Distance Bounding in Noisy Environments

Dave Singelée and Bart Preneel

ESAT-COSIC, K.U. Leuven,
Kasteelpark Arenberg 10, 3001 Heverlee-Leuven, Belgium,
{Dave.Singelee, Bart.Preneel}@esat.kuleuven.be

Abstract. Location information can be used to enhance mutual entity authentication protocols in wireless ad-hoc networks. More specifically, distance bounding protocols have been introduced by Brands and Chaum at Eurocrypt'93 to preclude distance fraud and mafia fraud attacks, in which a local impersonator exploits a remote honest user. Hancke and Kuhn have extended these protocols to cope with noisy channels. This paper presents an improved distance bounding protocol for noisy channels that offers a substantial reduction in the number of communication rounds compared to the Hancke and Kuhn protocol. The main idea is to use binary codes to correct bit errors occurring during the fast bit exchanges. Our protocol is perfectly suitable to be employed in noisy wireless environments such as RFID.

Keywords: *Secure Localization, Distance Bounding, Wireless Sensor Networks, Entity Authentication.*

1 Introduction

1.1 Proximity based authentication

In mobile networks, location information can be used to enhance mutual entity authentication protocols. Entities which are in a specific location or within a certain range of a device (*"the verifier"*) are granted some privileges, in contrast to all other entities. In most scenarios, one would like to determine an upper-bound on the distance to another entity. For instance, one could conduct a cryptographic identification protocol at the entrance to a building. Only entities with the correct credentials and who are not more than a few meters away are granted access to the building.

The concept of *proximity based authentication* is graphically depicted in Fig. 1. Authentication requests originating from devices that are located within the range d of the verifier V are accepted, all other requests are rejected. So in Fig. 1, authentication requests originating from device A are accepted (and as a consequence, A is granted some privileges), while the requests of B are rejected. Contact-less smart-cards and RFID tokens are often used for proximity-based authentication [1]. Such mobile devices have very limited processing power. Therefore, one has to employ low-cost cryptographic primitives to authenticate the mobile devices, and verify the distance between both parties.

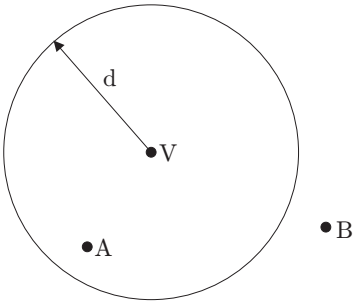

Fig. 1. Proximity based authentication

How can one securely verify if a certain device is within a specific range? There are different methods to accomplish this. Just asking the location will not be sufficient because the verifier does not trust the prover. If each device contains a trusted GPS receiver, one could use the GPS coordinates to perform location based authentication [2]. There are however some drawbacks to this method. E.g., it is very expensive and can not be used indoor. One could also apply *distance bounding protocols*. These protocols enable the verifying party to determine an upper-bound on the distance between itself and a prover, who claims to be within a certain range.

Basically, distance bounding protocols combine physical and cryptographic properties to determine an upper-bound on the distance between verifier and prover. They allow the prover to authenticate itself to the verifier, and in the same time enable the verifying party to check if the prover is located within a certain range. Distance bounding techniques can measure the received signal strength (RSS) [3], measure the angle of arrival (AoA), or measure the time of flight (ToF) to estimate an upper-bound on the distance. The first two techniques (RSS and AoA) are typically discarded because of security reasons: e.g., an attacker can construct a directional antenna to largely increase the sending or receiving range [4, 5]. This only leaves measuring the time of flight as a possible technique for secure distance bounding protocols.

1.2 Organization of the paper

This paper is organized as follows. In the introduction, we briefly discussed the idea of proximity based authentication in mobile ad-hoc networks. We put forward the idea of employing distance bounding protocols. The general principles of these protocols are discussed more in detail in section 2. Section 3 and 4 describe two important distance bounding protocols: Brands' and Chaum's protocol, and Hancke's and Kuhn's protocol respectively. In section 5, we show how to adapt the Mutual Authentication with Distance Bounding (MAD) protocol of Čapkun et al. (extended version of the Brands' and Chaum's protocol) to make it noise resilient. This protocol requires a lower number of communication

rounds than other noise resilient distance bounding protocols, as will be shown in section 6. Finally, section 7 concludes the paper.

2 Background

2.1 How do distance bounding protocols work?

Secure distance bounding protocols measure the time of flight to determine an upper-bound on the distance between prover and verifier. This is typically done during a challenge–response protocol, the main building block of the distance bounding protocol. During n fast bit exchanges, the time between sending a challenge and receiving the response is measured. Multiplying the time of flight with the propagation speed of the communication medium gives the distance between prover and verifier.

