A Note on a Privacy-Preserving Distance-Bounding Protocol

Jean-Philippe Aumasson[1], Aikaterini Mitrokotsa[2],
and Pedro Peris-Lopez[3]

[1] Nagravision, Switzerland
`jeanphilippe.aumasson@gmail.com`
[2] EPFL, Switzerland
`katerina.mitrokotsa@epfl.ch`
[3] TU Delft, Netherlands
`p.perislopez@tudelft.nl`

Abstract. Distance bounding protocols enable a device to establish an upper bound on the physical distance to a communication partner so as to prevent location spoofing, as exploited by relay attacks. Recently, Rasmussen and Čapkun (ACM-CCS'08) observed that these protocols leak information on the location of the parties to external observers, which is undesirable in a number of applications—for example if the leaked information leads to the identification of the parties among a group of devices. To remedy this problem, these authors proposed a "privacy-preserving" distance bounding protocol, i.e. that leaks no information on the location of the parties. The present paper reports results from an in-depth security analysis of that new protocol, with as main result an attack that recovers the ephemeral secrets as well as the location information of the two parties for particular choices of parameters. Overall, our results do not contradict the preliminary security analysis by the designers, but rather extends it to other parts of the attack surface.

Keywords: wireless communication, distance bounding, privacy.

1 Introduction

Wireless communications often have security goals that include not only traditional cryptographic notions—confidentiality, integrity, authenticity, availability—but also guarantees about the geographical location of a communicating device. For example, secured location information is necessary in battlefield ad hoc networks [1], in access control systems [2], and even in certain satellite DTV conditional access systems [3]. A particular case is when a "verifier" device has to ensure that a "prover" legitimate device is in the vicinity of the former; a trivial countermeasure is to ensure that the signal is low enough to prevent reception from farther locations. But this can be easily bypassed using a "proxy" device that amplifies and forwards the signal to a remote attacker—such

attacks are known as *relay attacks*[1]. A countermeasure against relay attacks is the use of *distance bounding protocols.*

As their name suggests, distance bounding protocols enable a *verifier* device (denoted V) to establish an upper bound on the physical distance to an untrusted *prover* device (P). These protocols were first introduced in 1994 by Brands and Chaum [5] to thwart distance fraud and the so-called mafia fraud attacks (two types of relay attacks) as applied against ATM's; the target application of [5] was thus to check the proximity between a legitimate ATM and a user's smartcard. Since this seminal work, a broad range of distance bounding protocols have been proposed for technologies as RFID [6, 7, 8, 9], ultra-wideband (UWB) devices [10, 11, 12], wireless ad hoc networks [13, 14, 15], or sensor networks [16]. Distance bounding can be extended to *location verification* (also called *secure positioning*) if multiple verifiers interact with the prover: the location returned is then in the intersection of the bounding spheres surrounding each verifier. Dedicated location verification protocols were proposed in [17, 14]. As an alternative to radio waves Sastry, Shankar, and Wagner [2] proposed to use ultrasound waves to obtain better spatial resolution, due to sound's much lower speed. However, this technology is susceptible to wormhole attacks [18] (e.g. if the signal is converted to a faster radio signal to deceive the verifier).

In 2008, Rasmussen and Čapkun noted [19] that known distance bounding protocols leak information to external observers on the distance and location of the prover and the verifier. In short, information leaks through the measure of messages' arrival times, which allows one to draw on the neighborhood map hyperboles representing physical bounds on the devices' location. Although locations and relatives distances are obvious in some cases—particularly with short-range signals, or with only two devices in the neighborhood—attackers are often not supposed to know them. For example, in a scenario with a large number of devices and vehicles (as a battlefield), it can be necessary to hide the identity of the communicating parties, who in turn need to ensure secure positioning; if an adversary can determine the relative position of the communicating parties, it then becomes much easier to identify them. Rasmussen and Čapkun thus investigated potential countermeasures against leakage of information and proposed a *privacy-preserving* distance bounding protocol [19].

Our contribution: The present work focuses on the Rasmussen-Čapkun (henceforth RČ) privacy-preserving distance bounding protocol, as specified in [19, 5.4]. We report results from our in-depth security analysis of the RČ protocol's, with as main result an attack against its privacy-preserving property. We argue that the properties of challenge numbers (called "nonces" in [19]) are of utmost importance for the security of the protocol, presenting attacks exploiting partial unpredictability as well as non-uniqueness. We provide a more complete specification of the protocol than in [19] in order to facilitate proof-of-concept implementations. Our results do not contradict the preliminary security analysis in [19], but rather extends it to other parts of the attack surface.

---

[1] An academic proof-of-concept relay attack against cars' remote locking has recently been realized by the team of Čapkun, as reported in [4].

