success probabilities of mafia fraud and attacks based on the birthday paradox. Section 5 introduces the post-ask strategy for key-recovery based divide-and-conquer. Finally, the advantages and weaknesses of previous works are discussed in Section 6.

2. THREAT MODEL

We now briefly review the definitions of distance-bounding protocol, mafia fraud, and terrorist fraud. Detailed explanations of these concepts can be found in [1]. Following this review, we refine the adversary model to capture its capability to mount key recovery attacks.

2.1 Definitions

DEFINITION 1 (DISTANCE-BOUNDING PROTOCOL)
A distance bounding protocol *authenticates a prover to a verifier and bounds the distance between them.*

DEFINITION 2 (NEIGHBORHOOD)
In a distance bounding protocol, the distance measurement allows the verifier to define an area, called the neighborhood, *in which the protocol execution is considered genuine.*

Three attacks exist against distance-bounding protocols: distance fraud, mafia fraud, and terrorist fraud. Distance fraud is not covered in this paper as it is not a MITM. Mafia and terrorist frauds, however, are two types of MITM that introduce an interaction between the legitimate prover, Alice, and the adversary, Eve.

DEFINITION 3 (MAFIA FRAUD)
Mafia fraud *is an attack where an adversary defeats a distance bounding protocol using a MITM between the verifier and an honest prover located outside the neighborhood.*

In mafia fraud, Alice and Bob are unaware of the presence of Eve, while the latter collaborates with Alice in a terrorist fraud.

DEFINITION 4 (TERRORIST FRAUD)
Terrorist fraud *is an attack where an adversary defeats a distance bounding protocol using a MITM between the verifier and a dishonest prover located outside the neighborhood. In this situation, the dishonest prover helps the adversary maximize her chances of a successful attack, without providing any advantage for future attacks.*

PROPOSITION 1 *Given a distance-bounding protocol, let $P_{\mathcal{M}}$ and $P_{\mathcal{T}}$ denote respectively the success probabilities of an adversary in the mafia and terrorist frauds. Then, the following inequality holds:*

$$P_{\mathcal{M}} \leq P_{\mathcal{T}}. \tag{1}$$

To justify the previous inequality, the relation between Alice and the device executing the protocol needs to be explored: we can consider a *black-box* [7] model or a *white-box* [31] model. If Alice controls the device executing the protocol at her side (white-box), she can provide Eve with information that she cannot obtain herself in mafia fraud. Otherwise (black-box), Alice can at least authorize Eve to mount a sort of "mafia fraud" in which she is aware of the attack.

DEFINITION 5 (SECURITY REGARDING TERRORIST FRAUD)
If a protocol satisfies Equation 1 with equality and $P_{\mathcal{M}} < 1$, it is considered secure against terrorist fraud.

REMARK 1 *We stress that the security of a protocol regarding terrorist fraud is strongly related to its resilience to mafia fraud. Indeed, taking into account Proposition 1 and the above definition, security with respect to terrorist fraud cannot be examined as an absolute value. The relevant value is the advantage Eve gains in mounting a terrorist fraud instead of a mafia fraud. Eve will not involve Alice if she cannot provide useful information.*

REMARK 2 *Secret values are involved during the execution of the protocol between Alice and Bob. An important requirement is that they cannot be recovered by Eve, using an attack better than an exhaustive search.*

2.2 Adversary

Eve is a man in the middle adversary with complete control of the channel between the legitimate parties. We consider three classes of adversaries depending on their capabilities to observe the protocol result:

DEFINITION 6 (BD-ADV)
The Blind-Adversary *does not learn whether the protocol succeeds.*

DEFINITION 7 (RES-ADV)
The Result-Adversary *can observe if the protocol succeeds.*

For instance, in a building access control system, the adversary knows that the protocol succeeded if the door opens.

Definition 8 (rd-adv)
The Round-Adversary *has the capability to observe the result of each round, e.g., using a side channel attack.*

The bd-adv case is in fact the hypothesis used in all the existing works related to the terrorist fraud (Section 6). However, distance bounding protocol designers should be aware that the observability of a protocol result is critical when evaluating key information leakage (Section 4 and 5). Hence, the designers should take into account stronger adversaries (res-adv and rd-adv) during the protocol conception and analysis.

3. THRESHOLD DISTANCE-BOUNDING

A brief introduction to secret-sharing and threshold cryptography are reviewed in Appendix A and are a prerequisite to understanding our protocols: tdb and ttdb.

In tdb, each round uses a different part of the long term key s and a given threshold scheme, *i.e.*, one share is used per round. In ttdb, the same part of s and its associated threshold scheme are used for several rounds. The notations used in the paper are summarized in Table 1.

Our protocols assume the use of an (n, k) threshold scheme Λ. From a secret s, n shares are computed such that any combination of k shares can be used to recover the secret. Gathering strictly less than k shares reveals no information about the secret.

N_A	nonce chosen by Alice
N_B	nonce chosen by Bob
ℓ_A	size of N_A in bits
ℓ_B	size of N_B in bits
G	a finite group
$\mathcal{R}$	$n \times m$ matrix over G
c_i	i-th challenge of Bob
$r_{i,j}$	an element of $\mathcal{R}$ (i-th row and j-th column) and a given share of Λ
m	number of rounds in tdb and ttdb
q	number of sub-rounds in ttdb and $m \vert q$
n	total number of shares
k	number of shares recovering the secret $k > 1$
$\mathcal{E}$	encryption algorithm
Λ	(n, k) threshold scheme over G
Λ'	(n, k) threshold scheme over G^q
f	pseudo-random function
$P_{\mathcal{M}}$	success probability of mafia fraud
$P_{\mathcal{T}}$	success probability of terrorist fraud
$P_{\mathcal{B}}$	success probability of birthday impersonation
P_X	success probability of impersonation knowing X elements of each column of $\mathcal{R}$
bd-adv	blind adversary
res-adv	adversary observing protocol result
rd-adv	adversary observing round results

Table 1: Notations and parameters for TDB.

3.1 The TDB scheme

tdb is similar to the protocol of Hancke and Khun [17]. It is based on a decision problem, *i.e.*, Bob's challenges are used to select the answers of Alice amongst n possible shares. In [17], $n = 2$ and the shares of Alice are created using a pseudo-random function. In tdb, the computation of Alice's answers is done differently.

Prerequisite.
Alice and Bob share a secret s viewed as a vector $(s_1, \ldots, s_m)$ of m coordinates over a group G. They can both compute an (n, k) threshold scheme Λ and a pseudo-random function f. The protocol is composed of three phases (Figure 1): initialization, interactive, and result.

Initialization phase.
Alice and Bob exchange nonces N_A and N_B of respective size ℓ_A and ℓ_B generated using a random number generator. Then, Alice and Bob compute an $n \times m$ matrix $\mathcal{R}$. The details on the computation of this matrix are given later. There are no time constraints required for this protocol phase.

Interactive phase.
Bob asks Alice at round i to send the element $r_{c_i,i}$ of $\mathcal{R}$ (c_i-th row and i-th column). Bob measures the round trip times δt_i of each exchange. The accuracy and implementation details depend on the underlying technology and are out of scope in this paper.