One should however take into account some important details. It should be impossible for the prover to send the response before receiving the challenge [6]. This implies that the response should be dependent on the (random) challenge. A second remark is that a challenge–response protocol is not enough. After execution of this protocol, the verifier only knows that some party is close. But how does one know that this entity is the prover? This problem arises for example in the Echo protocol [7]. That is why the prover has to identify itself somewhere in the scheme (not necessarily in the challenge–response protocol itself). Finally, one should notice that the round trip time is not equal to the propagation time. It takes some time to compute and transmit the response. This processing delay should be as small as possible compared to the propagation time, because we are only interested in the latter. Let's examine two communication technologies: (ultra-)sound and electromagnetic signals.

Ultra-sound: (Ultra-)sound is interesting to measure distances because it is relatively slow. The processing delay can hence be neglected compared to the propagation time and the accuracy of the measurements is not very critical. An example of a protocol using this technique can be found in [8]. There are however some security problems. (Ultra-)sound is not resistant to physically present attackers. Such an attacker can modify the medium (e.g., sound travels faster through metal than through the air) or use wormholes (e.g., by retransmitting the signal using electromagnetic waves) to claim that he is closer than he really is. By delaying the response, he can also claim to be further away.

Electromagnetic signals: If we don't take into account quantum cryptography, an active attacker can not use wormholes. The signals travel with the speed of light and nothing propagates faster. This means that an attacker can only claim to be further away than he really is (by delaying the response). There are however some practical issues. The verifier has to be able to measure the round trip time with very high precision. A small deviation of the time of flight influences the estimated distance a lot. A similar problem is estimating the processing delay. One has to design the distance bounding

protocol in such a way that the processing delay can be neglected to the (very small) time of flight.

2.2 Attack scenarios

By employing the principle of distance bounding attacks in a clever way, one can preclude one or more fundamental attacks.

One wants to prevent a dishonest prover claiming to be closer than he really is. This attack is called **distance fraud attack**. It is relatively easy to design a distance bounding protocol which prevents this type of attack. An overview of location mechanisms that are resistant to distance fraud attacks (sometimes only partially) can be found in [9].

Mafia fraud attacks, also called *relay attacks*, were first described in [10]. In this attack scenario, both prover and verifier are honest, but a malicious intruder is performing the fraud. It is a man–in–the–middle attack where the intruder I is modeled as a malicious prover $\bar{P}$ and verifier $\bar{V}$ cooperating together, as shown in Fig. 2. The malicious verifier $\bar{V}$ interacts with the honest prover P and the malicious prover $\bar{P}$ interacts with the honest verifier V. The physical distance between the intruder and the verifier is small. This attack enables the intruder to identify itself to V as P being close to V, without any of P and V noticing the attack.

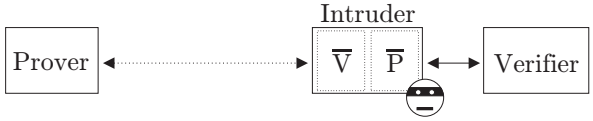

Fig. 2. Mafia fraud attack

Terrorist fraud attacks [10] are an interesting extension of the mafia fraud attack. The intruder and the prover will collaborate in this attack. This implies that a protocol which is resistant to terrorist fraud attacks, also prevents mafia fraud attacks. The terrorist fraud attack is shown in Fig. 3. The dishonest prover uses the intruder to convince the honest verifier that he is close, while in fact he is located at a large distance. The intruder does not know the private or secret key. This certainly has to be emphasized! If the intruder would know this private key, then it is impossible to make a distinction between the intruder and the prover, since distance bounding protocols only check if a party which knows the private key is close to the verifier. Several distance bounding protocols are resistant to terrorist fraud attacks [9, 11, 12].

2.3 Design principles for secure distance bounding protocols

Without going in too many details, one can formulate the following (simplified) cryptographic design principles of distance bounding protocols:

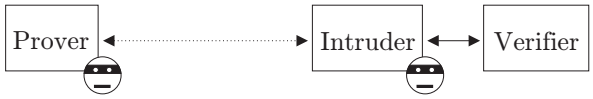

Fig. 3. Terrorist fraud attack

- In at least one of the messages of the distance bounding protocol, the prover has to identify itself.
- To prevent mafia fraud attacks, the distance bounding protocol contains a challenge–response protocol that consists of a series of rapid bit exchanges (n rounds in total) [6]. By measuring the round trip time in each of the n rounds, the verifier can determine an upper-bound on the distance between verifier and prover. To prevent the prover sending the response too soon, the challenge has to be random and unpredictable, and the response has to depend on this challenge.
- To avoid terrorist fraud attacks, one has to make sure that the fast bit exchanges and the phase in which the prover identifies itself, are intermingled in a cryptographic way. It has to be impossible to split the distance bounding protocol in these two distinct phases. There are at least two possibilities to accomplish this. Or one uses the private (or symmetric) key during the fast bit exchanges, or one uses trusted hardware. For more details, we refer to [11].

The design principles described above are not enough. Hancke et al. show that one has to optimize the choice of communication medium and transmission format according to the following four principles [13], if one wants to prevent certain "physical" attacks:

- Use a communication medium with a propagation speed as close as possible to the physical limit for propagating information through space-time.
- Use a communication format in which only a single bit is transmitted and the recipient can instantly react on its reception.
- Minimize the length of the symbol used to represent this single bit.
- Design the distance bounding protocol to cope with bit errors taking place during the rapid bit exchanges.