2   Background

2.1   Distance Bounding Protocols

A distance bounding protocol enables a verifier V to calculate an upper-bound on his physical distance to a prover P. Distance bounding protocols are usually composed of two phases: the *initialization phase* and the *distance bounding phase* (see Fig. 1). The initialization phase is not time critical and usually involves P's commitment to some random number. The distance bounding phase is time critical and usually involves a *rapid bit exchange* (RBE), i.e. a sequence of challenge-responses performed at maximum bit rate. The number j of rounds of the RBE phase is a security parameter. The verifier selects j unpredictable challenges $c_1, \ldots, c_j$ and sends each of them to P. The prover generates the responses $r_1, \ldots, r_j$ using a function f that opens the commitment bit by bit, and that is parameterized by j and by the secret key; V then measures the response times $\Delta t_1, \ldots, \Delta t_j$ for each challenge.

By the end of the RBE, V verifies the correctness of the responses r_i, where $i = 1, 2, ..., j$ and calculates an upper bound on the distance of P based on the response time Δt_i of each challenge c_i, where $i = 1, 2, ..., j$. As information cannot travel faster than light, the distance between P and V is upper bounded by $c\Delta t_{\max}/2$, where $\Delta t_{\max}$ is the maximum delay time between sending out the bit c_i and receiving the bit r_i back.

Security thus relies on the unpredictability of the challenges and on the speed-of-light bound on the propagation of physical signals. For example with devices operating at a 1 Gbps bit rate (thus using 1 GHz carrier waves) one bit can be sent every nanosecond, resulting in a spatial resolution of approximately 30 cm; at 10 Mbps, the resolution is 30 meters, which is acceptable for large-scale use cases.

2.2   Information Leakage and Countermeasures

In previous distance bounding protocols, an attacker can compute the distance between P and V just by listening on the channel, as explained in [19, §3]. This can cause serious implications in applications where location and distance information is critical (as location-based access control). As noted earlier, relative distance information can assist a static observer in identifying the devices that are communicating among a larger population. To obtain information on the distance and or location of parties, the adversary only needs to measure the arrival time to his radio interface of two or three consecutive messages exchanged during the RBE [19, §3].

As a general countermeasure to preclude the distance and location leakage in distance bounding protocols, one should prevent the adversary from computing the "time-of-flight" of the signal between the verifier and the prover. Adding a random delay between messages transmitted during the RBE would reduce the adversary's knowledge but P could fool V into thinking that he waits longer than he actually does—making himself appear closer. Alternatively, V could
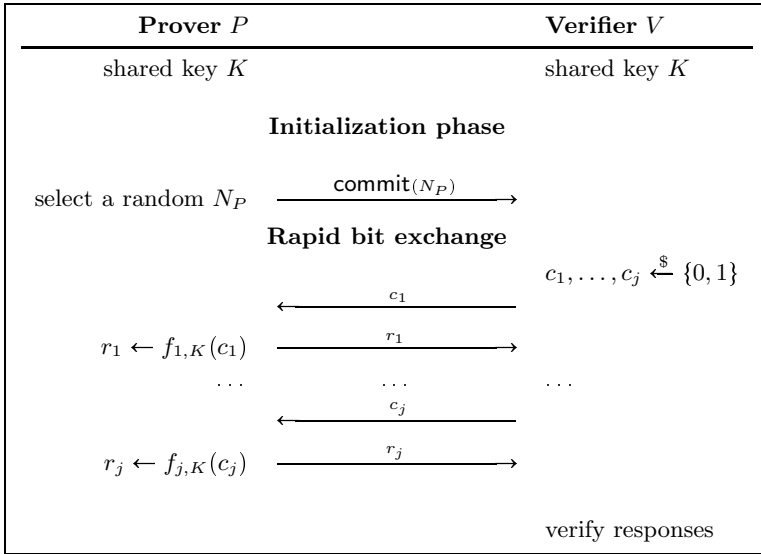
Fig. 1. A general view of distance bounding protocols ($\overset{\$}{\leftarrow}$ denotes sampling uniformly at random)

send multiple challenges to P before P responds. Nevertheless, the leakage of information is only alleviated if the adversary is unable to distinguish between the transmission originated from P and a transmission originated from V. As another countermeasure, P could send challenges at a fixed bit rate. By this mechanism, the adversary is not able to distinguish between receiving the responses from P and receiving the next challenge from V. Although this is an effective technique, in many scenarios the adversary can still compute the distance between V and P and his relative position to V or P. We refer to [19, §4] for a more detailed analysis of possible countermeasures.

Motivated by the fact that none of the above mentioned techniques provides effective protection against distance and location leakage, Rasmussen and Čapkun [19] proposed a new distance bounding protocol. The aim of the proposed protocol is to maintain the properties of existing distance bounding protocols while stoping the leakage of confidential information about distance and location.

3   The Rasmussen-Čapkun Protocol

The Rasmussen-Čapkun (RČ) protocol involves a prover P and a verifier V that communicate over an insecure channel. When the protocol succeeds V is able to calculate an upper bound on the physical distance to P, as in any distance bounding protocol. Fig. 2 gives a detailed description of the protocol, adding details that are only implicit in the original specification [19] (such as the authentication key K_2 used for computing and verifying the signatures).

Privacy preservation is ensured by hiding the actual RBE within a longer uninterrupted stream of bits.