Result phase.
Bob declares that the protocol succeeds if the received answers $r'_{c_i,i}$ match the expected values $r_{c_i,i}$ and if all round trip time $\delta t_i < \Delta$ where Δ is a given bound used to estimate if Alice is within the neighborhood of Bob. For now, we consider only noiseless communication. This result phase is the one targeted by the adversaries res-adv and rd-adv.

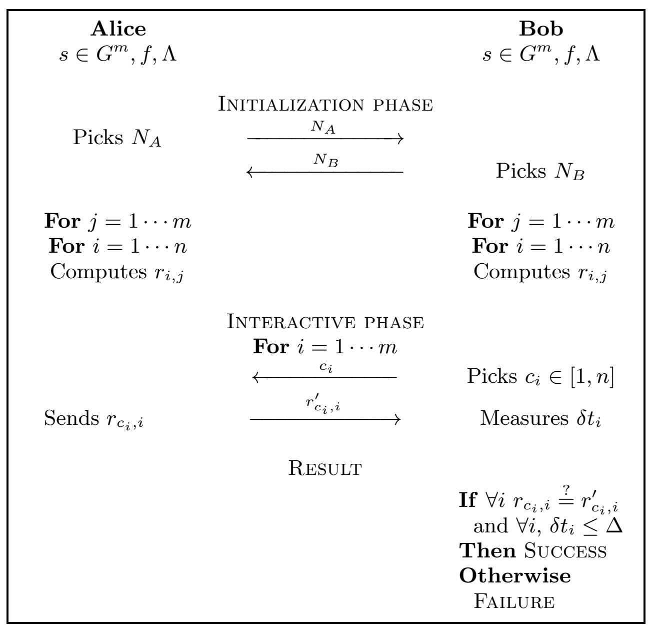

Figure 1: TDB protocol.

Matrix computation.
This is the core of tdb. Let us assume that $\mathcal{R}$ is gen-

erated randomly. It can be concluded that $I(s;\mathcal{R}) = 0$ where I denotes the mutual information, *i.e.*, the amount by which the uncertainty (entropy) of s is reduced by learning $\mathcal{R}$: $I(s;\mathcal{R}) = H(s) - H(s|\mathcal{R})$ in which $\mathcal{R}$ and s are assimilated to random variables on their respective domains and H is the Shannon's entropy. Alice can share $\mathcal{R}$ with Eve without revealing s. In order to defeat terrorist fraud, $\mathcal{R}$ must be computed such that Alice cannot reveal $\mathcal{R}$ to Eve without also leaking s, *i.e.*, $I(s;\mathcal{R}) > 0$.

The computation of $\mathcal{R}$ must satisfy two important criteria. First, the knowledge of any combination of k elements of a given column reveals a coordinate of the key. Second, Alice and Bob need to compute the same $n \times m$ matrix $\mathcal{R}$ over G. The first criterion is used to thwart the terrorist fraud and the latter is required for the RESULT phase. The matrix $\mathcal{R}$ is defined by:

$$\mathcal{R} = \begin{pmatrix} r_{1,1} & \cdots & r_{1,m} \\ \vdots & \ddots & \vdots \\ r_{n,1} & \cdots & r_{n,m} \end{pmatrix}$$

where each column $(r_{1,i}, r_{2,i}, \cdots, r_{n,i})^T$ of $\mathcal{R}$ is obtained using the (n,k) threshold scheme Λ applied on s_i. This construction is compliant with our first criterion. Indeed, we have for each column $i \in [1,m]$:

$$\begin{aligned} I(s_i;\varphi) &= 0, & \forall \varphi \in \Phi_{k-1}(r_{1,i}, \cdots, r_{n,i}) \\ I(s_i;\varphi) &= \lceil \log_2 |G| \rceil, & \forall \varphi \in \Phi_k(r_{1,i}, \cdots, r_{n,i}) \end{aligned}$$

with $\Phi_k(r_{1,i}, \cdots, r_{n,i})$ being the set of the combinations for k elements belonging to the i-th column. The previous equations are a strict interpretation of our criterion and of threshold cryptography. By using Λ for the computation of $\mathcal{R}$, Alice should not reveal to Eve more than $k-1$ elements of each column of $\mathcal{R}$. In an (n,k) threshold scheme Λ, random values are often needed and it can be problematic if Alice and Bob are not synchronized, *i.e.*, producing different random numbers in different shares. For this reason, they use a pseudo-random generator initialized with s, N_A, and N_B. A concrete study case is given with Example 1.

The computational cost of $\mathcal{R}$ depends on n, k, and $|G|$. For instance, schemes with $n = k$ are easy to implement (see Example 1). For $n \neq k$ and $G = \mathbb{F}_{2^i}$, construction using MDS codes can be obtained at the cost of multiplication by a constant in $\mathbb{F}_{2^i}$. This can be implemented very efficiently [28] using linear feedback shift registers (LFSRs).

EXAMPLE 1 *Consider the case of $n = k = 3$ and $G = \mathbb{F}_2$. Then, computing $3 \times m$ binary matrix $\mathcal{R}$ requires generating a random value of $2m$ bits. This value can be obtained using the pseudo-random function f by computing $f(s, N_A, N_B)$. These $2m$ bits represent the rows of the matrix $(r_{1,1}, \cdots, r_{1,m})$ and $(r_{2,1}, \cdots, r_{2,m})$. The last row is the sum modulo two of all other rows plus the corresponding secret bit of s. Matrix $\mathcal{R}$ becomes:*

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

The computation of $\mathcal{R}$ for any (n,m) easily follows.

REMARK 3 *The critical parameters for implementation on a radio device are n and $|G|$. Bob needs to send $\lceil \log_2 n \rceil$ bits per challenge. Alice replies with $\lceil \log_2 |G| \rceil$ bits. As the protocol does not target any given technology, we do not discuss on these values. The reader may consult [15, 16, 23, 29] for more details on this topic.*

3.2 The thrifty TDB scheme

The TDB and TTDB differ on three points: (1) the matrix computation, (2) the size of Alice's answers, and (3) additional conditions on the challenges sent by Bob.

We call this scheme thrifty because it reduces the number of systems of shares computed. First, TTDB works on vectors of q coordinates in G. An (n,k) threshold scheme Λ' compliant with this condition is used. The scheme Λ' is applied $\frac{m}{q}$ times with q a divisor of m. A column of $\mathcal{R}$ is used once in TDB, whereas in TTDB, it is used q times. Consequently, there are only $\frac{m}{q}$ distinct columns in $\mathcal{R}$. A column $(r_{1,i}, r_{2,i}, \cdots, r_{n,i})^T$ of $\mathcal{R}$ is obtained using the (n,k) threshold scheme Λ' applied on $(s_{qi-q+1}, \cdots, s_{qi})$. Each distinct column is repeated q times in the matrix. The overall number of challenges/responses is kept constant m. The resulting $n \times m$ matrix $\mathcal{R}$ over G^q is defined by:

$$\mathcal{R} = \begin{pmatrix} \overbrace{r_{1,1} \cdots r_{1,1}}^{q \text{ times}} & \cdots & \overbrace{r_{1,m/q} \cdots r_{1,m/q}}^{q \text{ times}} \\ \vdots & \ddots & \vdots \\ r_{n,1} \cdots r_{n,1} & \cdots & r_{n,m/q} \cdots r_{n,m/q} \end{pmatrix}.$$

The last difference between TDB and TTDB is how Bob generates challenges. When working on a giving distinct column of $\mathcal{R}$, the challenges c_i are not allowed to be repeated. We now define a round for TTDB as the series of q challenge/response (sub-rounds) with the same column.

