Advance Access publication on 4 April 2013

doi:10.1093/comjnl/bxt033

On Selecting the Nonce Length in Distance-Bounding Protocols

AIKATERINI MITROKOTSA[1,2,*], PEDRO PERIS-LOPEZ[3], CHRISTOS DIMITRAKAKIS[1] AND SERGE VAUDENAY[1]

[1]*EPFL, Lausanne, Switzerland*
[2]*University of Applied Sciences of Western Switzerland (HES-SO), Geneva, Switzelrand*
[3]*Carlos III University of Madrid, Madrid, Spain*
[*]*Corresponding author: mitrokatkm@gmail.com*

Distance-bounding protocols form a family of challenge–response authentication protocols that have been introduced to thwart relay attacks. They enable a verifier to authenticate and to establish an upper bound on the physical distance to an untrusted prover. We provide a detailed security analysis of a family of such protocols. More precisely, we show that the secret key shared between the verifier and the prover can be leaked after a number of nonce repetitions. The leakage probability, while exponentially decreasing with the nonce length, is only weakly dependent on the key length. Our main contribution is a high probability bound on the number of sessions required for the attacker to discover the secret, and an experimental analysis of the attack under noisy conditions. Both of these show that the attack's success probability mainly depends on the length of the used nonces rather than the length of the shared secret key. The theoretical bound could be used by practitioners to appropriately select their security parameters. While longer nonces can guard against this type of attack, we provide a possible countermeasure which successfully combats these attacks even when short nonces are used.

Keywords: RFID; distance-bounding protocols; relay attacks; high probability bounds; cryptanalysis

Received 4 February 2012; revised 19 February 2013
Handling editor: Jong Hyuk Park

1. INTRODUCTION

Relay attacks enable an adversary to act as man-in-the-middle and fool a legitimate verifier [e.g. radio frequency identification (RFID) reader] into thinking that he is in close proximity. Relay attacks can be mainly discriminated into the following three categories: *distance fraud*, *mafia fraud* and *terrorist fraud* attacks [1].

(1) The *distance fraud* attack involves a malicious prover Q who is trying to convince a legitimate verifier V of being nearer to V than he really is.
(2) In the *mafia fraud* attack, the adversary A is interacting with a legitimate verifier V and a legitimate prover P. The goal is to shorten the distance between the legitimate prover P and the verifier V. The adversary achieves that by convincing V that he is communicating with the legitimate prover P while in reality he communicates with the adversary A.
(3) The *terrorist fraud* attack involves again an adversary A interacting with a legitimate verifier V and a legitimate but dishonest prover P'. Again the goal is to shorten the distance between the verifier V and the dishonest but legitimate prover P'. Nevertheless, in this case P' collaborates with A but without revealing the secret key shared between V and P'.

Distance-bounding protocols were first introduced by Brands and Chaum [2] to preclude *distance fraud* and *mafia fraud* attacks. More precisely, they proposed a mechanism to infer an *upper bound* of the distance between the verifier and the prover by measuring the round trip delay during a rapid challenge–response bit exchange of n rounds. Around 15 years later, Hancke and Kuhn [3] proposed a distance-bounding protocol in the context of RFID technology which may be considered a seminal paper in this research area. Later, Munilla and Peinado [4] proposed a protocol inspired by Hancke and Kuhn [3] in which the success probability of an adversary to accomplish a mafia fraud attack is reduced. However, the feasibility of this scheme is questionable since it requires three physical states {0, 1, void}. Singelée and Preneel [5] proposed a distance-bounding

protocol which uses an error correction code to facilitate the correction of errors (in noisy channels) during the *rapid bit exchange* (RBE) phase. Nevertheless, this scheme's security and implementation cost on RFID tags is questioned in [6].

The above-mentioned protocols do not address *terrorist fraud* attacks. In 2007, Tu and Piramuthu [7] addressed both terrorist and mafia fraud attacks and proposed an enhancement scheme. The authors used ideas previously presented in [8] to prevent terrorist attacks. Kapoor *et al.* [9] have extended Tu's and Piramuthu's protocol to the case where multiple tags are authenticated for their simultaneous presence in the field of an RFID reader. Nevertheless, Kim *et al.* [10] noted that Tu's and Piramuthu's protocol is vulnerable to a simple active attack and proposed a new protocol called *Swiss-Knife*, attempting to correct the vulnerabilities of all its predecessors. All these protocols [7–10] use the same method to generate the responses that are later used in the RBE phase. Further distance-bounding protocols [11–13] have been proposed and many attacks against them have been published [14–17].

Our contribution: In this paper,[1] we perform a security analysis in a family including the above distance-bounding protocols [7–10]. The way these generate the responses during the RBE phase enables us to recover the shared secret after a number of random nonce repetitions logarithmic in the key length. The complexity of the attack depends mainly on the length of the random nonces used. We provide a detailed theoretical and experimental analysis on the number of sessions required to achieve full disclosure of the key. More precisely, we prove that, with high probability, the success of the attack mainly depends on the length of the nonces and the number of protocol executions (eavesdropped sessions) rather than the length of the key. In addition, we experimentally investigate the success rate of the attack under noisy conditions. The main utility of our results would be to practitioners. Our bounds can be used to select appropriate values for their security parameters when the aforementioned protocols are used.

We should note here that, ideally, nonces should never repeat. However, there exist protocols making use of too short nonces which do repeat. For instance, WEP uses 24-bit nonces [19, 20] which are valid for a single packet encryption. These nonces are typically encrypted (some weak ones are skipped though). So, repetition of nonces occur at least every 2^{24} packets.

Nevertheless, for typical RFID usage scenarios, there would not be a sufficient number of sessions for the attack to succeed with a significant probability. However, having a legitimate RFID tag at his disposal would allow an attacker to perform multiple protocol runs and obtain the secret key.

Our analysis shows that this event is a real threat while our countermeasure could mitigate the use of short nonces. Finally, we also describe a countermeasure that could be used in order to overcome this attack *even with short nonces*. In our analysis, we are mainly referring to RFID distance-bounding protocols and so the concepts Verifier/Reader and Tag/Prover are used interchangeably.

[1] Some partial and preliminary results from this paper were previously published in technical report arXiv:0906.4618 [18].

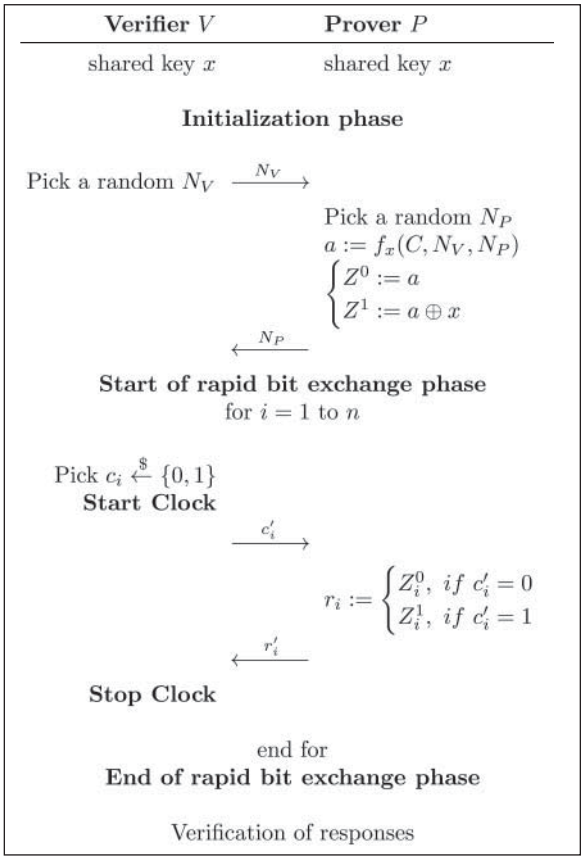

FIGURE 1. A general view of the distance-bounding protocols proposed in [7–10].