3 Brands' and Chaum's Distance Bounding Protocol

In 1993, S. Brands and D. Chaum presented their distance bounding protocol [6]. This clever protocol prevents mafia fraud attacks and embodies a series of n rounds (n is a security parameter). Each round consists of a single bit challenge and a rapid single bit response. The delay time for receiving the responses enables the verifier to compute an upper-bound on the distance. After **correct** execution of the distance bounding protocol, the verifier knows that an entity in possession of a certain secret is in the vicinity.

The protocol is carried out as follows. It contains 3 phases. First, the prover sends out a commitment to n random values m_i. Next, a series of n fast bit exchanges is performed. The verifier sends a random challenge α_i to the prover. This challenge is XOR'ed with the value m_i and the result (β_i) is sent back to the verifier. After the n fast bit exchanges, the prover opens the commitment and signs the bitstring y, which embodies the concatenation of the challenges α_i and the responses β_i. If the signature is correct, the protocol is successful. In each of the n rounds, an attacker has a $\frac{1}{2}$ probability of sending a correct response [6]. Note that in every of the n rounds, the prover has to compute the *XOR* of two bits. This can be done very efficiently in hardware.

Čapkun et al. extended the protocol to MAD, a mutual authentication protocol using distance bounding [14]. This protocol has the advantage that both parties can estimate an upper-bound on the distance between themselves, and learn each others' identity, which is not the case in the original protocol of Brands and Chaum. All the advantages of the distance bounding protocol of Brands and Chaum remain valid for MAD.

4 Hancke's and Kuhn's Protocol for RFID

Both the distance bounding protocol of Brands and Chaum [6], and the MAD protocol [14], were not designed to cope with bit errors during the fast bit exchanges. A single bit error causes the protocol to fail. This can be an important problem in noisy environments like RFID. That is why Hancke and Kuhn proposed a distance bounding protocol [15] that can easily be extended to deal with bit errors.

The protocol is carried out as follows. First, prover and verifier exchange a random nonce (N_P and N_V respectively). Both parties then use a pseudorandom function (typically a *MAC* [16]) to compute two n-bit sequences $v^{(0)}$ and $v^{(1)}$ (more in detail: $MAC_K(N_V, N_P) = v^{(0)}|v^{(1)}$). Next, a series of n fast bit exchanges is performed. In each round, the verifier sends a random single bit challenge C_i to the prover. If this challenge equals 0, then the prover responds with the *i-th* bit of $v^{(0)}$. If the challenge equals 1, then the prover sends the *i-th* bit of $v^{(1)}$. If all responses are correct, the protocol succeeds. In each round, an attacker has a $\frac{3}{4}$ probability of sending a correct response. After **correct** execution of the distance bounding protocol, the verifier knows that an entity in possession of the secret key K is in the vicinity.

If we compare the Hancke and Kuhn distance bounding protocol with the Brands and Chaum protocol, we notice that the latter requires a signature to be sent at the end of the protocol, while the former stops after the execution of the n fast bit exchanges. So the Brands and Chaum protocol requires more bits to be interchanged on the slower communication channel. Fortunately, this does not influence the cost a lot, since this is primarily related to the number of fast bit exchanges. Munilla et al. proposed to use *"void challenges"* in the Hancke and Kuhn protocol [17] to improve security. However the disadvantage of their solution is that is requires 3 (physical) states: 0, 1 and *void*.

The Hancke and Kuhn protocol can easily be adapted to make it noise resilient. First one has to select a security parameter x. This parameter denotes the number of bit errors that are allowed during the n fast bit exchanges, and depends on the bit error rate. The distance bounding protocol succeeds if at least $(n-x)$ of the responses sent by the prover were correct. The security parameter x has to be chosen very carefully. Incrementing the number of allowed errors x increases the false-acceptance ratio dramatically. A more detailed discussion on the influence of the different security parameters will be presented in Sect. 6.

5 Noise Resilient Mutual Authentication with Distance Bounding

As discussed in Sect. 3, the MAD protocol of Čapkun et al. offers mutual authentication. So if the application requires that both prover and verifier authenticate each other, one should use the MAD protocol. In all other cases, the distance bounding protocol of Brands and Chaum is preferred, because it requires less messages to be exchanged (except of the number of fast bit exchanges, this is equal for both distance bounding protocols). In all other points of view, both protocols are very similar. In the rest of this paper, we will assume that mutual authentication is required, and use the MAD protocol. But our suggested adaptations and numerical results equally apply to the Brands and Chaum protocol.

The MAD protocol has the nice property that in each of the n rounds of the fast bit exchange, an attacker only has a $\frac{1}{2}$ probability of replying to the verifier with a correct response. Another advantage is the resistance to terrorist fraud attacks, when it is executed in trusted hardware [11]. On the other hand, the distance bounding protocol of Hancke and Kuhn can be easily made resilient to bit errors during the fast bit exchanges, which is a very desirable feature. It would be ideal to combine all the good properties of both distance bounding protocols.