As the following results will show, TTDB is a generalization of TDB for the terrorist fraud.

4. BLIND ADVERSARY

In Section 2, we saw that mafia fraud and terrorist fraud cannot be dissociated. We now show the results of our protocols against mafia fraud.

4.1 The analysis of TDB

This section describes how n and k should be chosen. The parameter n is critical regarding mafia fraud while k impacts the probability of a successful terrorist fraud. We also provide recommendations on the size of the nonces exchanged by Alice and Bob.

An important intermediate result in our analysis is the probability of a successful impersonation attack considering that Eve knows $X > 0$ elements of each column of $\mathcal{R}$. This probability, denoted P_X, is equal to:

$$P_X = \left(\frac{X}{n} + \frac{n - X}{n|G|} \right)^m$$

With this result, we derive all the probabilities needed to evaluate the security of TDB.

MAFIA FRAUD.

The probability $P_{\mathcal{M}}$ of mafia fraud is:

$$P_{\mathcal{M}} = \max\left(\left(\frac{1}{|G|}\right)^m, \left(\frac{1}{n} + \frac{n-1}{n|G|}\right)^m\right). \qquad (2)$$

Note that $\left(\frac{1}{|G|}\right)^m$ corresponds to an adversary who attempts to answer on its own. Any value is possible and is referred to in the literature as the *no-ask strategy.* The right term is the probability of success of the *pre-ask strategy.* Within this strategy, the normal initialization phase is followed by Eve executing the interactive phase with Alice using her own challenges. Eve obtains m elements of $\mathcal{R}$ from Alice (one per column). Afterwards, Eve executes the interactive phase with Bob. The success probability for the pre-ask strategy is exactly $P_{X=1}$.

Equation 2 does not take into account the capability of Eve to observe different executions of the protocol. She can exploit the *birthday paradox* [38] and the *generalized birthday paradox*[1]. When both nonces are repeated, so does the matrix $\mathcal{R}$. An X-collision is the observation by Eve of X executions of the protocol between Alice and Bob with the same values N_A and N_B. Eve needs to observe:

$$C(X) \geq (X!)^{\frac{1}{X}} \times \left(2^{\ell_A+\ell_B}\right)^{\frac{X-1}{X}}, \qquad (3)$$

to obtain an X-collision on both N_A and N_B with a probability greater than $\frac{1}{2}$ for a large value of $2^{\ell_A+\ell_B}$. This result is a direct application of the work of Suzuki, Tonien, Kurosawa, and Toyota in [35].

When the X-th collision occurs, the success probability $P_{\mathcal{B}}$ for this birthday impersonation is: $P_{\mathcal{B}} = P_X$.

The previous computation of $P_{\mathcal{B}}$ assumes that Eve has obtained X different shares for each round only from observing the protocol execution. The birthday paradox needs to be also applied on Bob's challenges. Eve will need more than an X-collision to obtain the success probability equal to P_X. Fortunately for Eve, she can circumvent this problem by using a pre-ask strategy. When a collision occurs, she executes the interactive phase with Alice using challenges not previously recorded. Then, she executes the interactive phase with Bob. To thwart such attacks, the nonce size must be chosen such that observing $C(X)$ is not feasible.

At this point, readers may be wondering about the case when $X = k$. Indeed, Eve can recover each coordinates s_i of the secret s. However, it requires $C(k)$ executions of the protocol (Equation 3). A more effective attack is possible as shown in the next paragraph.

Key recovery.

With the lesson learned by birthday impersonation, Eve can devise a key recovery attack more efficient than the one suggested previously. Instead of observing and tampering with messages between Alice and Bob during protocol execution, she directly executes the protocol with Alice. This attack is possible since the authentication is unilateral. In this way, Eve keeps the value of the nonce N_B constant and observes a k-collision with probability greater than $\frac{1}{2}$ for:

$$C'(k) \geq (k!)^{\frac{1}{k}} \times \left(2^{\ell_A}\right)^{\frac{k-1}{k}} \qquad (4)$$

[1]The term generalized birthday paradox is used abusively to describe very different problems. We refer here to multi-collisions as used in the cryptography-related literature [20]

executions of the protocol. If ℓ_A is not chosen carefully, an attack against the secret s can be more efficient than an exhaustive search. To guarantee the security level of the key s, we require:

$$\begin{aligned} C'(k) &\geq 2^m, \\ \ell_A &\geq \frac{km}{k-1} - \frac{k\log_2(k!)^{\frac{1}{k}}}{k-1}, \\ \ell_A &\geq \frac{km}{k-1}, \end{aligned}$$

since $(k!)^{\frac{1}{k}} > 1$. For $k = 2$, we need to have $\ell_A = 2m$. Choosing $k > 2$ allows the designer to reduce the size of the nonces generated and exchanged by Alice.

We claim that this attack is the only key-recovery attack available to Eve when she is a BD-ADV. She is unable to recover k elements of a column of $\mathcal{R}$ even if she observes or tampers with the protocol. This claim is more explicit when we deal with RES-ADV and RD-ADV in Section 5.

Terrorist fraud.

How many elements of a column of $\mathcal{R}$ can be safely given to Eve? In the context of BD-ADV, this value is equal to $k-1$. The reasoning behind this choice is as follows: When Bob sends a challenge c_i for which Eve knows the answer $r_{c_i,i}$, there is no risk of information leakage on s. Otherwise, Eve tries to guess the answer. If her guess is correct, she obtains enough shares to recover a coordinate of the key. However, she is unable to detect a correct guess since we are dealing with a BD-ADV. Therefore, the success probability $P_{\mathcal{T}}^{\text{BD-ADV}}$ of terrorist fraud is:

$$P_{\mathcal{T}}^{\text{BD-ADV}} = P_{X=k-1} = \left(\frac{k-1}{n} + \frac{n-k+1}{n|G|}\right)^m. \qquad (5)$$

Example 2 *Let consider* TDB *implemented with an* $(2,2)$ *threshold scheme and* $G = \mathbb{F}_2$ *(similar to the protocol of Hancke and Khun [17]). In this case, the success probability against the mafia and terrorist frauds are respectively* $P_{\mathcal{M}} = \left(\frac{3}{4}\right)^m$ *and* $P_{\mathcal{T}}^{\text{BD-ADV}} = \left(\frac{3}{4}\right)^m$. *It is secure against terrorist fraud:* $P_{\mathcal{M}} = P_{\mathcal{T}}^{\text{BD-ADV}}$ *and* $P_{\mathcal{M}} < 1$. *Indeed,* TDB *implemented with* $(n,2)$ *threshold scheme is secure against terrorist fraud for any* $n \geq 2$ *if Eve is* BD-ADV.

Working with $n = k = 2$ *means that the birthday impersonation and the key recovery attack employ directly the birthday paradox:*

$$\begin{aligned} C(2) &\approx 2^{\frac{\ell_A+\ell_B}{2}} \\ C'(2) &\approx 2^{\frac{\ell_A}{2}}. \end{aligned}$$

When a collision is observed, $P_{\mathcal{B}} = 1$. *Alice needs to choose a nonce* N_A *of length* $\ell_A = 2m$ *to guarantee the security level of her key* s.