Organization: The paper is organized as follows. Section 2 provides a general view of the protocols which we investigate, and which all generate the responses used in the RBE phase in the same way. Section 3 describes the passive attack that can be launched against these protocols. Section 4 presents the theoretical analysis of the attack, while Section 5 describes the experimental results we have performed in order to find the required number of sessions to recover the key both under noisy conditions and if a noise-free channel is used in the RBE phase, when the attack is performed against the *Swiss-Knife* protocol. Finally, Section 6 describes the possible countermeasure that could be used to combat this attack, while Section 7 concludes the paper.

2. A GENERAL VIEW OF THE ANALYSED DISTANCE-BOUNDING PROTOCOLS

In this section, we describe a family of protocols [7–10] that use the same technique to generate the responses used in the RBE phase. Figure 1 depicts a general view of this family of

TABLE 1. Notation.

d	The size of nonce space (so $\log_2 d$ is the nonce bit length)
n	Number of rounds in the RBE phase
x	The secret key shared between the verifier and the prover
a	The session key
k	The bit length of the secret key x
V	The verifier
P	The prover
N_V	The random nonce generated by the verifier
N_P	The random nonce generated by the prover
f_x	A one-way, collision resistant, pseudorandom keyed function (using key x)
RBE	The RBE phase
c_i	The ith challenge sent by the verifier in RBE
c'_i	The ith challenge received by the prover in RBE
r_i	The ith response sent by the prover in RBE
r'_i	The ith response received by the verifier in RBE
$\xleftarrow{\$}$	Denotes sampling uniformly at random
Δt_i	The time difference in the ith round
T_{max}	Maximum allowed time difference
$\|m\|$	The bit length of variable m

protocols, while Table 1 depicts the notation used in the rest of the paper.

Assume that the prover P and the verifier V share a secret key x. Firstly, the *initialization phase*, which is not time critical, is executed. The verifier V chooses a random number N_V and transmits it to the prover P. On receiving it, P generates a random number N_P and computes a session key $a = f_x(C, N_V, N_P)$, where f denotes a pseudorandom function (PRF) and C represents any additional parameters such as the identifiers of the participants and is constant for all instances of the protocol used in an attack. In this family of protocols, the output of the PRF f should have the same length as x. We should note here that the computation of the session key a varies. More precisely, in the protocols [7–9] a depends on both random nonces N_V and N_P while in the *Swiss-Knife* [10] protocol it only depends on the random nonce N_P.

Then, P splits the secret key into two shares by computing the following:

$$Z^0 := a,$$
$$Z^1 := a \oplus x.$$

Finally, the prover sends N_P to the verifier.

After the *initialization phase*, the RBE phase starts, which involves the exchange of challenges–responses at maximum bit rate. This phase is repeated n times (rounds), with i varying from 1 to n. The number of rounds n is actually a security parameter. At each round i the challenge–response delay Δt_i is measured. V starts by choosing a random bit c_i, initializing the clock to zero and transmitting c_i to P. The values received by P are denoted by c'_i.[2] Next, P answers by sending $r_i := Z_i^{c'_i}$. The values received by V are denoted by r'_i. On receiving r'_i, V stops the clock and stores the received answer and the delay time Δt_i.

After the end of the *RBE* phase, a final *verification* phase may be required.

In the *Swiss-Knife* protocol, the noise during the distance-bounding phase is taken under consideration and the need for a threshold of allowed errors is indicated. A detailed analysis for the optimal selection of this threshold is provided in [21].

3. DESCRIPTION OF THE ATTACK

Ideally, nonces should never repeat. However, in practice there is always a non-zero probability of repetition, which becomes higher for shorter nonces. This family of protocols is particularly vulnerable to nonce repetitions, as will be made clear in the sequel.

In this section, we describe a passive attack that may be launched against any of the protocols [7–10], whose responses during the RBE phase are generated in the manner described in Section 2. The attack can be launched more easily against the *Swiss-Knife* [10] protocol because of the way the session key a is generated in this protocol (i.e. dependence only on the random nonce N_P). Nevertheless, it can also be generalized to the other three protocols [7–9]. To facilitate the exposition, in the following we use the variable N to denote the nonces that are used in the generation of the session key a. More precisely, it holds that

(1) $N = N_P$ for the case where the attack is launched against the *Swiss-Knife* [10] protocol;
(2) $N = N_P||N_V$ for the case where the attack is launched against any of the other three protocols [7–9].

A successful execution of the attack may lead to the full disclosure of the shared secret key x. We denote by b the number of bits that the attacker recovers during the attack out of the total n bits of the secret key x. The value of b is initialized to zero ($b = 0$). The attack is successful as soon as $b = n$.

To perform the attack, the attacker just needs to eavesdrop on the public insecure channel used for the communication between a legitimate prover P and a verifier V during both phases (i.e. the *initialization* and the RBE phase) and to follow these steps:

[2]The *RBE* is performed under noisy conditions. If we assume that the communication between two entities Y and Z is not noise free, then whenever a symbol $m \in M$ is sent from Y to Z, the symbol m' that Z receives may differ from m due to noise. This is modelled as a probability of erroneous transmission from Y to Z, $\omega_{YZ} = P(m' \neq m), \forall m, m' \in M$.

The attacker creates a hash table of vectors $(N, \{c_i, r_i\}_{i=1}^n)$ which are keyed by N. For each eavesdropped session:

(1) He captures:
   (a) the random nonce N sent either by P or by both P and V;
   (b) the challenge and response bits $\{c_i, r_i\}_{i=1}^n$ transmitted during the RBE phase.
(2) He inserts $(N, \{c_i, r_i\}_{i=1}^n)$ in the hash table unless a collision of N appears.
(3) When a value N appears again, the new values are not stored but the following action is performed. For $i = 1$ to n such that x_i is unknown so far:

$$\begin{cases} \text{If } c_i \neq c_i^*, \text{ then } x_i = r_i \oplus r_i^* \text{ and } b = b + 1, \\ \text{If } c_i = c_i^*, \text{ then } x_i \text{ remains undisclosed,} \end{cases}$$

where * denotes a new session with the same N but a different stream of challenges and responses.

We should note here that one can normally expect c_i and c_i^* to differ with probability $\frac{1}{2}$ so that on average half the remaining unknown bits of x will be revealed.

He *stops* once all the bits x_i have been disclosed, i.e. $b = n$.

4. THEORETICAL ANALYSIS OF THE ATTACK

4.1. Preliminaries

In this theoretical analysis, we only refer to collisions of nonces N; these nonces may either be $N = N_V$ sent by the verifier or $N = N_V \| N_P$ depending on the protocol that is being attacked (i.e. *Swiss-Knife* [10] or any of the other three protocols [7–9] correspondingly). Let $N_1, \ldots, N_t$, be a sequence of random nonces, with $N_t \in \mathcal{N}$ and $N_t \sim \mathit{Unif}(\mathcal{N})$, where $|\mathcal{N}| = d < \infty$, t is the total number of sessions, k is the length of the key and $\mathit{Unif}(\mathcal{N})$ denotes the uniform distribution over the set $\mathcal{N}$. We note that, in the family of protocols under investigation, the length of the key is identical to the number of rounds (i.e. $n = k$), since each round uses one bit of the key.

We define the auxiliary variable l_t to equal one when we see a nonce again:

$$l_t \triangleq \begin{cases} 1 & \text{if } \exists j < t : N_j = N_t, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

It is easy to see that l_t can be used to count the number of repetitions:

$$L_t \triangleq \sum_{j=1}^{t} l_j, \tag{2}$$

so that L_t equals the number of times elements of $\mathcal{N}$ are seen more than once. Thus, L_t denotes the number of repetitions in the first t sessions.