A trivial way of making the MAD protocol noise resilient, is exchanging all challenges and responses again on a slower communication channel with error correction (of course, this has to be done after the fast bit exchanges). However, this is not very efficient. We will now present an efficient modification of the MAD protocol, which is also resilient to some bit errors (we allow x bit errors in total) during the fast bit exchanges. Our protocol, in which the two parties (denoted by *Alice* and *Bob*) will authenticate each other, is shown in Fig. 4.

The protocol is carried out as follows. First, both parties agree on a (n, k) *Error Correcting Code (ECC)* to use in this distance bounding protocol. In order to correct at least x bit errors during the fast bit exchanges, this binary code should have a minimal Hamming distance d_{min} such that $x = \lfloor \frac{d_{min}-1}{2} \rfloor$. More information on which (n, k) error correcting code to use for a given distance d_{min} can be found in [18–21]. Next, *Alice* and *Bob* generate k random bits ($r_1, \ldots, r_k$ and $s_1, \ldots, s_k$ respectively). These k bits are extended to a n-bit bitstring ($r_1, \ldots, r_n$ and $s_1, \ldots, s_n$) by applying the error correcting code described

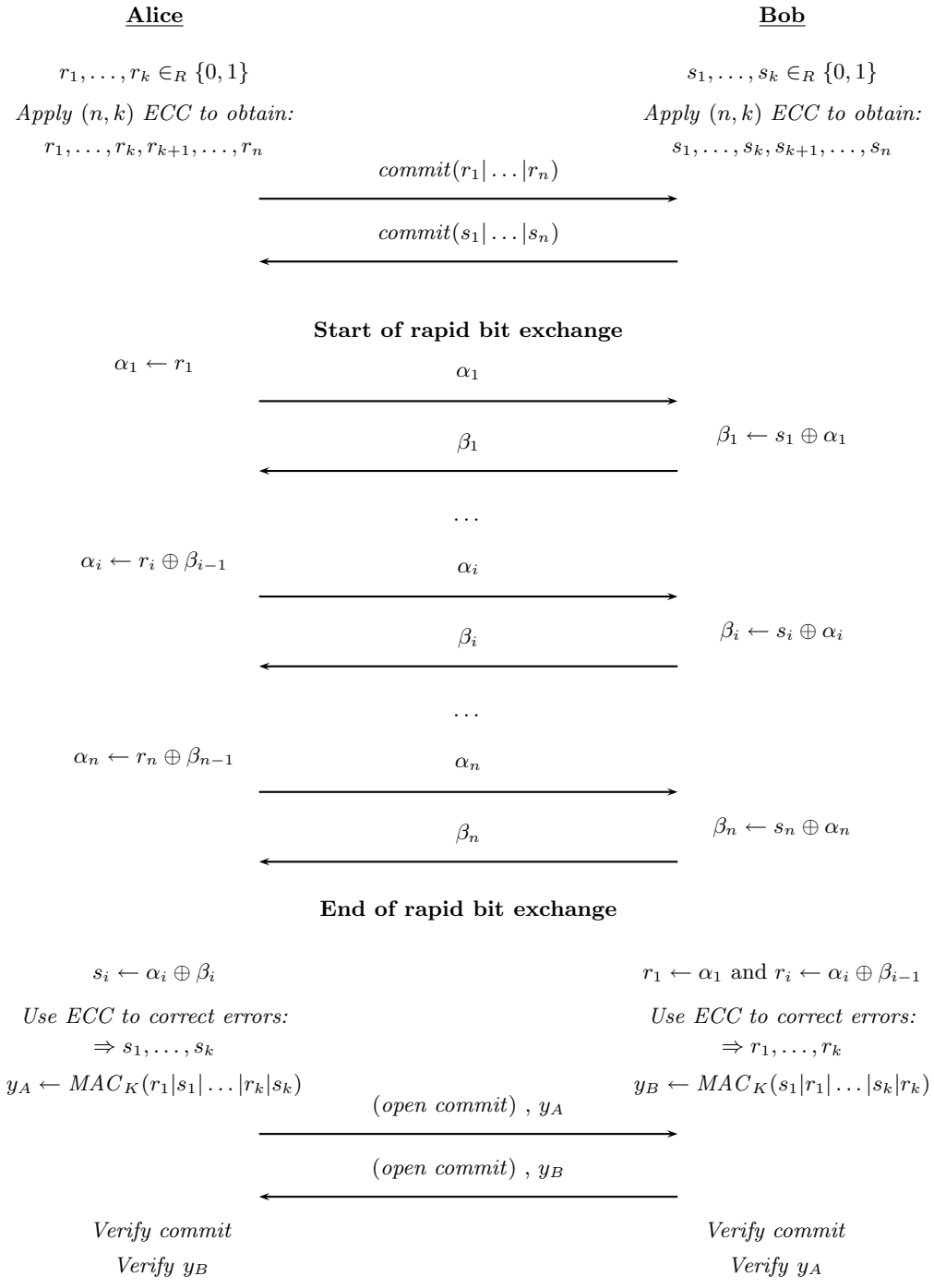

Fig. 4. Noise resilient mutual authentication with distance bounding protocol

above and a commitment to this bitstring is sent to the other party. During the fast bit exchanges, the following two steps are repeated n times:

- *Alice* sends the bit α_i to *Bob* where $\alpha_1 = r_1$ and $\alpha_i = r_i \oplus \beta_{i-1}$.
- *Bob* sends the bit β_i to *Alice* where $\beta_i = s_i \oplus \alpha_i$.