4.2 The analysis of TTDB

The security analysis of TTDB is essentially the same as the one for TDB. The main difference is the computation of P_X.

In order to compute the success probability of mafia and terrorist frauds, we first analyze the success probability of an impersonation given that Eve knows X shares in each round, with $X > 0$.

Let consider a given round r_w ($1 \leq w \leq \frac{m}{q}$). The analysis becomes tricky because the sub-rounds of r_w are not independent. Given a bit string B of length q, we define the events $\mathcal{A}_B$ that Eve succeeds in round r_w and Bob asks her known shares when the bits of B are equal to 1. By varying B in $\mathbb{F}_2^q$, we cover all the possible sequences of challenges. As the $\mathcal{A}_B$s are pairwise disjoint, we deduce the success probability of Eve impersonating Alice regarding Bob in the round r_w:

$$\Pr(\text{succ } r_w) = \sum_{B \in \mathbb{F}_2^q} \Pr(\mathcal{A}_B). \tag{6}$$

Now, we define the function $f_{X,B}(i) : \{0, \dots, q-1\} \mapsto [0,1]$, by:

$$\frac{1}{n-i} \cdot \begin{cases} X - \sum_{j=0}^{j=i} B_j + 1 & \text{if } B_i = 1, \\ \left((n-i) - \left(X - \sum_{j=0}^{j=i} B_j\right)\right) \cdot \frac{1}{|G|^q} & \text{otherwise,} \end{cases}$$

where B_j is the jth bit of B. $f_{X,B}(i)$ represents the success probability of Eve in the $(i+1)$th sub-round. Indeed, two cases occur (a) Bob asks for a known share, this happens with probability $\frac{X - \sum_{j=0}^{j=i} B_j + 1}{n-i}$ and Eve wins with probability 1. Or (b) he does not and Eve has to guess the answer, so she succeeds with probability $\frac{1}{|G|^q}$. By definition of $f_{X,B}$:

$$\Pr(\mathcal{A}_B) = \prod_{i=0}^{i=q-1} f_{X,B}(i). \tag{7}$$

Hence, Equations 6, 7, yield to:

$$\Pr(\text{succ } r_w) = \sum_{B \in \mathbb{F}_2^q} \left(\prod_{i=0}^{i=q-1} f_{X,B}(i) \right).$$

Finally, by noticing that the rounds are independent, we find the probability of a successful impersonation given that Eve knows X shares in each round, P_X, as:

$$P_X = \left(\sum_{B \in \mathbb{F}_2^q} \left(\prod_{i=0}^{i=q-1} f_{X,B}(i) \right) \right)^{\frac{m}{q}} \tag{8}$$

Mafia fraud.

The success probability for mafia fraud is the maximum between the no-ask and pre-ask strategy:

$$P_{\mathcal{M}} = \max\left(\left(\frac{1}{|G|^q}\right)^m, \ P_{X=q} \right),$$

We compute the success probability of a birthday impersonation given that Eve has observed a x-collision as:

$$P_{\mathcal{B}} = P_{X=qx}.$$

Indeed, Eve learns q elements of column of $\mathcal{R}$ at each round and this for every collision she observed. $C(X)$ remains the same (Equation 3).

Key recovery.

Eve uses the same key recovery as in TDB. She executes the protocol with Alice with a fixed N_B. However, Eve collects more information during each collision with TTDB than with TDB. Indeed, a collision exposes q shares with TTDB for only one with TDB. Thus, we need:

$$C'\left(\left\lceil \frac{k}{q} \right\rceil\right) \geq 2^m$$

which implies from Equation 4:

$$\ell_A \geq m \cdot \left(1 + \frac{k}{q}\right),$$

considering $\lceil \frac{k}{q} \rceil \leq \frac{k}{q} + 1$ and Equation 4.

Terrorist fraud.

For each distinct column of $\mathcal{R}$, Alice can provide $k-1$ elements to Eve without revealing the key. We have:

$$P_{\mathcal{T}}^{\text{BD-ADV}} = P_{X=k-1}. \tag{9}$$

Example 3 *Let consider TTDB implemented with an $(3,3)$ threshold scheme, $q = 2$ and $|G| = 4$. The size of Alice's nonce is $\ell_A \approx 2m$. Using the previous equations, we have the following computations for $P_{X=2}$:*

$$\begin{aligned} P_{X=2} &= \left(\frac{2}{3}\cdot\frac{1}{2}+\frac{2}{3}\cdot\frac{1}{2}\cdot\frac{1}{16}+\frac{1}{3}\cdot\frac{1}{16}\cdot 1+\frac{1}{3}\cdot\frac{1}{16}\cdot 0\cdot\frac{1}{16}\right)^{\frac{m}{2}} \\ &= \left(\frac{3}{8}\right)^{\frac{m}{2}} = \left(\frac{\sqrt{3}}{2\sqrt{2}}\right)^m. \end{aligned}$$

Thus, the success probability against mafia and terrorist frauds are respectively $P_{\mathcal{M}} = \left(\frac{\sqrt{3}}{2\sqrt{2}}\right)^m$ and $P_{\mathcal{T}}^{\text{BD-ADV}} = \left(\frac{\sqrt{3}}{2\sqrt{2}}\right)^m$. Hence, this scheme is secure against terrorist fraud: $P_{\mathcal{M}} = P_{\mathcal{T}}^{\text{BD-ADV}}$ and $P_{\mathcal{M}} < 1$. This property is verified for TTDB if $q = k-1$.

5. STRONGER ADVERSARIES

Apart from the birthday results, the analysis found in Section 4 is the one used by the existing literature, and is limited to BD-ADV. Our analysis goes a step further with RES-ADV and RD-ADV and we see drastically different results. This section is dedicated to key recovery attacks pertaining the terrorist fraud.

5.1 The analysis of TDB

In Section 4, we concluded that TDB with $(n,2)$ threshold cryptography is enough to defeat terrorist fraud. We now show under the same assumptions that Eve can recover the secret s using a *post-ask strategy* in RES-ADV and RD-ADV.

Post-ask strategy.

This strategy was originally designed to carry out mafia fraud against Brands and Chaum's protocol [8]. This protocol differs from Hancke and Khun [17] by requiring that Alice compute a signature over all received challenges c_i and all answers $r_{c_i,i}$ at the end of the protocol. She sends this signature to Bob for verification in addition to the initialization and interactive phases. The idea of the post-ask strategy is to force Alice to generate the final signature. When Eve executes the interactive phase with Bob, she must also execute the interactive phase with Alice and forward the legitimate challenges. At the end, Alice transmits the correct signature and Eve relays it to Bob. Eve has "only" to succeed in the interactive phase with Bob. She does not solve a cryptographic problem.

The use of this strategy may look dubious on TDB for Eve at a first sight: TDB and TTDB are not using any final signature. However, a slight modification of this attack results in learning two elements of each column of $\mathcal{R}$ for each protocol round.