3.1   Notations

The description of the RČ protocol in Fig. 2 assumes that P and V share the knowledge of:

- A k-bit encryption key K_1.
- A k-bit authentication key K_2.
- A symmetric encryption scheme (Enc, Dec).
- A symmetric authentication scheme (Sign, Verif) i.e. a MAC.
- A pseudorandom generator connected to a source of physical entropy.
- A timestamp counter of precision at least $1/r$ seconds, where r is the bit rate of the communication channel (for instance, with a 1 Gbps bit rate the precision should be up to a nanosecond).

In addition the RČ protocol is parameterized by the bit length n of N_P and N_V, the bit length m of M, and the (approximate) bit length of the bit streams of the distance bounding phase.

3.2   Detailed Description of the Protocol

The RČ protocol is composed of two main phases:

- **Initialization phase:** In this phase P selects a random N_P, which he encrypts-and-signs, before sending the resulting values to V. V decrypts and verifies the signatures and selects a random N_V and a random hidden marker M, which will be used in the distance bounding phase. V then encrypts-and-signs $M\|N_P$ and sends the results to P.
- **Distance bounding phase:** This phase involves the continuous exchange of bit streams between P and V. It can be separated into the following substeps:
    a. Initially V sends to P random data $\mathsf{Rand}_V()$ (i.e. generated with its pseudorandom generator); P receives this random data, XORs it with his own random generated data $(\mathsf{Rand}_V() \oplus \mathsf{Rand}_P())$ and sends them back[2] to V.
    b. At some randomly selected[3] point (within a setup window) V stops transmitting random data and starts transmitting the marker M. P keeps XORing the data received from V (at this time the M) with his own random data $(M \oplus \mathsf{Rand}_P())$ and sends them back to V.

---

[2] It would be equivalent, both functionally and in terms of security, to send back only $\mathsf{Rand}_P()$. The purpose of the XOR is rather to ensure a constant-time processing of the challenges, since XOR must be used later in a critical step of the distance bounding phase.

[3] The distribution considered is not specified in [19].

   c. As soon as the hidden marker M has been transmitted V starts transmitting N_V and starts storing the response that he will receive. P continuously monitors the data received to detect M, and as soon as it detects M he XORs the subsequent bits received with N_P followed by random bits.

   d. By the time V has sent the last bit of N_V, he starts again transmitting random data. Similarly, P starts XORing the received random data with his own random data ($\mathsf{Rand}_V()\oplus\mathsf{Rand}_P()$). After some short random delay V stops transmitting data and similarly after another random delay P ceases his transmission, to avoid leaking information on the synchronization of P and V.

By the end of the distance bounding phase, V counts the number of bits received between the time he transmitted the first bit of N_V and the time he received the first bit of $N_V \oplus N_P$. Given the bit rate and the processing delay, V can calculate the round trip time and thus an upper bound on the distance to P.

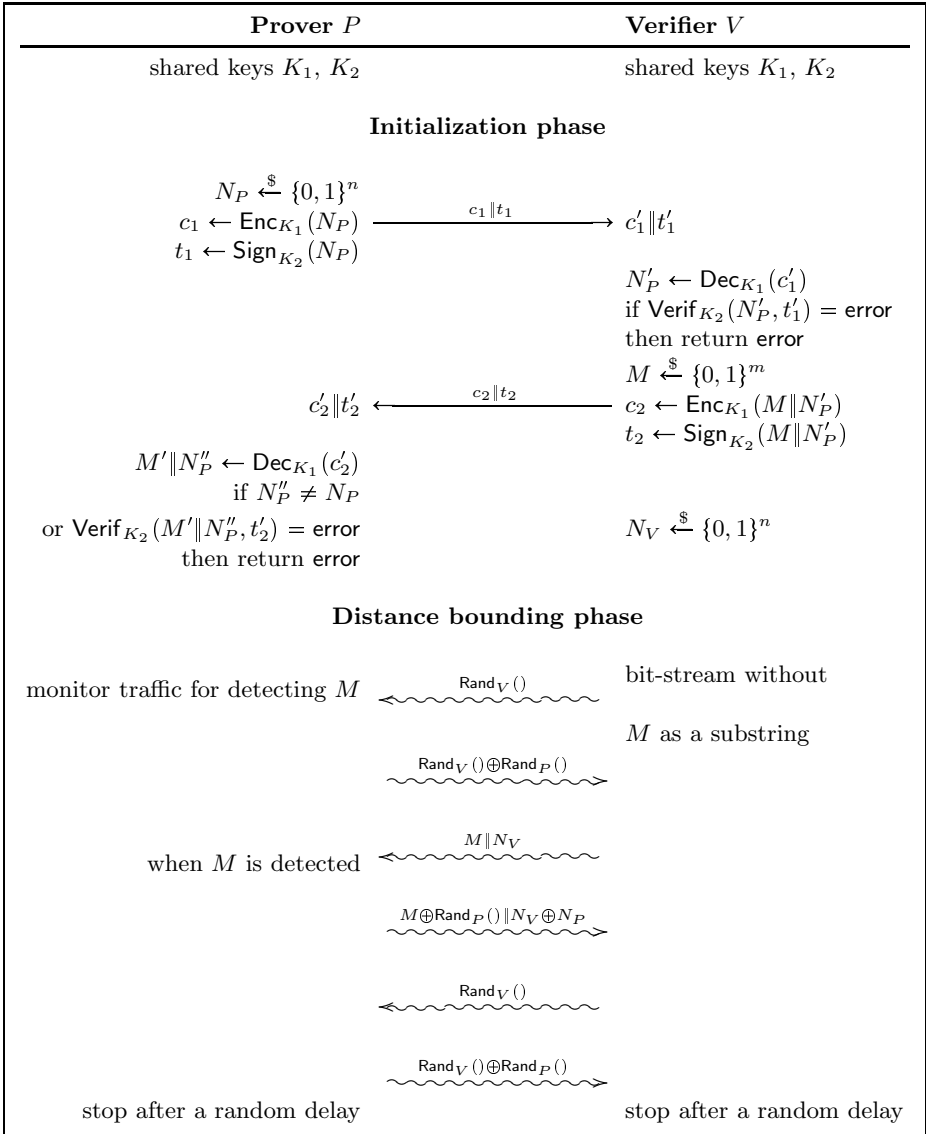## 4 Attacks for the RČ Protocol with Predictable Nonces