During the first k fast bit exchanges, the time between sending α_i and receiving β_i (or sending β_i and receiving α_{i+1}) is measured. These measurements determine an upper-bound on the estimation of the distance between *Alice* and *Bob*. During the last $(n-k)$ rounds, the round trip time should preferably be within a certain margin of the average round trip time measured during the first k rounds (if the RTT is higher than expected, one adapts the estimation of the distance between both parties, if it is lower than expected, the estimation of the distance is not adapted). After the n fast bit exchanges, both parties use the (n, k) ECC to correct bit errors (each party can correct a maximum of x bit failures) and this way recover the bits $s_1, \ldots, s_k$ and $r_1, \ldots, r_k$ respectively. Finally, Alice (and Bob) compute a MAC on the concatenation of r_i and s_i (or s_i and r_i) and open the commitment sent in the beginning of the protocol. If the MAC is correct, the protocol is successful. In each of the first k rounds, an attacker has a $\frac{1}{2}$ probability of sending a correct response.

6 Performance Analysis

6.1 False-rejection and false-acceptance ratio

In our analysis, we assume that the fast communication channel used during the rapid bit exchanges is symmetric. So a bit error is as likely to occur in a challenge as in a response. We also assume that a bit error is independent of previous bit errors. The bit error rate is denoted by P_b.

Before numerically analyzing and deriving the statistical properties of our distance bounding protocol, let us first clearly define the notion of a *round* (during the fast bit exchange). This definition depends on the distance bounding protocol that is being used. In the Hancke and Kuhn protocol, we define a **round** as a challenge and the corresponding response. In our noise resilient MAD protocol, a **round** are two consecutive messages (so α_i and β_i, or β_i and α_{i+1}).

Some of the challenges and/or responses will be corrupted by noise. The probability that a round fails is denoted by ε. A **round fails** if the verifying party receives an incorrect response, or if one of the parties in our noise resilient MAD protocol gets a corrupted bit $\overline{r_i}$ or $\overline{s_i}$. Let us first have a look to the Hancke and Kuhn protocol. A bit error can appear in the challenge, or in the response (both with probability P_b). We neglect the probability that a bit error occurs in both messages. If the prover receives an incorrect challenge, he still has a $\frac{1}{2}$ probability of sending the correct response (this event happens when the responses for both the challenges 0 and 1 are equal). If the verifier receives a corrupted response, the round fails certainly. So one can easily compute the probability ε that a round fails in the Hancke and Kuhn distance bounding

protocol. It is shown in Eq. 1. In our noise resilient MAD protocol, a round fails by definition with 100% probability when a bit α_i or β_i is corrupted. The probability ε that a round fails in our noise resilient MAD protocol is shown in Eq. 2.

$$\varepsilon = \frac{3}{2} P_b \tag{1}$$

$$\varepsilon = 2P_b \tag{2}$$

We can now compute the false-rejection and false-acceptance ratio, two important parameters to evaluate (noise resilient) distance bounding protocols. An honest prover is falsely rejected if more than x bit errors occur during the fast bit exchanges (which consist out of n rounds). The false-rejection ratio is shown in Eq. 3, and depends on the probability ε. This expression is valid for both distance bounding protocols.

$$P_{FR} = \sum_{i=0}^{n-x-1} \binom{n}{i} \cdot (1-\varepsilon)^i \cdot \varepsilon^{(n-i)} \tag{3}$$

An attacker can use the uncertainty of which bits are corrupted by noise, in its advantage. In the worst case, no bit errors occur, but the (honest) verifier expects a maximum of x bit errors. As a consequence, an attacker only has to guess $(n-x)$ responses right in the Hancke and Kuhn distance bounding protocol to perform a successful attack (without taking into account noise, an attacker should have to guess all n responses correctly to be successful). The false-acceptance ratio of the Hancke and Kuhn protocol is displayed in Eq. 4.

$$P_{FA} = \sum_{i=n-x}^{n} \binom{n}{i} \cdot \left(\frac{3}{4}\right)^i \cdot \left(\frac{1}{4}\right)^{(n-i)} \tag{4}$$

The situation is slightly different in our noise resilient MAD protocol. To be successful, an attacker has to guess all first k bits r_i (or s_i) right[1]. The last $(n-k)$ bits depend of the first k bits and can be easily computed by applying the (n,k) error correcting code. These $(n-k)$ bits do not offer extra security. The false-acceptance ratio of our noise resilient MAD protocol is shown in Eq. 5.

$$P_{FA} = \left(\frac{1}{2}\right)^k \tag{5}$$

6.2 Numerical results

Both noise resilient distance bounding protocols have some interesting characteristics. We will now compare both protocols, and have a closer look to the most interesting properties.