First consider the case of RD-ADV and the attack described in Figure 2. Eve executes the interactive phase with Bob. The challenge c_i is answered by $\hat{r}_{c_i,i}$. She observes the result of each round. If a round succeeds, then $\hat{r}_{c_i,i} = r_{c_i,i}$. She is now half way to recovering the secret key. Now, Eve sends her own challenges $\hat{c}_i$ to Alice such that they all differ from the legitimate ones, *i.e.*, $\forall i\ , c_i \neq \hat{c}_i$. By doing so, Eve is guaranteed to obtain legitimate elements from $\mathcal{R}$ which are not expected by Bob. If a given round i succeeds, Eve has two distinct elements, $r_{\hat{c}_i,i}$ and $r_{c_i,i}$, of the same column of $\mathcal{R}$. With two shares, she can recover the corresponding coordinate s_i. On average $\frac{|G|}{2}$ post-ask attacks are needed to recover the whole secret s.

Now consider the case of RES-ADV. Eve allows the interactive phase to be carried out correctly except for a single round i. In this round, Eve modifies the challenge c_i sent by Bob. Alice receives $\hat{c}_i \neq c_i$. Then, Eve records the answer of Alice $r_{\hat{c}_i,i}$ and sends a random answer $\hat{r}_{c_i,i}$ to Bob. If the protocol succeeds, Eve knows that $\hat{r}_{c_i,i} = r_{c_i,i}$. Otherwise, she knows two elements of the same column $r_{c_i,i}$ and $r_{\hat{c}_i,i}$. She can recover the corresponding coordinate s_i of the secret. Eve needs on average $\frac{|G|}{2}$ executions of this attack to recover s_i. Repeating this process m times, once per coordinate, she recovers s using a typical divide-and-conquer strategy.

The main difference between RD-ADV and RES-ADV is that RD-ADV can work on all rounds in parallel. Eve is limited to a single round per attack with RES-ADV.

EXAMPLE 4 *Recall Example 2, i.e., $n = k = 2$ and $G = \mathbb{F}_2$. The attack described in Figure 2 can be obviously applied. However, there is a much simpler strategy for Eve to recover two shares at each round. It exploits the fact that $|G| = 2$. The post-ask attack given in Figure 2 can be replaced by a fault attack: Eve changes one or all the challenges of Bob.*

- *In* RES-ADV, *Eve changes only one challenge c_i. She knows that $r_{0,i} = r_{1,i}$ if the protocol succeeds. Otherwise, she concludes that $r_{0,i} \neq r_{1,i}$ and she observes either $r_{0,i}$ or $r_{1,i}$. After m executions of the protocol with one fault at a different round, she recovers the whole secret s. This strategy was first unveiled by Kim et al. in [22] against the protocol of Tu and Piramuthu [37].*
- *In* RD-ADV, *Eve can flip all the challenge bits and recover the secret s with only one execution of the protocol.*

For $|G| > 2$, this fault attack only reveals equality between the two shares. If all shares are different, this fault attack does not help Eve. This is most likely if $n \ll |G|$ (this is yet another application of the birthday paradox). The post-ask strategy directly attempts to recover two shares and is not affected by this problem.

We can conclude that TDB cannot be used with $(n, 2)$ threshold scheme if Alice has to deal with RES-ADV or RD-ADV. TDB is weak for $k = 2$ because Eve can recover two shares at each round. The parameter k must be chosen as $k = \alpha + 1$ where α is the maximum number of shares that can be recovered by Eve in a given round. For TDB, we have shown with the previous attack that $\alpha = 2$. So, $k \geq 3$ is a safe choice for TDB. If we refer to the result given for terrorist fraud in Section 4 (Equation 5), $k \geq 3$ implies that:

$$P_{\mathcal{M}} < P_{\mathcal{T}}^{\text{BD-ADV}}.$$

So, it seems that our solution is not secure against terrorist fraud. However, the analysis of the terrorist fraud has to be re-computed taking into account the new capabilities of Eve (RES-ADV and RD-ADV).

TERRORIST FRAUD.

For now, $k \geq 3$ is only considered. In BD-ADV, it was assumed that Alice can provide $k-1$ shares for each round of the protocol without revealing the secret s. Let us assume that Alice provided $k-1$ elements of each column of $\mathcal{R}$. Each time, Alice has not provided the legitimate answer $r_{c_i,i}$ to Eve, Eve sends $\hat{r}_{c_i,i}$. If the round succeeds $\hat{r}_{c_i,i} = r_{c_i,i}$, she recover s_i. Otherwise, she eliminates a possible value for s_i. On average, $\frac{|G|}{2}$ attempts are needed on a round for which Eve does not know the answer. Moreover, Eve with RD-ADV capabilities can explore on average $\frac{m}{2}$ rounds in parallel.

In the case of RES-ADV, Eve obtains less information. If the terrorist fraud is successful, she obtains on average $\frac{m}{2}$ coordinates of the key. Otherwise, Eve has chosen i coordinates for the secret s (on average $\frac{m}{2}$). These coordinates are incorrect since the protocol fails. The $|G|^{m-i}$ secrets with these coordinates can be eliminated by Eve. Several executions of terrorist fraud result in an exploration of the key space faster than exhaustive search. The same reasoning holds for mafia fraud.

Therefore, it is wiser for Alice in the terrorist fraud to expose to Eve only $k-2$ shares at each round. When Eve succeeds in terrorist fraud, she obtains for some columns of $\mathcal{R}$ $k-1$ elements. This is not enough to recover the corresponding coordinates of the key s_i. The probability of successful terrorist fraud becomes:

$$P_{\mathcal{T}}^{\text{RES-ADV}} = P_{\mathcal{T}}^{\text{RD-ADV}} = \left(\frac{k-2}{n} + \frac{n-k+2}{n|G|}\right)^m. \qquad (10)$$

REMARK 4 *In the context of* RES-ADV *and* RD-ADV, TDB *used with $(n, 3)$ threshold schemes is secure against the terrorist fraud since we have for $k = 3$ and $|G| \geq 2$:*

$$\forall n \geq 3,\ P_{\mathcal{T}}^{\text{RES-ADV}} = P_{\mathcal{T}}^{\text{RD-ADV}} = P_{\mathcal{M}}.$$

5.2 The TTDB analysis

Fundamentally, the attacks against TTDB are identical to the ones used against TDB in RES-ADV and RD-ADV. The only modification is the overall number of shares recovered by Eve. For TDB, the post-ask helps to recover at most $\alpha = 2$ shares and explains why $(n, 3)$ schemes are safe. For TTDB, Eve can recover at most $\alpha = 2q$ shares with the same method. Therefore, $(n, 2q+1)$ threshold schemes ensure that TTDB never reveals enough information to the adversary.