This section presents attacks when nonces are predictable, as in stream ciphers or block ciphers in CTR mode. The attacks are simple and only aim to give evidence that unpredictability is necessary.

4.1 Are Nonces Nonces?

The values N_P and N_V are referred to as "nonces" in [19], but defined as random numbers that are not necessarily unique. In cryptography, however, nonces are often defined as "numbers used only once" and have as only requirement their uniqueness, as for stream ciphers or block ciphers in CTR mode; a (long enough) counter can there be used securely. In many applications nonces are public, and need not be unpredictable, as their sole purpose is to ensure the "freshness" of a protocol session.

   In the context of distance bounding protocols, Hancke and Kuhn used "nonces" defined as "unpredictable bitstrings that will never again be used for the same purpose" [6, §3.1]. Ensuring both uniqueness and unpredictability can be achieved by several means:

1. The combination of a counter, ensuring uniqueness, and of a random generator, ensuring unpredictability (e.g. by concatenation of the two values).
2. The use of random numbers and the storage in memory of previously generated numbers to ensure uniqueness.
3. The use of random numbers and of a probabilistic data structure to reduce the memory requirement, compared to the previous solution. For instance, the use of a Bloom filter to encode the set of previously used seed would ensure uniqueness of nonces, with a false positive probability parametrized by the size of the register. Such a solution, however, requires the implementation of an additional—non cryptographic—hash function.

Prover P	**Verifier** V
shared keys K_1, K_2	shared keys K_1, K_2

$$\text{Initialization phase}$$

$N_P \stackrel{\$}{\leftarrow} \{0,1\}^n$
$c_1 \leftarrow \mathsf{Enc}_{K_1}(N_P)$ $\qquad \xrightarrow{\quad c_1 \| t_1 \quad} \qquad c_1' \| t_1'$
$t_1 \leftarrow \mathsf{Sign}_{K_2}(N_P)$

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad N_P' \leftarrow \mathsf{Dec}_{K_1}(c_1')$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ if $\mathsf{Verif}_{K_2}(N_P', t_1') = \mathsf{error}$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ then return error
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad M \stackrel{\$}{\leftarrow} \{0,1\}^m$
$\qquad\qquad c_2' \| t_2' \xleftarrow{\quad c_2 \| t_2 \quad} \quad c_2 \leftarrow \mathsf{Enc}_{K_1}(M \| N_P')$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad t_2 \leftarrow \mathsf{Sign}_{K_2}(M \| N_P')$

$M' \| N_P'' \leftarrow \mathsf{Dec}_{K_1}(c_2')$
if $N_P'' \neq N_P$
or $\mathsf{Verif}_{K_2}(M' \| N_P'', t_2') = \mathsf{error}$ $\qquad\qquad N_V \stackrel{\$}{\leftarrow} \{0,1\}^n$
then return error

$$\text{Distance bounding phase}$$

monitor traffic for detecting M $\quad\overleftarrow{\quad \mathsf{Rand}_V()\quad}\quad$ bit-stream without

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad M$ as a substring

$\qquad\qquad\overrightarrow{\quad \mathsf{Rand}_V() \oplus \mathsf{Rand}_P() \quad}$

when M is detected $\quad\overleftarrow{\quad M \| N_V \quad}$

$\qquad\qquad\overrightarrow{\quad M \oplus \mathsf{Rand}_P() \| N_V \oplus N_P \quad}$

$\qquad\qquad\overleftarrow{\quad \mathsf{Rand}_V() \quad}$

$\qquad\qquad\overrightarrow{\quad \mathsf{Rand}_V() \oplus \mathsf{Rand}_P() \quad}$

stop after a random delay $\qquad\qquad\qquad\qquad\qquad$ stop after a random delay

Fig. 2. Rasmussen and Čapkun's distance bounding protocol, where $\stackrel{\$}{\leftarrow}$ denotes sampling uniformly at random, $\leftarrow$ denotes a simple message transmission, and $\leftrightsquigarrow$ denotes a continuous stream transmission at maximal bit rate

To the best of our knowledge, it is not known how to design nonce generators of type 1 that are both cheap to implement and that do not leak the value of the counter, thus making nonces partially predictable. The other two solutions require non-negligible memory, and are thus inappropriate for the most lightweight devices.