[1] The reason is that the number of allowed errors x is always strictly smaller than the minimal Hamming distance d_{min} of the (n,k) error correcting code.

An attacker has a major advantage when bit errors due to noise can appear. In the worst case scenario, an honest verifier expects to receive some corrupted bits due to noise, while in fact there is no noise at all. As a direct consequence, an attacker can obtain a major advantage. Whenever he guesses a response wrongly, he can blame it to the noise. As long as an attacker has a maximum of x wrong guesses, the Hancke and Kuhn distance bounding protocol will be successful (because the verifier thinks the incorrect bits were corrupted by noise). The more errors that are allowed, the larger the false-acceptance ratio. The same property is also valid for our noise-resilient MAD protocol. For a fixed number of rounds n, the more errors x that have to be corrected, the smaller the parameter k has to be [18–21]. And because only the first k rounds of the fast bit exchanges contribute to the security, the false-acceptance ratio will increase with decreasing k. This property is demonstrated for both distance bounding protocols in table 1. In this numerical example, the number of rounds n is 37, and the bit error rate P_b is 0.01. The information on which error correcting code to use in our noise resilient MAD protocol, is based on [18]. The results in table 1 clearly show that the false-acceptance ratio increases significantly with increasing number of allowed errors x. One can also notice that the false-acceptance ratio is remarkably smaller in our noise resilient MAD protocol (several orders of magnitude). We will discuss this constatation later in this section.

Table 1. Influence of the number of allowed errors x on the false-acceptance ratio

Hancke and Kuhn protocol	
# allowed errors x	P_{FA}
$x = 4$	0.0284
$x = 3$	0.0089
$x = 2$	0.0021
$x = 1$	$3.1784 \cdot 10^{-4}$
$x = 0$	$2.3838 \cdot 10^{-5}$

Noise resilient MAD protocol	
# allowed errors x	P_{FA}
$(37, 16)$ ECC, $x = 4$	$1.5259 \cdot 10^{-5}$
$(37, 22)$ ECC, $x = 3$	$2.3842 \cdot 10^{-7}$
$(37, 26)$ ECC, $x = 2$	$1.4901 \cdot 10^{-8}$
$(37, 31)$ ECC, $x = 1$	$4.6566 \cdot 10^{-10}$
$(37, 37)$ ECC, $x = 0$	$7.2760 \cdot 10^{-12}$

The false-rejection ratio is slightly lower in the Hancke and Kuhn distance bounding protocol. Whereas noise helps an attacker to deceive an honest verifier, it is disadvantageous for an honest prover behaving correctly. The higher the bit error rate P_b, the higher the probability that the distance bounding protocol will fail because of too many bit errors during the fast bit exchanges. If no bit errors occur during the fast bit exchange phase, an honest prover will always be able to authenticate itself successfully. To decrease the false-rejection ratio, one has to allow more bit errors to take place (denoted by x) for a fixed number of rounds n, or decrease the number of rounds (without changing x). The choice of the parameter x has to be in accordance to the expected number of errors, which depends on the number of rounds n and the bit error rate P_b (of course, one has to build in a certain safety margin).

As demonstrated in Eq. 1 and 2 in Sect. 6.1, the probability ε of a round to fail is higher in our noise resilient MAD protocol than in the Hancke and Kuhn distance bounding protocol. A direct consequence of this fact, is that the false-rejection ratio is lower in the Hancke and Kuhn protocol (for equal number of rounds n and allowed errors x). This property is demonstrated in table 2. In this numerical example, the number of rounds n is 37, and the bit error rate P_b is 0.01. Note that the difference in false-rejection ratio between both distance bounding protocols is relatively small. When the number of rounds n is larger (e.g., around 50), one should allow one or two more errors to occur in our MAD resilient to keep the false-rejection ratio comparable in both protocols (e.g., for $n = 47$: $P_{FR}(Hancke, x = 9) = 1.7985 \cdot 10^{-9} \approx P_{FR}(MAD, x = 10) = 1.8353 \cdot 10^{-9}$).

Table 2. Comparison of the false-rejection ratio

# allowed errors x	Hancke-Kuhn: P_{FR}	Noise Res. MAD: P_{FR}
$x = 6$	$1.1849 \cdot 10^{-6}$	$7.7770 \cdot 10^{-6}$
$x = 5$	$1.7760 \cdot 10^{-5}$	$8.7314 \cdot 10^{-5}$
$x = 4$	$2.2184 \cdot 10^{-4}$	$8.1806 \cdot 10^{-4}$
$x = 3$	0.0023	0.0062
$x = 2$	0.0179	0.0375
$x = 1$	0.1062	0.1689
$x = 0$	0.4283	0.5265

The false-acceptance ratio is significantly higher in the Hancke and Kuhn distance bounding protocol. As demonstrated above, to decrease the false-acceptance ratio, one has to allow less bit errors to take place (denoted by x) for a fixed number of rounds n, or increase the number of rounds (without changing x).