For the ℓth *repetition*, with $\ell \leq L_t$ we use c_ℓ for the challenge string in the first session involved in the collision and c_ℓ^* for the challenge in the last such session. We denote by $c_{\ell i}$ and $c_{\ell i}^*$ the ith[3] bit of the challenges sent during the first and last sessions involved in the collision, respectively.

We set $\Delta_{\ell i} = c_{\ell i} \oplus c_{\ell i}^*$. Since we only count repetitions of nonces, the following assumption can be made.

ASSUMPTION 1. *$\exists q > 0$ such that, for all $i \in \{1, \ldots, k\}$, $y \in \mathbb{N}$, $\ell \leq y$, it holds that*

$$\mathbb{P}(\Delta_{\ell i} = 0 \mid L_t = y) = q, \tag{3}$$

meaning that the events $\{\Delta_{\ell i} = 0\}$ are independent of i, ℓ, t and y.

4.2. High probability bound for the attack

The following theorem shows that, with high probability, the attacker will not discover the secret key after t sessions. The bound depends mainly on the number of nonces d, and only very weakly on the length of the secret key k. Thus, practitioners should be able to use this bound to select an appropriate length for the nonce.

THEOREM 4.1. *Given Assumption 1 for some $q \in [0, 1]$, when $d \in \mathbb{N}$ is the number of possible nonces, we can recover a key of length k, with probability at least $1 - \delta$, $\forall \delta \in [0, 1]$ after at most t sessions*:

$$t = O\left(\max\left\{\sqrt{d \log_{1/q}\left(\frac{2kq}{\delta(1-q)}\right)}, d^{2/3}(\ln(2/\delta))^{1/3}\right\}\right). \tag{4}$$

Proof. We define P_y to be the probability of finding the key after $L_t = y$ collisions. This is defined by

$$P_y \triangleq \mathbb{P}(\forall i, \exists \ell \leq L_t : \Delta_{\ell i} = 1 \mid L_t = y). \tag{5}$$

The probability of finding the ith bit of the key is

$$P_{y,i} = \mathbb{P}(\exists \ell \leq L_t : \Delta_{\ell i} = 1 \mid L_t = y). \tag{6}$$

Finally, the probability of not finding the ith bit of the key is given by

$$\begin{aligned} Q_{y,i} &= 1 - P_{y,i} = 1 - \mathbb{P}(\exists \ell \leq L_t : \Delta_{i,\ell} = 1 \mid y = L_t) \\ &= \mathbb{P}(\forall \ell \leq L_t, \Delta_{i\ell} = 0 \mid y = L_t) = q^y, \end{aligned} \tag{7}$$

for any i. Thus, it holds that $P_{y,i} = 1 - q^y$ and $P_y = (1 - q^y)^k$, where k is the length of the key.

Thus, the probability Q_y of not finding the key after y collisions is equivalent to the probability of not finding one bit

[3] In the *Swiss-Knife* protocol, it is obvious that the length of the challenge is the same as the length of the key, thus $i \in \{1, \ldots, k\}$.

of the key and, thus, equivalent to the probability of not finding bit i for some i. It holds

$$Q_y = \mathbb{P}\left(\bigcup_{i=1}^{k}\{\forall \ell \leq L_t, \Delta_{i\ell} = 0\} \mid y = L_t\right) = 1 - (1 - q^y)^k$$
$$\leq \sum_{i=1}^{k} \mathbb{P}(\forall \ell \leq L_t, \Delta_{i\ell} = 0 \mid y = L_t) = kq^y, \quad (8)$$

via the union bound. Thus,

$$P_y = 1 - Q_y \geq 1 - kq^y. \quad (9)$$

We use Z_t to denote the event $\{\forall i, \exists \ell \leq L_t : \Delta_{\ell i} = 1\}$ (of finding the key after at most t sessions) and Z_t' the complementary event $\{\exists i : \forall \ell \leq L_t : \Delta_{\ell i} = 0\}$. Then, for any $y \in \mathbb{N}$,

$$\mathbb{P}(Z_t) \geq \mathbb{P}[Z_t \wedge L_t > y] = \sum_{z>y} \mathbb{P}[Z_t \wedge L_t = z] \quad (10)$$
$$\geq \sum_{z>y} (1 - kq^z)\, \mathbb{P}[L_t = z] \quad (11)$$
$$= \mathbb{P}[L_t > y] - k \sum_{z>y} q^z\, \mathbb{P}[L_t = z] \quad (12)$$
$$\geq \mathbb{P}[L_t > y] - kq^{y+1} \frac{1}{1-q} \quad (13)$$
$$= 1 - \frac{kq}{1-q} q^y - \mathbb{P}[L_t \leq y]. \quad (14)$$

From *Lemma* A.1 *in Appendix* 1, we get

$$\mathbb{P}(L_t \leq y) \leq \exp(-2\alpha^2 t), \quad (15)$$

where we set y to

$$y = \mathbb{E}\, L_t - \alpha t, \quad (16)$$

and $\alpha > 0$.

Thus, the probability of successfully recovering the key after at most t sessions is bounded by

$$\mathbb{P}(Z_t) \geq 1 - \frac{kq}{1-q} q^y - \exp(-2\alpha^2 t). \quad (17)$$

But, according to [22, Theorem 5.15],

$$\mathbb{E}\, L_t = t - d + d\left(1 - \frac{1}{d}\right)^t. \quad (18)$$

From *Lemma* A.2 *in Appendix* 1, we get

$$\mathbb{E}\, L_t \geq \frac{t(t-1)}{2ed}, \quad (19)$$

where e is the base of the natural algorithm and $d \geq 2$. For $\alpha = \beta t / 2ed$ with $\beta < 1$, Equation (17) gives us

$$\mathbb{P}(Z_t) \geq 1 - \frac{kq}{1-q} \exp\left(\frac{((1-\beta)t^2 - t)\ln q}{2ed}\right) - \exp\left(-\frac{2\beta^2 t^3}{4e^2 d^2}\right).$$

By choosing $\beta = \frac{1}{2}$, we obtain

$$\mathbb{P}(Z_t) \geq 1 - \frac{kq}{1-q} \exp\left(\frac{(t^2 - 2t)\ln q}{4ed}\right) - \exp\left(-\frac{t^3}{8e^2 d^2}\right). \quad (20)$$

Now note that, in order to have

$$\frac{kq}{1-q} \exp\left(\frac{(t^2 - 2t)\ln q}{4ed}\right) \leq \delta_1, \quad \exp\left(-\frac{t^3}{8e^2 d^2}\right) \leq \delta_2, \quad (21)$$

for some $\delta_1, \delta_2 \in [0, 1]$, it is sufficient for t to be

$$t \geq 1 + \sqrt{4ed \log_{1/q}\left(\frac{kq}{\delta_1(1-q)}\right) + 1}, \quad (22)$$
$$t \geq (2ed)^{2/3} (2\ln(1/\delta_2))^{1/3}. \quad (23)$$

Now let us set $\delta_1 = \delta_2 = \delta/2$. By substituting $t = 1 + \sqrt{4ed \log_{1/q}(2kq/\delta(1-q)) + 1}$ in the first exponential term of Equation (20) and $t = (2ed)^{2/3}(2\ln(2/\delta))^{1/3}$ in the second exponential term, we obtain

$$\mathbb{P}(Z_t) \geq 1 - \delta. \quad (24)$$

□

Theorem 4.1. implies that the attack's success probability depends mostly on the length of the nonce. On the other hand, it only very weakly depends on the length of the key. Consequently, the protocols in this family are inherently weak if nonces are short. Since the protocols under consideration use single-bit challenges for each round, the worst case value for q is $\frac{1}{2}$.