In the following, we present attacks on variants of the RČ protocol that use unique and predictable, or partially predictable, nonces. We then present an attack on the original RČ protocol, exploiting the non-uniqueness of nonces, and propose efficient countermeasures.

For more about the problem of generating and resetting nonces, we refer the reader to the extensive study by Zenner [20]. We conclude this section with Anderson's definition of a nonce, which proves appropriate to the RČ protocol: "The term *nonce* can mean anything that guarantees the freshness of a message. A nonce may, according to the context, be a random number, a serial number, or a random challenge received from a third party. There are subtle differences between the three approaches, such as in the level of resistance they offer to various kinds of replay attacks. But in very low-cost systems, the first two predominate, as it tends to be cheaper to have a communication channel in one direction only." [21, p15].

4.2   Attack with a Predictable N_V

Suppose that given the value of N_V used at some distance bounding session, P can determine the N_V that V will use at the subsequent run of the protocol. It follows that in the distance bounding phase of the said subsequent session, P can start sending $N_V \oplus N_P$ before the complete M is received, thus deceiving V in concluding that P is closer than it actually is. Secure distance bounding is thus not ensured.

If N_V is only partially predictable (for example, a 64-bit N_V is formed of a 32-bit predictable counter followed by a 32-bit random string), then the above attack does not work.

4.3   Attack with a Predictable N_P

Suppose an attacker comes to know the N_P of a given session, and can predict the value of N_P that will be used at the subsequent session. The privacy-breaking attack then consists in identifying N_V by searching for a collision between contiguous n-bit windows from the V-to-P stream and the n-bit windows formed from XORing each contiguous n-bit window of the P-to-V stream with N_P, for the windows chronologically following those of the V-to-P stream. Efficient substring search methods as the Boyer-Moore algorithm [22] may be employed.

This attack requires the recording of the sessions, and allows the attacker to determine (a bound on) the distance of P to V, as computed by the latter.

If N_P is only partially predictable, the attack works with a slightly lower success probability, due to the increased chance of false positives. That probability

depends on the parameters n and ℓ and on the amount of predictable information in N_P.

4.4   Discussion

The analysis in this section suggests that the best choice for P is to use totally unpredictable nonces N_P, and to choose them as random (thus potentially non-unique) numbers for the sake of efficiency. This is exactly how N_P is defined in [19]. The next section shows how to detect and exploit repetitions of nonces, and suggests a modification in the protocol to dissimulate nonces repetition, thus thwarting our attack.

5   Attack for the Original RČ Protocol

We present a passive attack that recovers N_P, N_V, and M for two sessions of the RČ protocol, which allows an attacker to deduce information on the relative distance of P and V during each of those sessions. The distance between P and V does not need to be the same at each session. More precisely, we show how one can exploit repeated occurrences of the same N_P in two distinct sessions to recover the ephemeral secrets of those sessions.

Below we first give a general description of our attack, then we provide a detailed complexity analysis for each part of the attack, with concrete performance figures for typical parameters of the RČ protocol.

5.1   Description of the Attack

The proposed attack observes many sessions of the RČ protocol between P and V, and goes as follows:

1. For each session observed, record synchronously the two data streams exchanged after c_2 is sent, and store the c_1's in a dynamically sorted table. When a value of c_1 has appeared twice—i.e., when an N_P has been repeated—stop recording sessions and delete all previous recordings but those of the two sessions where the repetition occurred.
2. For each of the two sessions with a same N_P, do the following:
   (a) divide the V-to-P stream into n-bit windows $VP_0, VP_1, \ldots$
   (b) divide the P-to-V stream into n-bit windows $PV_0, PV_1, \ldots$
   (c) construct and sort a table containing all $VP_i \oplus PV_j$ values, $0 < i < j$
   Since the two sessions used the same N_P, each of the two tables T_1 and T_2 will contain an element equal to this N_P. Indeed, the XOR between VP_i's and PV_i's will cancel the value of each N_V ($N_V \oplus (N_V \oplus N_P) = N_P$).
3. Search for a collision between an element of T_1 and an element of T_2. If a unique collision is found, then the value is N_P.
4. Given N_P, identify M and N_V in the bit-streams of each session. Count the number of bits between the reception of N_V from the verifier and the reception of P's response to deduce information on the relative positions of P and V.

5.2   Complexity Analysis

We analyse the generic time and space complexity of the proposed attack, assuming that ℓ is the least number of bits sent by either P or V during the distance bounding phase.

Memory required before detecting a collision. Since N_P is n-bit long, an N_P will be repeated after approximately $2^{n/2}$ sessions. One thus needs to record of the order of $2 \cdot \ell \cdot 2^{n/2}$ bits.