Table 1 clearly shows that the false-acceptance ratio is remarkably higher in the Hancke and Kuhn protocol. The basic reason is that an attacker has a $\frac{3}{4}$ probability of guessing a response correctly in the Hancke and Kuhn protocol, but only a $\frac{1}{2}$ probability in our noise resilient MAD protocol. This difference is amplified exponentially, and not entirely compensated by the fact that an attacker has to guess more bits correctly in the Hancke and Kuhn protocol ($(n - x)$ bits, compared to k bits in our noise resilient MAD protocol). This property is also demonstrated in table 3. In this numerical example, the number of rounds n is 63, and the bit error rate P_b is 0.02. The information on which error correcting code to use in our noise resilient MAD protocol, is based on [19].

Note that the difference in false-acceptance ratio is so large, that even allowing a slightly lower number of errors x in the Hancke and Kuhn protocol does not really help to remove this inequality (e.g., if we have a look at table 3: $P_{FA}(Hancke, x = 1) = 2.9599 \cdot 10^{-7} > P_{FA}(MAD, x = 7) = 3.7253 \cdot 10^{-9}$). One could also fix the number of allowed errors x, but perform more bit fast bit exchanges in the Hancke and Kuhn protocol (or in other words, increase the

Table 3. Comparison of the false-acceptance ratio

Hancke and Kuhn protocol	
# allowed errors x	P_{FA}
$x = 13$	0.2611
$x = 10$	0.0584
$x = 7$	0.0052
$x = 5$	$5.1111 \cdot 10^{-4}$
$x = 3$	$2.3004 \cdot 10^{-5}$
$x = 1$	$2.9599 \cdot 10^{-7}$

Noise resilient MAD protocol	
# allowed errors x	P_{FA}
$(63, 12)$ ECC, $x = 13$	$2.4414 \cdot 10^{-4}$
$(63, 18)$ ECC, $x = 10$	$3.8147 \cdot 10^{-6}$
$(63, 28)$ ECC, $x = 7$	$3.7253 \cdot 10^{-9}$
$(63, 37)$ ECC, $x = 5$	$7.2760 \cdot 10^{-12}$
$(63, 47)$ ECC, $x = 3$	$7.1054 \cdot 10^{-15}$
$(63, 57)$ ECC, $x = 1$	$6.9389 \cdot 10^{-18}$

number of rounds n). This would however make the distance bounding protocol more expensive, as the cost is directly related to the number of fast bit exchanges n. Fig. 5 shows the relation between the false-acceptance ratio and the number of rounds n, for a fixed number of allowed errors x. In this example, we fixed the number of allowed errors to 5, the bit error rate P_b is 0.005, and the information on which error correcting code to use (in our noise resilient MAD protocol) is based on [19]. Fig. 5 clearly shows that the Hancke and Kuhn protocol needs significantly more rounds n to obtain the same false-acceptance ratio. This largely increases the cost, and also causes the false-rejection ratio to rise several orders of magnitude.

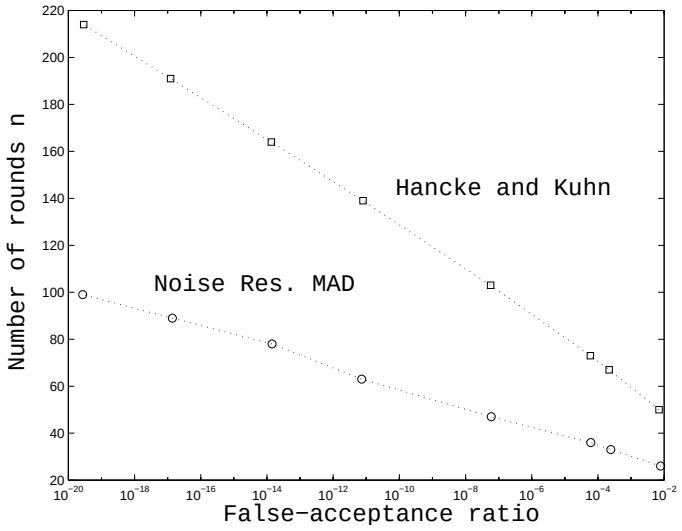

Fig. 5. Relation between the number of rounds n and the false-acceptance ratio

7 Conclusion

Location information can be used to enhance mutual entity authentication protocols in wireless ad-hoc networks. Distance bounding protocols, which have been introduced by Brands and Chaum at Eurocrypt'93 to inhibit distance fraud and mafia fraud attacks, can be employed in proximity based authentication schemes to determine an upper-bound on the distance to another entity. Hancke and Kuhn have extended these protocols to cope with noisy channels, which is important in mobile environments.