5. EXPERIMENTAL RESULTS

In this section, we experimentally estimate the number of sessions that need to be eavesdropped for a successful full disclosure of the secret key in the *Swiss-Knife* [10] protocol. A similar approach could be followed for the other three protocols [7–9].

We start by considering the simple scenario in which there are no transmission errors in the channel. Then, we adopt a more realistic approach and consider that transmission errors can occur both in the backward (tag-to-reader) and in the forward channel (reader-to-tag). In both cases, we follow the approach described in the previous section. All simulations report averages over 2^{14} runs.

5.1. Ideal communication channel

In these experiments, we examined the case when there are no transmission errors in the channel. Initially, we investigated the

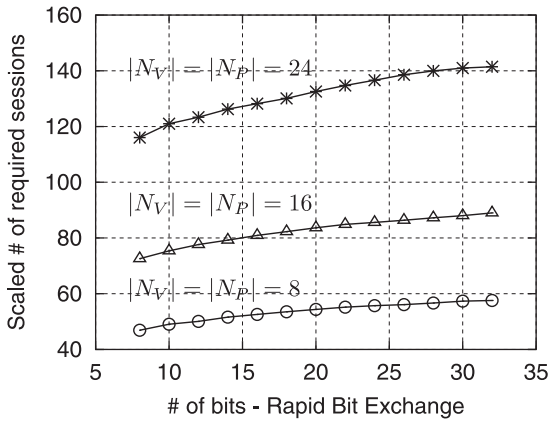

FIGURE 2. Expected number of required sessions for ideal communication channel (BER = 0.0) for various nonce lengths m as the number of rounds n and the key length $k = n$ increase. Note that the number of sessions is scaled down by 10 and 100 for $m = 16$ and $m = 24$, respectively.

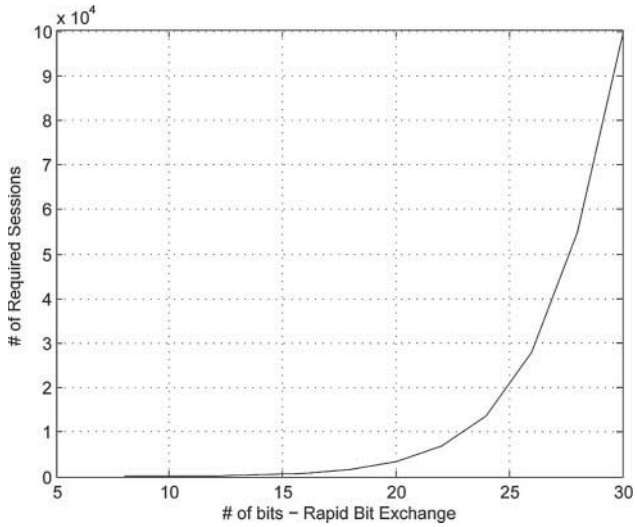

FIGURE 3. Expected number of required sessions for a successful attack in the ideal communication channel (BER = 0.0) as n increases with $k = n$, $d = 2^n$.

attacker's success probability when the number of nonces d was kept fixed and the key length k and number of rounds n, which are equal in this protocol family, increase. In particular, we varied the nonce length $m \in \{8, 16, 24\}$ and set $d = 2^n$ and $k = n$. The results are presented in Fig. 2. Note that the number of sessions is *scaled* by dividing by 1 ($m = 8$), 10 ($m = 16$) and 100 ($m = 24$). While the number of eavesdropped sessions increases when n, k rise, this effect is insignificant, as the theory predicted. The greatest effect is due to the length of the nonce.

For this reason, in the remaining experiments, we set all the variables in the protocol to the same bit length (i.e. $|N_P| = |N_V| = |x| = n$). In particular, Fig. 3 shows the number of sessions required to recover the session key for an ideal channel under an increasing bit length. It is obvious that the expected number of required sessions rises exponentially with the bit length.

5.2. Noisy communication channel

In this section, we consider that errors may appear in the communication due to noise. More precisely, we assume an independent, symmetric noise model for both forward and backward channels. In fact, we assume that both channels have error rate ω. We use the results of the second experiment from the previous section as follows. We vary n and set $d = 2^n$ and $k = n$ and, for each n, the attacker eavesdrops on a number of sessions equal to the expected number of sessions required to obtain the key in Fig. 3. Owing to the noise in the channel some of the disclosed bits may be incorrect. In our experiments, if one or more bits of the key are incorrect, we count it as an unsuccessful attack and only when the whole key is revealed, a success is scored. Figure 4 depicts the results of our simulations. We can observe that the probability of success is over 80% when

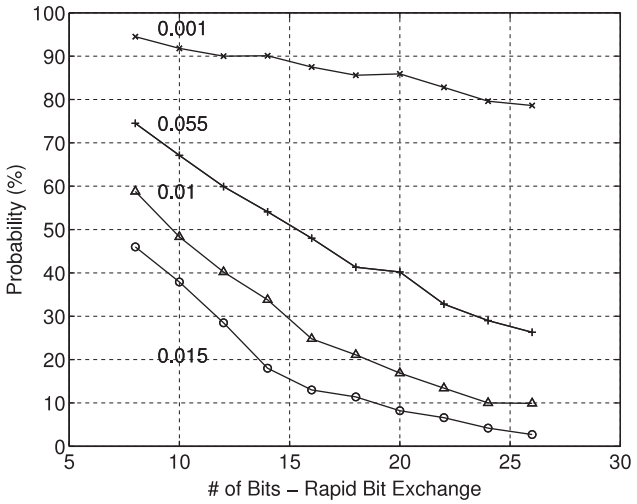

FIGURE 4. Adversary's success probability for noisy communication channel and four values of the BER as n increases, with $k = n, d = 2^n$.

the bit error rate (BER)[4] is 10^{-3}. On the contrary, when the BER is higher (BER = 0.015), the probability of success declines exponentially as we increase the number of bits transmitted during the RBE phase.

To improve upon the results of Fig. 4, the attacker has to eavesdrop on a number of sessions greater than the sessions required for the case where the communication channel is noise free. In our experiments, we consider the probability of partial

[4]We should note here that the BER rates that are usually experienced in RFID tags vary from 10^{-1} to 10^{-4} [23], while the BER rates that are usually experienced by an RFID reader vary from 10^{-1} to 10^{-6} [24].

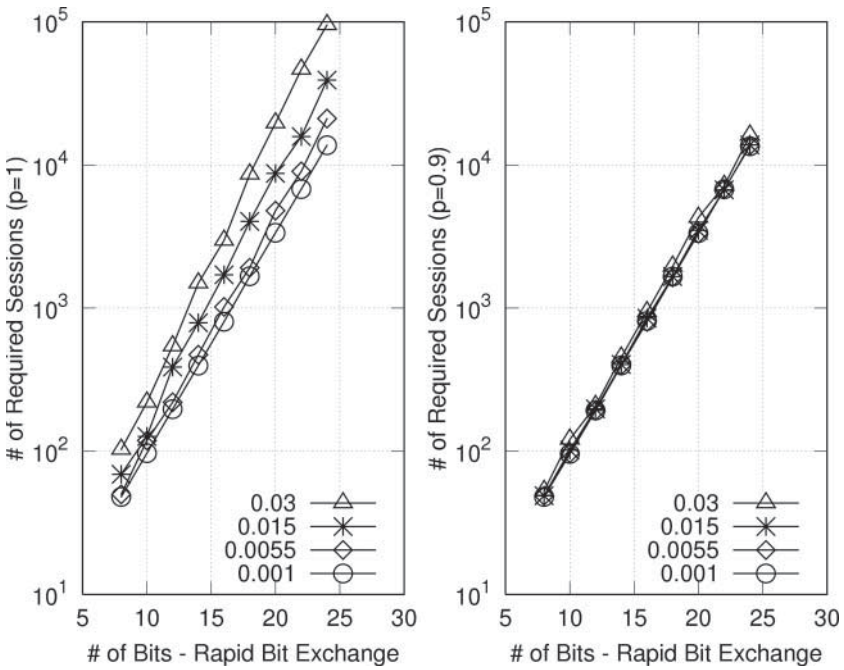

FIGURE 5. Expected number of eavesdropped sessions required for noisy communication channel with $k = n$, $d = 2^n$ as the BER varies. The left and right panels show the result for success probability $P = 1$ and $P = 0.9$, respectively.