Consequently, Alice can only reveal q shares to Eve. The probability of terrorist fraud is:

$$\forall n \geq 3 \text{ and } q \geq 1,\ P_{\mathcal{T}}^{\text{RES-ADV}} = P_{\mathcal{T}}^{\text{RD-ADV}} = P_{\mathcal{M}}.$$

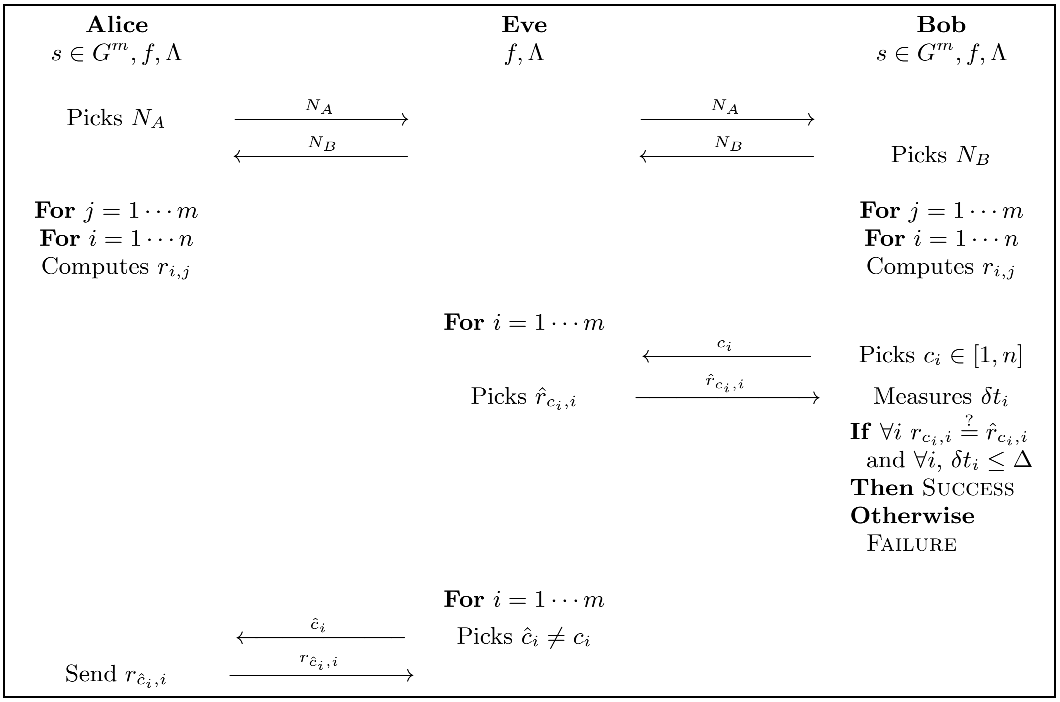

Figure 2: Post-ask attack for RD-ADV against TDB.

6. RELATED WORKS

Although existing work is primarily based on (2,2) threshold schemes, individual constructions use matrix computation based on encryption, or, introduce a challenge verification step. We consider both approaches by studying, respectively, the protocols of Reid *et al.* [30], and Kim *et al.* [22].

6.1 Matrix computation based on encryption

This class of protocols is derived from the first solution to terrorist fraud by Bussard and Bagga [11]. This solution based on asymmetric encryption was adapted by Reid *et al.* [30] to symmetric encryption. The protocol is described in Figure 3.

Let us summarize the main differences between the protocol of Reid *et al.* [30] and TDB with respect to our notations. The protocol of Reid *et al.* uses an $2 \times m$ matrix $\mathcal{R}$ over $G = \mathbb{F}_2$. Let denote r_1 and r_2 the rows of $\mathcal{R}$. The first row r_1 is obtained using the pseudo random function f: $r_1 = f(s, A, B, N_A, N_B)$. The second row is obtained by encrypting s with r_1: $r_2 = \mathcal{E}_{r_1}(s)$. For the choice of $\mathcal{E}$, Reid *et al.* gave, in the early version of their paper [30], the following comment (with adapted notation for consistency):

$\mathcal{E}$ is a semantically secure encryption function, i.e. an adversary does not learn any (computational) information about the plaintext. In practice, because the strings to be encrypted are short and the key varies for each run of the protocol, we can use a one-time pad, i.e $\mathcal{E}_{r_1}(s) = s \oplus r_1$.

If $\mathcal{E}$ is the one-time pad, then we are exactly in the setup of Example 2. The protocol of Reid *et al.* is an instance of TDB with a $(2, 2)$ threshold scheme and $G = \mathbb{F}_2$. Indeed, the additive cipher is the basic tool used to design (n, n) threshold scheme (see Appendix A). Section 5 has shown that such a scheme is not secure against the post-ask strategy for RES-ADV and RD-ADV.

If $\mathcal{E}$ is a block cipher or an asymmetric encryption scheme, the attack remains the same for RD-ADV. However, the case of RES-ADV is more difficult. Eve recovers $\frac{m}{2}$ bits of the key r_1 used to encrypt s and $\frac{m}{2} + 1$ of the ciphertext r_2. However, this information alone cannot be used to recover the secret s when $\mathcal{E}$ is a pseudo-random permutation. Otherwise, the information recovered may depend on the cipher characteristics. The additional cost of using a block cipher or an asymmetric encryption scheme must also be taken into account.

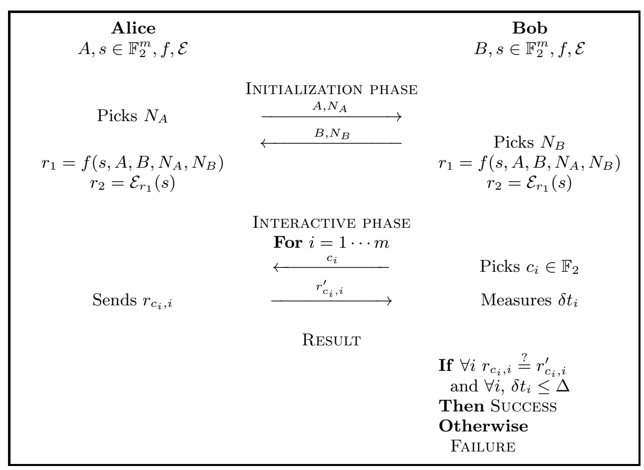

Figure 3: Reid *et al.*'s protocol. The use of the identifier A and B is made to solve a complexity issue at Bob's side [2]. They are omitted in TDB and TTDB for the sake of simplicity.

6.2 Challenge verification and $(2, 2)$ threshold

Amongst all the solutions to terrorist fraud, the Swiss-Knife RFID Distance Bounding Protocol [22] is particularly interesting. It uses an $(2, 2)$ threshold scheme that is combined with a challenge verification step. An extended analysis of this protocol is given below. The protocol is depicted in Figure 4.

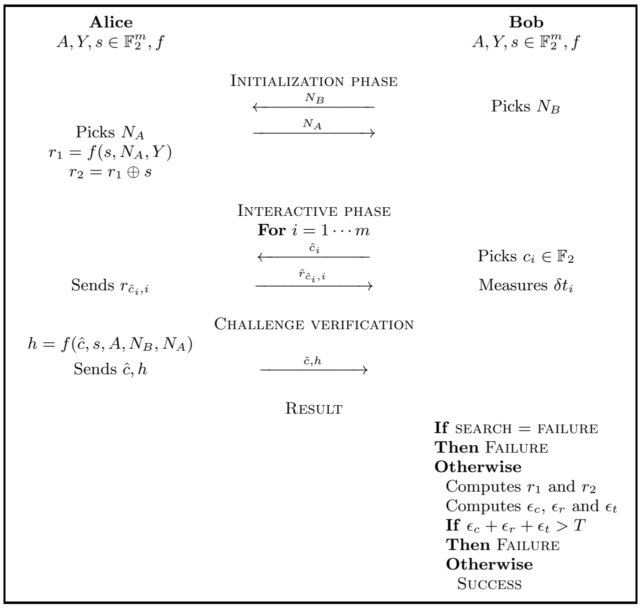

Figure 4: Kim *et al.*'s protocol.