Memory to store the tables. There are $W = \ell - n + 1$ distinct windows of n bits in the V-to-P stream. The i-th window, starting from the first one when $i = 1$, is XORed with $W - i - 1$ n-bit windows of the P-to-V stream. Thus, there are in total:

$$N = \sum_{i=1}^{W} (W - 1 - i) = \frac{W^2 - 3W}{2}$$

entries in each table.

Computation to sort the tables. Sorting with comparison-based algorithm (as heap sort or quicksort) takes $O(N \log N)$ comparisons, where N is the number of n-bit words in the table (the above values). As the cost of comparing n-bit elements is arguably proportional to n, this gives a complexity $O(nN \log N)$. Using radix sort, one can achieve complexity $O(nN)$ complexity (see e.g. [23] for a similar analysis). However, note that this analysis ignores the extra cost caused by memory accesses to a large table, which arguably makes the above estimates misleading.

An alternative to classical sort algorithm, discussed in [23], is to use Schimmler's [24] or Schnorr-Shamir's [25] algorithms on a parallel machine. The later sorts N elements using a $\sqrt{N} \times \sqrt{N}$ mesh of N processors in approximately $3\sqrt{N}$ steps.

Number of false alarms. After sorting the two N-element tables, we search for an element common in the two tables. One collision is known to exist, as each table is known to contain N_P. Other collisions that may be found are thus "false positives". The expected number of false alarms is the expected number of collisions besides the known N_P collision, that is $(N^2 - 1)/2^n$.

Efficiency for typical parameters. In section [19, §5.4] is discussed an implementation of the protocol over a communication channel of bit rate $1\,\text{Gbps}$ (i.e., one bit is transmitted each nanosecond) with hidden markers M of $m = 160$ bits, such that the distance bounding phase lasts (approximately) 500 milliseconds. In this case streams of 2^{30} bits are exchanged in each direction. As noted in [19, §5.4], however, an attacker has no way to verify a guess of M, but only a guess of N_P in time approximately N for each of the 2^n possible values.

No value is suggested for the length n of the random numbers N_P and N_V, nor for the length of K_1 or K_2. We shall thus study the efficiency of our attack for the typical values of n, as 32, 64, or 128. Note that the cost of an offline

Table 1. Evaluation of our attack for several typical parameters (length of RBE ℓ, in bits, and length of the nonces n). Time complexity of sorting the table is that of the Schnorr-Shamir parallel algorithm. The number of collisions is at least one, in which case the unique collision corresponds to the solution for N_P. If more than one collision are found, then the solution is known to lie in this set.

(n, ℓ)	sessions monitored	memory required	tables size (N)	sorting time	number of collisions
$(32, 2^{10})$	2^{16}	2^{27}	2^{19}	2^{11}	2^6
$(32, 2^{20})$	2^{16}	2^{37}	2^{39}	2^{21}	2^{45}
$(64, 2^{10})$	2^{32}	2^{43}	2^{19}	2^{10}	1
$(64, 2^{20})$	2^{32}	2^{53}	2^{39}	2^{21}	2^{14}
$(64, 2^{30})$	2^{32}	2^{63}	2^{59}	2^{31}	2^{53}
$(128, 2^{10})$	2^{64}	2^{75}	2^{19}	2^{11}	1
$(128, 2^{20})$	2^{64}	2^{85}	2^{39}	2^{21}	1
$(128, 2^{30})$	2^{64}	2^{95}	2^{59}	2^{31}	1

attack recovering N_P and N_V (and thus M) for a given session has complexity $N2^n$, as observed above.

Table 1 summarizes the complexity of our attack for various combinations of choices of n and ℓ. The theoretical complexity bottleneck (in time and space) is the recording of $2^{n/2}$ sessions of the protocol. The attack is clearly practical when using 32-bit nonces with a 2^{10}-bit long distance bounding phase; increasing the nonce to 64-bit still gives an attack arguably practical ("only" 1 Tb of memory is necessary). In any case, the attack is only applicable if a large number of protocol sessions can be initiated, and if the communication can be captured and recorded at an appropriate rate (which, admittedly, may be impossible for the highest frequencies).

6   Strengthening the RČ Protocol

We propose the following modifications to the RČ protocol to thwart our collision-based attack, and as countermeasures against future attacks on components or implementations of the attacks.

- **Probabilistic encryption**, so that repetitions of N_P cannot be detected (since two ciphertexts of the same N_P are then distinct). This modification alone is sufficient to thwart the attack in §5.
- **Better nonces**: as discussed in §4, unique N_P nonces should be used, if possible, for example by using Bloom filters to save memory at the price of a non-zero false positive probability.
- **Encrypt-then-sign** instead of encrypt-and-sign; the former being at least as secure as the latter, and benefiting of provable security properties (cf. [26]).

– **Distinct keys** for encryption and for authentication; this modification is included in our description of the protocol, but was not explicit in its original definition in [19].

7   Security against Relay Attacks

In this section we analyse the security of the RČ protocol against distance, mafia, and terrorist fraud attacks. For a detailed description of each of the attacks we urge the reader to consult [5].