success P, which denotes the percentage of bits of the total key length that is disclosed. Then, for a partial success P, an average of np bits are disclosed. For different values of P, we estimate the expected number of sessions required. We follow the algorithm described in Section 3. Nevertheless, we performed a small modification in this algorithm to reduce the required number of sessions. Let us assume that an invocation of the mentioned algorithm is represented as $k^{(j)} = \text{call}(S(j))$, where $k^{(j)}$ is the secret key obtained at the jth invocation. We follow the algorithm described below:

(1) Execute the algorithm $k^{(j)} = \text{call}(S(j))$.
(2) Derive the most common value of the keys obtained.
For $i = 1, \ldots, n$ do:

$$\begin{cases} \text{If } \sum_j (k_i^{(j)} = 1) \geq \sum_j (k_i^{(j)} = 0), \text{ then } y_i = 1, \\ \text{else } y_i = 0, \end{cases}$$

where $k_i^{(j)}$ and y_i represent the ith bit of $k^{(j)}$ and y, respectively.
(3) Check whether $P \cdot n$ bits of the key are already disclosed (i.e. $d(k, x) \leq P \cdot n$, where d denotes the Hamming distance). If not, jump to Step (1).

We perform an analysis of the number of sessions required under noisy conditions. As previously, we perform 2^{14} independent trials to obtain an average value. Figures 4–6 summarize the results obtained. Figure 5 depicts the expected number of sessions required for $P = 1$ and $P = 0.9$. By comparing these results, we observe that the expected number of sessions is increased by an order of magnitude when the percentage of recovered bits of the key is increased from 90 to 100%. For $P = 0.9$, the influence of errors in the channel is only slightly more noticeable when the BER is higher. However, for $P = 1$, the effect of the BER is marked. As expected, the required number of sessions increases when the number of challenge–response bits transmitted during the RBE phase rises and/or the noise (BER) in the channel increases.

Figure 6 depicts the expected number of eavesdropped required sessions for very noisy channels. Specifically, we show how the required number of eavesdropped sessions changes as the number of bits increases for different values of P ($P = \{0.6, 0.7, 0.8, 0.9\}$) and BER values (BER = 0.03 or BER = 0.06). For a moderately higher BER = 0.03, there is no significant difference between the required number of sessions for recovering a part of the key (i.e. 60–70%) or almost the whole key (i.e. 90%). On the other hand, this difference is evident for higher BERs (i.e. 0.06 in Fig. 6). From these figures, we conclude that when the BER is low, the additional effort required to obtain a larger percentage of the key is small. The effect of noise becomes clearer in Fig. 7, which shows the number of required eavesdropped sessions to fully obtain the key ($P = 1$), versus the number of bits in the RBE phase, for three different BER values (BER = $\{0.03, 0.06, 0.09\}$). As a rule of thumb, we observe that when the BER is twice as much as was previously simulated, the expected number of required sessions is multiplied at least by a factor of 3.

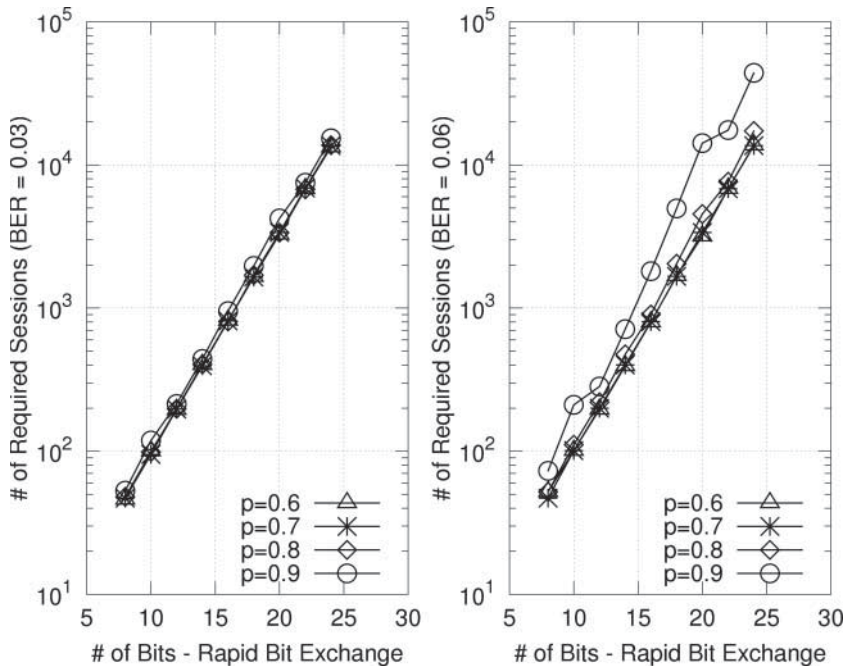

FIGURE 6. Expected number of required eavesdropped sessions for noisy communication channel as the required success probability P varies. The left and right panels show the result for BER $= 0.03$ and BER $= 0.06$, respectively, with $k = n$, $d = 2^n$.

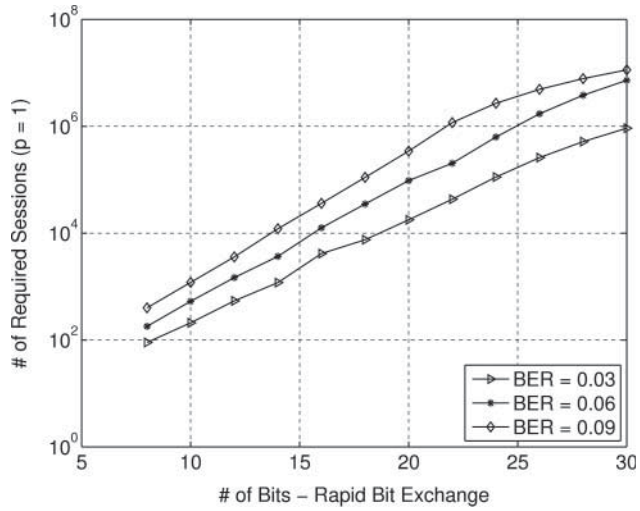

FIGURE 7. Expected number of required eavesdropped sessions for a noisy communication channel (BER $= \{0.03, 0.06, 0.09\}$) with $k = n$, $d = 2^n$.

Nevertheless, even for BER $= 0.03$, the expected number of required eavesdropped sessions is inferior to the number of attempts that an adversary would need to perform a brute force attack[5] (i.e. $\sharp\, of\ Required\ Sessions/2^n < 1$).

[5]If the number of bits transmitted during the RBE phase is n, then the attacker has to guess a word of n bits.

6. COUNTERMEASURE AGAINST THE ATTACK

A simple way to circumvent the attack would be to implement the Hancke and Kuhn protocol [3]. However, that protocol does not prevent *terrorist fraud* attacks. We urge the reader to consult [25] in which secret-sharing schemes and terrorist frauds are studied in detail. Instead, we propose a countermeasure which could be easily applied to all four distance-bounding protocols in this family [7–10]. Our countermeasure avoids terrorist fraud attacks, and leaks very little information to a passive attacker. The *main idea* is to force the attacker to use multiple collisions of the *same pair* of nonces to obtain new information. In other words, observing multiple, but *distinct*, pairs of nonces, does not give any more information than observing only one collision.