In this paper, we have extended the mutual authentication distance bounding (MAD) protocol of Čapkun et al. to make it tolerant to bit errors. This is accomplished by employing binary codes to correct bit errors occurring during the fast bit exchanges, the main building block of the distance bounding protocol. The protocol is best used for radio frequency communications, which is more suited for secure applications than ultrasonic. Our noise resilient MAD protocol requires significantly less number of communication rounds to obtain the same false-acceptance ratio as the Hancke and Kuhn protocol. It also provides mutual authentication and can be made robust to terrorist fraud attacks by executing the protocol in trusted hardware. Compared to the Hancke and Kuhn protocol, our noise resilient MAD protocol requires slightly more bits to be exchanged on the slower communication channel. Fortunately the cost is primarily related to the number of rounds executed on the fast communication channel (because of the technical restrictions). Our distance bounding protocol is perfectly suitable to be employed in noisy wireless environments such as RFID.

Acknowledgments

Dave Singelée is funded by a research grant of the Katholieke Universiteit Leuven (K.U.Leuven). This work was also supported by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government. The authors would also like to thank Markus G. Kuhn for the interesting discussion on distance bounding protocols, his comments were very valuable.

References

1. Bardram, J., Kjær, R., Pedersen, M.: Context-Aware User Authentication - Supporting Proximity-Based Login in Pervasive Computing. In: Proceedings of the 5th International Conference on Ubiquitous Computing (Ubicomp 2003). Lecture Notes in Computer Science, LNCS 2864, Springer-Verlag (2003) 107–123
2. Denning, D., MacDoran, P.: Location–Based Authentication: Grounding Cyberspace for better Security. In: Computer Fraud and Security, Elsevier (1996) 12–16
3. Bahl, P., Padmanabhan, V.: RADAR: An In-Building RF–based User Location and Tracking System. In: Proceedings of the 19th annual conference on Computer Communications (INFOCOM '00). Volume 2., IEEE (2000) 775–784

4. Humphrey Cheung: The Bluesniper Rifle. http://www.tomsnetworking.com/Sections-article106.php (2004)
5. DEFCON: Computer Underground Hackers Convention. http://www.defcon.org (2004)
6. Brands, S., Chaum, D.: Distance–Bounding Protocols. In: Advances in Cryptology - EUROCRYPT '93. Lecture Notes in Computer Science, LNCS 765, Springer-Verlag (1994) 344–359
7. Sastry, N., Shankar, U., Wagner, D.: Secure Verification of Location Claims. www.cs.berkeley.edu/~nks/locprove/csd-03-1245.pdf (2003)
8. Kindberg, T., Zhang, K.: Validating and Securing Spontaneous Associations between Wireless Devices. In: Proceedings of the 6th Information Security Conference (ISC '03). Lecture Notes in Computer Science, LNCS 2851, Springer-Verlag (2003) 44–53
9. Bussard, L.: Trust Establishment Protocols for Communicating Devices. PhD thesis, ENST Paris (2004) 233 pages.
10. Desmedt, Y.: Major Security Problems with the "Unforgeable" (Feige)–Fiat–Shamir Proofs of Identity and how to overcome them. In: Proceedings of SecuriCom '88. (1988) 15–17
11. Singelée, D., Preneel, B.: Location Verification using Secure Distance Bounding Protocols. In: Proceedings of the 2nd IEEE International Conference on Mobile, Ad Hoc and Sensor Systems (MASS '05). (2005) 7 pages.
12. Waters, B., Felten, E.: Proving the Location of Tamper–Resistant Devices. http://www.cs.princeton.edu/bwaters/research/location_proving.ps (2003)
13. Clulow, J., Hancke, G., Kuhn, M., Moore, T.: So Near and Yet So Far: Distance-Bounding Attacks in Wireless Networks. In: Proceedings of the 1st European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS '06). Lecture Notes in Computer Science, LNCS 4357, Springer-Verlag (2006) 83–97
14. Čapkun, S., Buttyán, L., Hubaux, J.: SECTOR: Secure Tracking of Node Encounters in Multi–hop Wireless Networks. In: Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03). (2003) 21–32
15. Hancke, G., Kuhn, M.: An RFID Distance Bounding Protocol. In: Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM '05), IEEE Computer Society (2005) 67–73
16. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1996)
17. Munilla, J., Ortiz, A., Peinado, A.: Distance Bounding Protocols with void-challenges for RFID. Workshop on RFID Security – RFIDSec '06 (2006)
18. Jaffe, D.: Information about binary linear codes. http://www.math.unl.edu/~djaffe2/codes/webcodes/codeform.html
19. Litsyn, S.: Table of Nonlinear Binary Codes. http://www.eng.tau.ac.il/~litsyn/tableand/index.html
20. MacWilliams, F., Sloane, N.: The Theory of Error-Correcting Codes. North-Holland (1977)
21. Pless, V., Brualdi, R., Huffman, W.: Handbook of Coding Theory. Elsevier Science Inc. (1998)