Prerequisite.

Alice has an identifier A and a secret key $s \in \mathbb{F}_2^m$. Bob knows a database $\mathbb{DB}$ which consists of pair of the form (key,identifier). The pair (s, A) is included in Bob's database. Alice and Bob also share a constant Y. They can both compute a pseudo-random function f and a $(2, 2)$ threshold scheme.

Initialization phase.

Bob and Alice pick respectively the nonces N_B and N_A. Both nonces are exchanged. Then, Alice computes the $2 \times m$ matrix $\mathcal{R}$ over $\mathbb{F}_2$. The first row of the matrix is $r_1 = f(s, Y, N_A)$. The second row r_2 is given by $r_2 = r_1 \oplus s$.

Interactive phase.

At each of the m rounds of this phase, Bob picks randomly a challenge $c_i \in \mathbb{F}_2$. Alice received the challenge $\hat{c}_i$ and replies with $r_{\hat{c}_i,i}$. Bob receives $\hat{r}_{\hat{c}_i,i}$ and measures the timing of the round δt_i.

Challenge verification.

Alice computes a signature of the received challenge $\hat{c}_i$:

$$h = f(\hat{c}_1, \cdots, \hat{c}_m, A, N_A, N_B).$$

Then, Alice sends to Bob the vector $\hat{c} = (\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_m)$ and h. A similar phase is also found in the protocol of Bussard and Bagga [11].

Result.

Bob first needs to recover the identity of the prover. He performs an exhaustive search on his database $\mathbb{DB}$ (search function) to match the value h using $\hat{c}$, Y, N_A and N_B. If this search is unsuccessful, the protocol fails. Otherwise, Bob recovers the pair (s, A). He computes the matrix $\mathcal{R}$ as done previously by Alice, along with the following quantities:

- ϵ_c the number of positions for which $c_i \neq \hat{c}_i$,
- ϵ_r the number of positions for which $c_i = \hat{c}_i$ but $r_{c_i,i} \neq \hat{r}_{c_i,i}$,
- ϵ_t the number of positions for which $c_i = \hat{c}_i$, $r_{c_i,i} = \hat{r}_{c_i,i}$ but $\delta t_i > \Delta$.

If $\epsilon_c + \epsilon_r + \epsilon_t \geq T$ the authentication fails. Otherwise, it succeeds.

Remark 5 *Kim et al. propose a variant of this protocol in which the number of rounds is smaller than the key size m. At the beginning of each instance of the protocol, Bob picks a random mask of m bits and Hamming weight w. This mask is used to select w bits of the key s. The interactive phase consists of w rounds. This does not affect our analysis.*

Post-ask strategy.

Assume that $T = 0$, *i.e.*, no error is tolerated. This scheme defeats the post-ask attack described in Section 5 for res-adv but not for rd-adv. For res-adv, the attack is detected by the ϵ_c variable and the protocol can never succeed. Eve can attempt to bypass this problem by forging a valid signature for the legitimate challenges c_i. We assume that this cryptographic task cannot be afforded by Eve.

If $T > 1$, Eve can manipulate T challenges. However, the values sent by Eve are not taken into account when determining if the protocol succeeds or not. When $c_i \neq \hat{c}_i$, the corresponding answer is discarded.

Terrorist fraud.

This protocol uses a $(2, 2)$ threshold scheme and, as pointed out in section 4, does not leak any information to a bd-adv. We focus our discussion on res-adv and rd-adv.

In order to mount a terrorist fraud, Eve first relays the initialization phase. Then she asks Alice for a row of the matrix $\mathcal{R}$, without loss of generality we assume that it is the first row. After the interaction phase with Bob, she transmits to Alice the challenges she received. Alice computes the signature, and sends it to Eve who ends the protocol with Bob by transmitting him this signature.

A res-adv Eve, capable of detecting protocol success, gains information about the secret. Indeed, if the protocol succeeds, she knows the answers given to Bob were all correct. Thus, for the answers coming from the second row of $\mathcal{R}$, Eve learns the corresponding secret bits. We conclude that Alice should never help Eve.

Finally, in the case of the rd-adv, Eve knows whether the round succeeds or not. Hence, when Bob asks her for a second row element, she is able to determine the expected answer, and so she retrieves the corresponding secret bits. The conclusion is Alice must absolutely not provide any help to Eve.

7. CONCLUSION

We demonstrated in this paper that using threshold cryptography thwarts terrorist fraud. Previously proposed distance bounding protocols using a $(2, 2)$ threshold scheme do not resist to terrorist fraud with powerful adversaries. Our results show that, at least, a $(3, 3)$ threshold scheme should be used. We illustrated our results on the protocol of Hancke and Kuhn, yielding two variants: the threshold distance bounding (TDB) protocol and the thrifty threshold distance bounding (TTDB) protocol. We refined the adversary model, introducing three classes of adversaries: BD-ADV, RES-ADV, and RD-ADV. We provided an accurate analysis of our protocols, including the adversary's success probabilities. Finally, we applied our adversarial model to previous works, and highlighted their weaknesses.

Acknowledgement

The authors want to thank the reviewers for their comments and particularly John Solis for his help to clarify this work.

This work is partially funded by the Walloon Region Marshall plan through the SPW DG06 Project TRASILUX. Benjamin Martin was supported by a grant from Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture (FRIA), rue d'Egmont 5, 1000 Brussels, Belgium.

8. REFERENCES

[1] G. Avoine, M. A. Bingöl, S. Kardaş, C. Lauradoux, and B. Martin. A Framework for Analyzing RFID Distance Bounding Protocols. *Journal of Computer Security – Special Issue on RFID System Security*, 2010.

[2] G. Avoine, E. Dysli, and P. Oechslin. Reducing Time Complexity in RFID Systems. In *Selected Areas in Cryptography – SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 291–306, Kingston, Canada, August 2005. Springer-Verlag.

[3] G. Avoine, C. Floerkemeier, and B. Martin. RFID Distance Bounding Multistate Enhancement. In *International Conference on Cryptology in India - Indocrypt 2009*, volume 5922 of *Lecture Notes in Computer Science*, pages 290–307. Springer-Verlag, 2009.

[4] S. Bengio, G. Brassard, Y. Desmedt, C. Goutier, and J.-J. Quisquater. Secure implementation of identification systems. *Journal of Cryptology*, 4(3):175–183, 1991.

[5] T. Beth and Y. Desmedt. Identification Tokens - or: Solving the Chess Grandmaster Problem. In *Advances in Cryptology - CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 169–177, Santa Barbara, CA, USA, August 1990. Springer-Verlag.

[6] G. R. Blakley. Safeguarding cryptographic keys. In *AFIPS 1979 National Computer Conference*, volume 48, pages 313–317, Arlington, NY, USA, 1979–317.

[7] M. Blaze. Looking on the Bright Side of Black-Box Cryptography (Transcript of Discussion). In *Security Protocols Workshop*, volume 2133 of *Lecture Notes in Computer Science*, pages 54–61, Cambridge, UK, April 2000. Springer-Verlag.

[8] S. Brands and D. Chaum. Distance-Bounding Protocols. In *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359, Lofthus, Norway, May 1993. Springer-Verlag.