The countermeasure is deployed in the *initialization* phase, which is modified as follows:

(1) Again the verifier V chooses a random nonce N_V and transmits it to the prover P.
(2) The prover P generates the nonce N_P and a random number R and computes the temporary keys $\{a', b'\}$ as described below:

$$a' \| b' := f_x(N_V, N_P, C),$$

where C denotes any extra but constant parameters.

Then, the prover P splits its permanent secret key x into two shares by computing:

$$\begin{cases} Z^0 := a', \\ Z^1 := b', \end{cases}$$

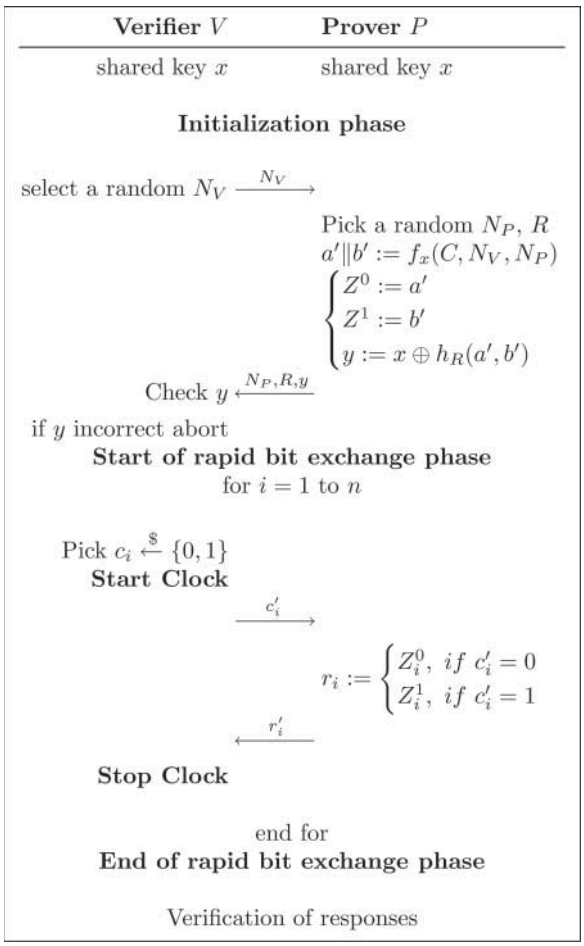

FIGURE 8. The general view of the distance-bounding protocols proposed in [7–10] modified with the proposed countermeasure secure against terrorist fraud attacks.

Additionally, the prover P calculates the value $y := x \oplus h_R(a', b')$, where h is a universal hash function. Finally, the prover P transmits the values of N_P, R and y to the verifier.

(3) The verifier receives N_P, R and y and checks the correctness of y. If y is not correct, the whole protocol is aborted.

In Fig. 8 is depicted the general view of the distance bounding-protocols [7–10] modified with the proposed countermeasure. According to the *Leftover Hash Lemma* (*Lemma* A.3 *in Appendix* 1), it holds that $(h_R(a', b'), R)$ is ϵ-indistinguishable from a uniformly distributed bit string if the following holds:

$$k \leq H_\infty(a', b') - 2\log_2 \frac{1}{\epsilon}, \tag{25}$$

where h denotes a universal hash function with range 2^k and $H_\infty(a', b')$ denotes the min entropy of a', b' (i.e. the expected number of bits in a',b' which remain unknown after running the attack).

The calculation of y and its transmission from P to V adds no extra security risk to the protocol. Additionally, the use of a', b', y and R prevents the type of attack presented in [26, Section 3.2] since the secret sharing is well designed.

In this case, an adversary A must know a', b', R and y in order to complete the protocol with a legitimate verifier V (potentially by colluding with a dishonest prover P in the initialization phase and then by himself in the RBE phase). However, this knowledge implies that A will also be able to calculate the shared secret key x, which contradicts the definition of the *terrorist fraud* attack.

In the family of protocols, we have investigated [7–10], a simple (disjoint) collision of the pair (N_P, N_V) (or just N_P in the case of the *Swiss-Knife*) reveals some bits of the key. In the proposed countermeasure, this is not the case. Simple disjoint collisions of pairs (N_P, N_V) reveal bits of different vectors a', b'. In order to recover all the bits of a', b' and thus the key x, a sufficient number of multi-collisions (i.e. the same pair (N_V, N_P) used in many eavesdropped successful runs of the protocol) is required; this happens much more rarely than simple disjoint collisions l_t, which are sufficient for the disclosure of the key in the original protocol family.

More precisely, an adversary A that eavesdrops on a successful execution of the protocol may be able to disclose half of the bits of $(a'\|b')$, since when $c_i = 0$, it holds $r_i = a_i'$, and when $c_i = 1$, it holds $r_i = b_i'$. If $2n$ is the total length of $(a'\|b')$, then after a protocol run, the min-entropy will be n. If a collision appears in the used nonces (N_P, N_V), then the attacker may be able to recover more bits (using an attack similar to the one described in Section 3) and the min-entropy will become $n/2$. From any two sessions where a collision appears the adversary observes two pairs of values:

$$R,\ y = x \oplus h_R(a', b') \quad \text{and} \quad R^*,\ y^* = x \oplus h_{R^*}(a', b').$$

Since $(a', b') \to (h_R(a', b'), h_{R^*}(a', b'))$ is also universal, for

$$2k \leq \frac{n}{2} - 2\log_2 \frac{1}{\epsilon}, \tag{26}$$

this is ϵ-indistinguishable from pure randomness (based on the *Leftover Hash Lemma* (*Lemma* A.3)). More generally, l_t simple collisions leak no more information. Full disclosure of the key is significantly harder using this modification, since multi-collisions are much rarer than the disjoint collisions exploited in our proposed attack.

In particular, according to the *Theorem* 1.2 *in Appendix* 1, if d denotes the possible pairs of nonces (N_V, N_P), then the probability P_r of having an r-collision (i.e. the same nonce pair (N_V, N_P) is used r times) after recording t sessions is bounded as follows:

$$P_r \leq \frac{1}{d^{r-1}} \binom{t}{r} \leq \left(\frac{t \cdot e}{r \cdot d}\right)^r \cdot d, \tag{27}$$

where the second inequality follows from a standard variant of Stirling's approximation of the binomial coefficient. Rewriting,

we obtain that $t \geq (P^{1/r}/e)rd^{1-1/r}$. Thus, the complexity of the attack is

$$t = \Omega(P^{1/r} r d^{1-1/r}). \tag{28}$$

For $r = \Omega(\log n)$, Equation (28) gives us $t \approx \Omega(d)$, while our attack is successful against the original protocol after $t \approx O(\sqrt{d})$ based on *Theorem* 4.1 *in Appendix* 1.

While the number of rounds n can be different from the length of the key k, they are nevertheless not completely independent. More precisely, n should satisfy the condition

$$n \geq 4k + 4\log_2 \frac{1}{\epsilon},$$

due to Equation (26), when the key x has k bits. On the other hand, we have security beyond the birthday barrier related to the size of the nonce.

Note that a PRF assumption alone is not enough to guarantee the security for this protocol, as shown in [27]. We may need some extra techniques to prove security.