7.1   Distance Fraud

A distance fraud attack is possible when there is no relationship between the challenge bits and the response bits exchanged during the distance verification. If a fraudulent prover $\overline{P}$ knows when the challenge bits will be sent, it can trick V by sending the response bits before receiving the challenge bits. Thus, V computes a wrong upper bound regarding its physical distance to $\overline{P}$.

In RČ, the response bit-streams are functions of the challenges. More precisely, the response bit-stream $N_v \oplus N_P$ is used for distance-checking. So, the probability that an attacker performs a successful distance fraud attack is upper bounded by $(1/2)^n$ [5], where n is the bit-length of N_V and N_P.

7.2   Mafia Fraud

The best known strategy to launch a successful mafia fraud attack consists in querying the prover P before receiving the challenge from the verifier V and obtaining the right response for this value. When the actual challenge bit is sent to P, in half of all cases, the adversary knows the response beforehand because this value coincides with the value that was previously queried. In the other half of cases, the attacker can answer a random challenge. If a protocol is vulnerable to the above attack, the success probability with which a mafia fraud attack can succeed is bounded by $(\frac{3}{4})^n$, which is higher than the optimal value $(\frac{1}{2})^n$, where n is the number of bits exchanged in the RBE.

In [27], Munilla and Peinado proposed a protocol inspired by [6] in which the success probability of an adversary to accomplish a mafia fraud attack is reduced. The proposal is based on *void* challenges in order to detect that P is not waiting to receive the challenge bits. A void challenge is a challenge which V intentionally leaves without sending. That is, the challenge bit sent by P can take three different values $\{0, 1, void\}$. If V detects that a response bit is received during the interval of a void challenge, the dishonest prover $\overline{P}$ is detected. Although, from a theoretical point of view the proposal is interesting, the feasibility of this scheme is questionable since it requires three physical states.

Alternatively, Rasmussen and Čapkun introduced the use of a hidden marker M which provides the advantage of detecting whether P is waiting to receive the challenge bit-streams. The position of the hidden marker M is randomly chosen

in the setup window. The adversary may try to guess the start of hidden marker M and then may use the strategy of asking in advance to know the response of the bit-stream N_V. Let r be the bit rate of the channel and consider a setup window of ω_s seconds. The success probability with which a mafia attack can succeed is upper bounded by:

$$\left(\frac{1}{2}\right)^{r \cdot \omega_s} \cdot \left(\frac{3}{4}\right)^{n}$$

where n is the bit-length of N_V and N_P. Thus, the probability of running a successful mafia fraud attack is reduced. Nevertheless, there are some drawbacks that the reader should be aware. First of all, the number of bits (streams) exchanged during the RBE is significantly increased in comparison with when only the challenge bits (N_V bit-stream) are sent. Secondly, the time required to complete the distance checking is thus longer, which means a decrease in the efficiency of the protocol and an increase in the power consumption required by P.

7.3   Terrorist Fraud

This attack can be viewed as an extension of the mafia fraud. In this fraud, the dishonest prover collaborates with a terrorist prover $\overline{P}$: P uses $\overline{P}$ to convince V that he lies in close proximity, while he does not. Despite the cooperation between P and $\overline{P}$, it is assumed that the long-term secret key of P is not revealed to $\overline{P}$.

   In the RČ protocol, a dishonest P can reveal N_P and M to a terrorist prover $\overline{P}$, without compromising the secret keys K_1 and K_2. Then, the distance-bounding phase is executed between V and $\overline{P}$, while the latter is nearer to V than P is. Thus, $\overline{P}$ misleads V into believing that P is located nearer. Thus, the probability that an attacker runs a successfully terrorist fraud attack is 1. This attack can be avoided by making interdependent the bit-streams $\{N_P, M\}$ and the secret keys, as in [28, 29].

8   Conclusions

We reported the first third-party cryptanalysis of the Rasmussen-Čapkun privacy-preserving distance bounding protocol: we identified strengths and weaknesses of that protocol, and presented a detailed attack exploiting nonce collisions. We proposed simple modifications to the protocol to thwart our attack and to enhance the overall security of the protocol. These modifications include the use of probabilistic encryption and the implementation of an encrypt-then-sign (rather than encrypt-and-sign) scheme. Furthermore, we investigated the security of the RČ protocol against distance, terrorist and mafia fraud attacks.

   Our detailed definition of the protocol and the results of our in-depth security analysis are complementary to the preliminary security and implementation analysis in [19]. The natural next step is a proof-of-concept implementation of the Rasmussen-Čapkun protocol.