[9] E. F. Brickell and D. M.Davenport. On the classification of ideal secret sharing schemes. *Journal of Cryptology*, 4:123–134, 1991.

[10] L. Bussard. *Trust Establishement Protocols for Communications Devices*. PhD thesis, Eurecom-ENST, 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*, volume 181 of *IFIP International Federation for Information Processing*, pages 223–238. Springer-Verlag, 2005.

[12] L. Csirmaz. The Size of a Share Must Be Large. In *Advances in Cryptology - EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 13–22, Perugia, Italy, 1994. Springer-Verlag.

[13] Y. Desmedt, C. Goutier, and S. Bengio. Special Uses and Abuses of the Fiat-Shamir Passport Protocol. In *Advances in Cryptology - CRYPTO'87*, volume 293 of *Lecture Notes in Computer Science*, pages 21–39, Santa Barbara, CA, USA, August 1988. Springer-Verlag.

[14] S. Drimer and S. J. Murdoch. Keep your enemies close: distance bounding against smartcard relay attacks. In *16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–16, Santa Clara, CA, USA, June 2007. USENIX Association.

[15] M. Flury, M. Poturalski, P. Papadimitratos, J.-P. Hubaux, and J.-Y. L. Boudec. Effectiveness of distance-decreasing attacks against impulse radio ranging. In *ACM Conference on Wireless Network Security - WISEC 2010*, pages 117–128, Hoboken, NJ, USA, 2010. ACM.

[16] G. P. Hancke. Design of a Secure Distance-Bounding Channel for RFID. *Journal of Network and Computer Applications*, May 2010.

[17] G. P. Hancke and M. Kuhn. An RFID Distance Bounding Protocol. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*, pages 67–73, Athens, Greece, September 2005. IEEE Computer Society.

[18] G. P. Hancke and M. G. Kuhn. Attacks on time-of-flight distance bounding channels. In *ACM Conference on Wireless Network Security - WISEC 2008*, pages 194–202, Alexandria, VA, USA, March 2008. ACM.

[19] International Organization for Standardization. ISO/IEC 9798 – Information technology – Security techniques – Entity authentication, 1997 – 2008.

[20] A. Joux. Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions. In *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 306–316, Santa Barbara, CA, USA, August 2004. Springer-Verlag.

[21] C. H. Kim and G. Avoine. RFID distance bounding protocol with mixed challenges to prevent relay attacks. In *International Conference on Cryptology and Network Security - CANS*, volume 5888 of *Lecture*

Notes in Computer Science, pages 119–133, Kanazawa, Ishikawa, Japan, December 2009. Springer-Verlag.

[22] C. H. Kim, G. Avoine, F. Koeune, F.-X. Standaert, and O. Pereira. The Swiss-Knife RFID Distance Bounding Protocol. In *International Conference on Information Security and Cryptology – ICISC'08*, volume 5461 of *Lecture Notes in Computer Science*, pages 98–115, Seoul, Korea, December 2008. Springer-Verlag.

[23] M. Kuhn, H. Luecken, and N. O. Tippenhauer. UWB Impulse Radio Based Distance Bounding. In *Workshop on Positioning, Navigation and Communication 2010 - WPNC'10*, Dresden, Germany, March 2010.

[24] J. L. Massey. Minimal Codewords and Secret Sharing. In *Proceedings of the 6th Joint Swedish-Russian International Workshop on Information Theory*, pages 276–279, 1993.

[25] R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Communication of the ACM*, 24(9):583–584, 1981.

[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, July 2010.

[27] 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(9):1227–1232, 2008.

[28] C. Paar. *Efficient VLSI Architectures for Bit Parallel Computation in Galois Fields*. PhD thesis, Universität GH Essen, 1994.

[29] K. B. Rasmussen and S. Čapkun. Realization of RF Distance Bounding. In *USENIX Security Symposium*, Washington, DC, USA, August 2010.

[30] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji. Detecting relay attacks with timing-based protocols. In *ACM symposium on Information, computer and communications security - ASIACCS '07*, pages 204–213. ACM, 2007. Early version available at citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.70.5584.

[31] A. Saxena, B. Wyseur, and B. Preneel. Towards Security Notions for White-Box Cryptography. In *Information Security Conference- ISC 2009*, volume 5735 of *Lecture Notes in Computer Science*, pages 49–58, Pisa, Italy, September 2009. Springer-Verlag.

[32] A. Shamir. How to share a secret. *Communication of the ACM*, 22(11):612–613, 1979.

[33] G. J. Simmons. *Contemporary Cryptology: The Science of Information Integrity*. IEEE Press, 1991.

[34] D. Singelée and B. Preneel. Distance Bounding in Noisy Environments. In *European Workshop on Security in Ad-hoc and Sensor Networks - ESAS'07*, volume 4572 of *Lecture Notes in Computer Science*, pages 101–115, Cambridge, UK, July 2007. Springer-Verlag.

[35] K. Suzuki, D. Tonien, K. Kurosawa, and K. Toyota. Birthday Paradox for Multi-collisions. In *Information Security and Cryptology - ICISC 2006*, volume 4296 of *Lecture Notes in Computer Science*, pages 29–40, Busan, Korea, November 2006. Springer-Verlag.

[36] R. Trujillo Rasua, B. Martin, and G. Avoine. The Poulidor Distance-Bounding Protocol. In S. O. Yalcin, editor, *Workshop on RFID Security – RFIDSec'10*, volume 6370 of *Lecture Notes in Computer Science*, pages 239–257, Istanbul, Turkey, June 2010. Springer-Verlag.

[37] Y.-J. Tu and S. Piramuthu. RFID Distance Bounding Protocols. In *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.

[38] D. Wagner. A Generalized Birthday Problem. In *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 288–303, Santa Barbara, CA, USA, August 2002. Springer-Verlag.

APPENDIX

A. SECRET SHARING

A secret-sharing scheme Λ allows a user to share a secret s amongst n participants according to an access control list Γ. Γ determines the subset of participants that are allowed to recover s. The parameter k of a secret-sharing scheme is the size of the smallest subset of participants that can recover the secret. The first solution for achieving secret-sharing was the threshold cryptography [6, 32]. In an (n, k) threshold scheme, any subset of k participants can recover s. Threshold schemes were first implemented by Shamir [32] using interpolation problem. It was subsequently re-interpreted by Sarwate and McEliece [25] in terms of Reed-Solomon codes. Several important results for secret-sharing coming from coding theory have followed [24]. To conclude this overview of secret-sharing, the classical construction of (n, n) threshold schemes is given. This textbook example is particularly useful since the existing works used $(2, 2)$ schemes and $(3, 3)$ schemes.

Example 5 *Consider an additive group $(G, +)$ and a secret $s \in G$. To construct an (n, n) threshold scheme, the owner of s chooses randomly $n-1$ shares $s_i \in G$, $i \in [1, n-1]$. The last share s_n is defined by $s_n = s - \sum_{i=1}^{n-1} s_i$. Knowing all shares, one can compute $s = \sum_{i=1}^{n} s_i$. Knowing strictly less than n shares does not provide information about s.*

The size of shares in a secret sharing scheme is also an important problem. More details on this problem can be found in [9, 12, 33].