7. CONCLUSION

We have presented an attack on a family of distance-bounding protocols, which relies on nonce repetition. We have provided a high probability bound on the success of this attack, showing that it depends mainly on the nonce length $\log_2(d)$ and only sub-logarithmically on the shared secret length k. In addition, we have experimentally investigated the success probability under noise-free and noisy conditions. This shows that the attacker's chances decrease as noise increases. Consequently, our theoretical bound can be used by practitioners to appropriately select their security parameters (i.e. the lengths of the employed nonces and the secret key) depending on the application scenario. In fact, k must be sufficiently high to make brute-force attack impractical and d must be large enough for our attack to have appropriately low success probability. Finally, we have introduced a countermeasure, based upon a modification of the initialization phase. This significantly improves security, since the number of sessions required for a successful attack with high probability is bounded below by d, whereas only $\sqrt{d}$ sessions were sufficient for the original protocol.

FUNDING

This work was partially supported by the Marie Curie IEF project 'PPIDR: Privacy-Preserving Intrusion Detection and Response in Wireless Communications', grant number: 252323 and the Marie Curie IEF Project 'ESDEMUU: Efficient sequential decision making under uncertainty', grant number: 237816.

REFERENCES

[1] Desmedt, Y. (1988) Major Security Problems with the 'Unforgeable' (Feige)-Fiat-Shamir Proofs of Identity and How to Overcome Them. *Proc. 6th Worldwide Congress on Computer and Communications Security and Protection—SecuriCom '88*, Paris, France, March 15–17, pp. 147–159. SEDEP.

[2] Brands, S. and Chaum, D. (1994) Distance-Bounding Protocols. *Proc. Advances in Cryptology—EUROCRYPT '93*, Lofthus, Norway, May, Lecture Notes in Computer Science 765, pp. 344–359. Springer, Berlin.

[3] Hancke, G. and Kuhn, M. (2005) An RFID Distance Bounding Protocol. *Proc. SECURECOMM'05*, Athens, Greece, September 5–9, pp. 67–73. IEEE Computer Society, Washington DC, USA.

[4] Munilla, J. and Peinado, A. (2008) Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels. *Wirel. Commun. Mob. Comput.*, **8**, 1227–1232.

[5] Singelée, D. and Preneel, B. (2007) Distance Bounding in Noisy Environments. *Proc. Security and Privacy in Ad-hoc and Sensor Networks: 4th European Workshop—ESAS'07*, Cambridge, UK, July, Lecture Notes in Computer Science 4572, pp. 101–115. Springer, Berlin.

[6] Munilla, J. and Peinado, A. (2010) Attacks on a distance dounding protocol. *Comput. Commun.*, **33**, 884–889.

[7] Tu, Y.-J. and Piramuthu, S. (2007) RFID Distance Bounding Protocols. *Proc. 1st Int. EURASIP Workshop on RFID Technology*, Vienna, Austria, September, pp. 67–68. EURASIP, Greece.

[8] Reid, J., Gonzalez Nieto, J.M., Tang, T. and Senadji, B. (2007) Detecting Relay Attacks with Timing-based Protocols. *ASIACCS '07: Proc. 2nd ACM Symp. Information, Computer and Communications Security*, Singapore, March, pp. 204–213. ACM.

[9] Kapoor, G., Zhou, W. and Piramuthu, S. (2008) Distance Bounding Protocol for Multiple RFID Tag Authentication. In Xu, C.-Z. and Guo, M. (eds), *Proc. 2008 IEEE/IFIP Int. Conf. Embedded and Ubiquitous Computing—Volume 02—EUC'08*, Shanghai, China, December, pp. 115–120. IEEE, Washington DC, USA.

[10] Kim, C.H., Avoine, G., Koeune, F., Standaert, F.-X. and Pereira, O. (2008) The Swiss-Knife RFID Distance Bounding Protocol. *Int. Conf. Information Security and Cryptology—ICISC*, Seoul, Korea, December, Lecture Notes in Computer Science 5461, pp. 98–115. Springer, Berlin.

[11] Bussard, L. and Roudier, Y. (2004) Embedding Distance-Bounding Protocols within Intuitive Interactions. *Security in Pervasive Computing: 1st Int. Conf.*, Boppard, Germany, Lecture Notes in Computer Science 2802, pp. 143–156. Springer.

[12] Bussard, L. and Bagga, W. (2004) Distance-Bounding Proof of Knowledge Protocols to Avoid Terrorist Fraud Attacks. Technical Report RR-04-109, Institute EURECOM.

[13] Rasmussen, K. and Čapkun, S. (2008) Location Privacy of Distance Bounding. *Proc. Annual Conf. Computer and Communications Security (CCS)*, Alexandria, VA, USA, pp. 149–160. ACM.

[14] Mitrokotsa, A., Dimitrakakis, C., Peris-Lopez, P. and Castro, J.C.H. (2010) Reid et al.'s distance bounding protocol and mafia fraud attacks over noisy channels. *IEEE Commun. Lett.*, **14**, 121–123.

[15] Aumasson, J.-P., Mitrokotsa, A. and Peris-Lopez, P. (2011) A Note on a Privacy-Preserving Distance-Bounding Protocol. *Proc. 13th Int. Conf. Information and Communications Security (ICICS 2011)*, Beijing, China, November, Lecture Notes in Computer Science, pp. 78–92. Springer-Verlag, Berlin, Heidelberg.

[16] Bay, A., Boureanu, I., Mitrokotsa, A., Spulber, I. and Vaudenay, S. (2012) The Bussard–Bagga and Other Distance Bounding Protocols under Man-in-the-Middle Attacks. *Proc. Inscrypt'2012, 8th China Int. Conf. Information Security and Cryptology*, Beijing, China, Lecture Notes in Computer Science. Springer.

[17] Mitrokotsa, A., Onete, C. and Vaudenay, S. (2012) Mafia Fraud Attack against the RC Distance-Bounding Protocol. *Proc. 2012 IEEE RFID Technology and Applications (IEEE RFID T-A)*, Nice, France, November, pp. 74–79. IEEE Press.

[18] Peris-Lopez, P., Hernandez-Castro, J.-C., Dimitrakakis, C., Mitrokotsa, A. and M.E. Tapiador, J. (2010). Shedding light on RFID distance bounding protocols and terrorist fraud attacks. Preprint, 2009, arXiv:0906.4618.

[19] Cam-Winget, N., Housley, R., Wagner, D. and Walker, J. (2003) Security flaws in 802.11 data link protocols. *Commun. ACM*, **46**, 35–39.

[20] Vaudenay, S. and Vuagnoux, M. (2007) Passive-Only Key Recovery Attacks on RC4. *Proc. Selected Areas in Cryptography, 14th Int. Workshop, SAC 2007, Ottawa, Canada*, August 16–17, Revised Selected Papers, Lecture Notes in Computer Science 4876, pp. 344–359. Springer.

[21] Dimitrakakis, C., Mitrokotsa, A. and Vaudenay, S. (2012) Expected Loss Bounds for Authentication in Constrained Channels. *Proc. INFOCOM 2012*, Orlando, FL, USA, March, pp. 478–85. IEEE press.

[22] Bogart, K., Stein, C. and Drysdale, S. (2004) *Discrete Math for Computer Science Students*. Key College Publishing, CA, USA.

[23] Thomas, S. and Reynolds, M. (2010) QAM Backscatter for Passive UHF RFID tags. *Proc. 2010 IEEE Int. Conf. RFID*, April, Orlando, Florida, USA, pp. 210–214. IEEE Press.

[24] Lazaro, A., Girbau, D. and Villarino, R. (2009) Effects of interferences in UHF RFID systems. *Prog. Electromagn. Res.*, **98**, 445–443.

[25] Avoine, G., Lauradoux, C. and Martin, B. (2011) How secret-sharing can defeat terrorist fraud. *Proc. 4th ACM Conf. Wireless Network Security—iSec'11*, Hamburg, Germany, June, pp. 145–156. ACM Press, New York, NY, USA.

[26] Abyaneh, S. and Reza, M. (2011) Security Analysis of Two Distance-Bounding Protocols. *Proc. Workshop on RFID Security—RFIDSec'11*, Amherst, MA, USA, June, Lecture Notes in Computer Science 7055, pp. 94–107. Springer, Berlin.