References

1. Papadimitratos, P., Poturalski, M., Schaller, P., Lafourcade, P., Basin, D., Čapkun, S., Hubaux, J.P.: Secure neighborhood discovery: a fundamental element for mobile ad hoc networking. IEEE Communications Magazine 46(2), 132–139 (2008)
2. Sastry, N., Shankar, U., Wagner, D.: Secure verification of location claims. In: Proceedings of the 2nd ACM Workshop on Wireless Security (WiSe 2003), pp. 1–10 (2003)
3. BroadcastEngineering: TV GLOBO TVDR, Article available online at `http://broadcastengineering.com/excellence-awards/tv-globo-tdvr/` (accessed January 11, 2011)
4. Francillion, A., Danev, B., Čapkun, S.: Relay attacks on passive keyless entry and start systems in modern cars. In: Cryptology ePrint Archive: Report 2010/332 (2010) (to appear in proceedings of NDSS 2011)
5. Brands, S., Chaum, D.: Distance Bounding Protocols. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 344–359. Springer, Heidelberg (1994)
6. Hancke, G.P., Kuhn, M.G.: An RFID distance bounding protocol. In: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communication Networks (SECURECOMM 2005), pp. 67–73 (2005)
7. Tu, Y.J., Piramuthu, S.: RFID Distance Bounding Protocols. In: Proceedings of the First International EURASIP Workshop on RFID Technology (2007)
8. Munilla, J., Peinado, A.: 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)
9. Kim, C.H., Avoine, G., Koeune, F., Standaert, F.X., Pereira, O.: The Swiss-Knife RFID Distance Bounding Protocol. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 98–115. Springer, Heidelberg (2009)
10. Lee, J.Y., Scholtz, R.A.: Ranging in a dense multipath environment using an UWB radio link. IEEE Journal on Selected Areas in Communications 20(9) (2002)
11. Gezici, S., Tian, Z., Biannakis, G.B., Kobayashi, H., Molisch, A.F., Poor, V., Sahinoglu, Z.: Localization via ultra-wideband radius: a look at positioning aspects for future sensor networks. IEEE Signal Processing Magazine 22(4), 70–84 (2005)
12. Kuhn, M., Luecken, H., Tippenhauer, N.O.: UWB impulse radio based distance bounding. In: Proceedings of the 7th Workshop on Positioning, Navigation and Communication 2010, WPNC 2010 (2010)
13. Čapkun, S., Buttyán, L., Hubaux, J.P.: SECTOR: secure tracking of node encounters in multi-hop wireless networks. In: Proceedings of the 2003 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2003), pp. 21–32 (2003)
14. Singelee, D., Preneel, B.: Location verification using secure distance bounding protocols. In: Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS 2005), pp. 834–840 (2005)

15. Clulow, J., Hancke, G.P., Kuhn, M.G., Moore, T.: So near and yet so far: Distance-bounding attacks in wireless networks. In: Proceedings of the Third European Workshop on Security and Privacy in Ad hoc and Sensor Networks, pp. 83–97 (2006)
16. Meadows, C., Syverson, P., Chang, L.: Towards more efficient distance bounding protocols for use in sensor networks. In: Proceedings of the International Conference on Security and Privacy in Communication Networks (SECURECOMM 2006), pp. 1–5 (2006)
17. Čapkun, S., Hubaux, J.P.: Secure positioning of wireless devices with application to sensor networks. In: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), Miami, FL, USA, pp. 1917–1928 (2005)
18. Hu, Y.C., Perrig, A., Johnson, D.B.: Packet leashes: A defense against wormhole attacks in wireless networks. In: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM 2003), San Francisco, CA, USA, vol. 3, pp. 1976–1986 (2003)
19. Rasmussen, K.B., Čapkun, S.: Location privacy of distance bounding protocols. In: Proceedings of the 15th ACM Conference on Computer and Communications Security (ACM CCS 2008), pp. 149–160 (2008)
20. Zenner, E.: Nonce Generators and the Nonce Reset Problem. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C.A. (eds.) ISC 2009. LNCS, vol. 5735, pp. 411–426. Springer, Heidelberg (2009)
21. Anderson, R.J.: Security Engineering: A Guide to Building Dependable Distributed Systems, 2nd edn. Wiley (2008)
22. Boyer, R.S., Moore, J.S.: A fast string searching algorithm. Communications of the ACM 20(10), 762–772 (1977)
23. Bernstein, D.J.: Better price-performance rations for generalized birthday attacks. In: Proceedings of the International Conference on Special-purpose Hardware for Attacking Cryptographic Systems, SHARCS 2007 (2007)
24. Schimmler, M.: Fast sorting on the instruction systolic array. Technical Report 8709, Christian-Albrechts Universität Kiel (1987)
25. Schnorr, C.P., Shamir, A.: An optimal sorting algorithm for mesh connected computers. In: Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC 1986), pp. 255–263 (1986)
26. Percival, C.: Encrypt-then-MAC. Blog entry on Daemonic Dispatches, http://www.daemonology.net/blog/2009-06-24-encrypt-then-mac.html (accessed January 11, 2011)
27. Munilla, J., Peinado, A.: 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. Bussard, L., Bagga, W.: Distance-bounding proof of knowledge to avoid real-time attacks. In: Security and Privacy in the Age of Ubiquitous Computing. IFIP AICT, pp. 223–238 (2005)
29. Reid, J., Gonzalez Nieto, J.M., Tang, T., Senadji, B.: Detecting relay attacks with timing-based protocols. In: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS 2007), pp. 204–213. ACM, Singapore (2007)