[27] Boureanu, I., Mitrokotsa, A. and Vaudenay, S. (2012) On the Pseudorandom Function Assumption in (Secure) Distance-Bounding Protocols—PRF-ness Alone Does Not Stop the Frauds! *Progress in Cryptology—LATINCRYPT 2012—2nd Int. Conf. Cryptology and Information Security in Latin America*, October, Lecture Notes in Computer Science 7533, Santiago, Chile, pp. 100–120. Springer.

[28] Azuma, K. (1967) Weighted sums of certain dependent random variables. *Tôhoku Math. J.*, **13**, 357–367.

[29] Impagliazzo, R., Levin, L. and Luby, M. (1989) Pseudo-Random Generation from One-Way Functions. *Proc. 21st Annual ACM Symp. Theory of Computing (STOC'89)*, Seattle, WA, USA, May, pp. 12–24. ACM Press, New York, NY, USA.

[30] Suzuki, K., Tonien, D., Kurosawa, K. and Toyota, K. (2008) Birthday paradox for multi-collisions. *IEICE Trans. Fund. Electron. Commun. Comput. Sci.*, **E91-A**, 39–45.

APPENDIX 1

THEOREM A.1 (Azuma–Hoeffding inequality [28]). *Let V_j be a martingale difference sequence (with range g_j), with respect to some sequence X_j. If $S_t = \sum_{j=0}^{n} V_j$, then, for any $u > 0$,*

$$\mathbb{P}(S_t \geq u) \leq \exp\left(-\frac{2u^2}{\sum_{j=1}^{t} g_j^2}\right), \tag{A.1}$$

$$\mathbb{P}(S_t \leq -u) \leq \exp\left(-\frac{2u^2}{\sum_{j=1}^{t} g_j^2}\right). \tag{A.2}$$

LEMMA A.1. *If L_t denotes the number of disjoint collisions for a sequence $x^t \triangleq x_1, \ldots x_t$ with $x_t \in X$ and $x_t \sim Unif(X)$, where $|X| = d < \infty$, then*

$$\mathbb{P}(L_t \leq \mathbb{E}\, L_t - \alpha t) \leq \exp(-2\alpha^2 t), \tag{A.3}$$

where $\alpha > 0$ and t denotes the number of sessions we have recorded.

Proof. Let

$$P_t \triangleq \mathbb{P}(L_{t+1} = L_t + 1 \mid x^t) \quad \text{where } x^t \triangleq x_1, \ldots, x_t. \tag{A.4}$$

Then

$$\mathbb{E}(L_{t+1} \mid x^t) = L_t + P_t, \tag{A.5}$$

and since L_t, P_t depend only on x^t, by defining

$$V_{t+1} \triangleq L_{t+1} - (L_t + P_t), \tag{A.6}$$

we obtain that V_t is a martingale difference sequence with respect to x^t since

$$\mathbb{E}(V_{t+1} \mid x^t) = 0. \tag{A.7}$$

It holds that

$$S_t = \sum_{j=1}^{t} V_j = L_t - \sum_{j=1}^{t-1} P_j = L_t - \mathbb{E}\, L_t, \tag{A.8}$$

which follows since P_t is the probability that we have a new collision after the session t.

Consequently, the rightmost sum is equal to the expected number of collisions until the session t. By applying *Theorem* A.1 (Azuma–Hoeffding inequality), we have

$$\mathbb{P}(S_t \leq -u) = \mathbb{P}(L_t \leq \mathbb{E}\, L_t - u) \leq \exp\left(-\frac{2u^2}{t}\right), \tag{A.9}$$

where $u > 0$.

In our case $|V_j| \leq 1$, so $g_j = 1$ for all j. If we set $u = \alpha t$ where $\alpha > 0$, then we get

$$\mathbb{P}(S_t \leq -u) = \mathbb{P}(L_t \leq \mathbb{E}\, L_t - \alpha t) \leq \exp(-2\alpha^2 t), \quad \text{(A.10)}$$

where $\alpha > 0$. □

LEMMA A.2. *For any $d \geq 2$, it holds that*

$$t - d + d\left(1 - \frac{1}{d}\right)^t \geq \frac{t(t-1)}{2ed}, \quad \text{(A.11)}$$

where e is the base of the natural logarithm.

Proof. Let us set

$$f(t) = t - d + d\left(1 - \frac{1}{d}\right)^t \quad \text{(A.12)}$$

and

$$g(t) = \frac{t(t-1)}{cd}. \quad \text{(A.13)}$$

We want to find c such that $f(t) \geq g(t)$.

The first partial derivatives of f and g with respect to t are correspondingly

$$f'(t) = 1 + d(1 - 1/d)^t \ln(1 - 1/d), \quad g'(t) = \frac{2t-1}{cd}. \quad \text{(A.14)}$$

In order to have $f(t) \geq g(t)$, it is sufficient to have $f'(t) \geq g'(t)$ and $f(1) \geq g(1)$ $\forall t \in [1, d]$.

For $t = 1$, we have $f(1) = g(1) = 0$.

In order to have $f'(t) \geq g'(t)$, it is sufficient to have $f''(t) \geq g''(t)$ and $f'(1) \geq g'(1)$ $\forall t \in [1, d]$.

The second partial derivatives of f and g with respect to t are correspondingly

$$f''(t) = d(1 - 1/d)^t [\ln(1 - 1/d)]^2, \quad g''(t) = \frac{2}{cd}. \quad \text{(A.15)}$$

In order to have $f''(t) \geq g''(t)$, we should have

$$c \geq \frac{2}{d^2(1 - 1/d)^t \ln^2(1 - 1/d)} \triangleq \sigma(t). \quad \text{(A.16)}$$

It is easy to see that $\sigma(t)$ is monotonic and increasing since $\sigma'(t) > 0$, $\forall t \in [1, d]$. Thus, we should have $c \geq \max_{t \in [1,d]} \sigma(t) = P(d)$, where

$$P(d) = \frac{2}{d^2(1 - 1/d)^d \ln^2(1 - 1/d)}. \quad \text{(A.17)}$$

It is easy to show that $P(d)$ is also monotonic and increasing and that $P'(d) > 0$. We have

$$c \geq \sup_{d \geq 1} P(d) = \lim_{d \to \infty} \frac{2}{d^2(1 - 1/d)^d \ln^2(1 - 1/d)} = 2e. \quad \text{(A.18)}$$

Thus, it holds that $f''(t) \geq g''(t)$ whenever $c \geq 2e$.

In order to have $f'(1) \geq g'(1)$, we should have

$$c \geq \frac{1}{d + (d^2 - d)\ln(1 - 1/d)}. \quad \text{(A.19)}$$

If we set

$$h(d) = \frac{1}{d + (d^2 - d)\ln(1 - 1/d)}, \quad \text{(A.20)}$$

it is easy to show that $h(d)$ is monotonic and increasing for $d \geq 2$ and that $h'(d) > 0$. Thus, we conclude that $f(t) > g(t)$ when $c \geq 2e$ and $d \geq 2$. □

LEMMA A.3 (Impagliazzo–Levin–Luby [29]). *If $m \leq H_\infty(X) - 2\log_2(1/\epsilon)$ and h is a universal hash function with a range of size 2^m, then $(h_N(X), N)$ and (U, N) have distributions which are ϵ-indistinguishable, where X, N, U are independent and N and U are uniformly distributed.*

THEOREM 1.2 [30]. *If d denotes the number of possible nonces and t the number of recorded sessions, then the probability to have an r-collision is bounded by*

$$\mathop{\mathbb{P}}_{r} \leq \frac{1}{d^{r-1}} \binom{t}{r}. \quad \text{(21)}$$