Distance Bounding Protocols on TH-UWB Radios

Ahmed Benfarah *,†, Benoit Miscopine *, Jean-Marie Gorce †, Cédric Lauradoux † and Bernard Roux †
* Orange LABS, F-38240 MEYLAN
† Université de Lyon, INRIA, INSA-Lyon, CITI laboratory
‡ Université de Lyon, CNRS, INSA-Lyon, Institut Camille Jordan

Abstract—Relay attacks pose a real threat to the security of wireless communications. Distance bounding protocols have been designed to thwart these attacks. In this paper, we study the way to adapt distance bounding protocols to time-hopping ultra wide band (TH-UWB) radios. Two protocols are proposed which are based on the milestones of the TH-UWB radio: the time-hopping sequence and the mapping code. The security and the different merits of those protocols are analyzed.

Index Terms—TH-UWB, security, authentication, distance bounding.

I. INTRODUCTION

The development of wireless technology has increased the threat of the relay attacks. This class of attacks has been introduced by Desmedt et al. [1] to defeat the authentication protocols. Recently, these attacks, which include the mafia fraud, the terrorist fraud and the distance fraud, have been demonstrated as possible on RFID systems [2] and wireless networks [3]. Moreover, the relay attacks can be used as a mean to implement more advanced attack such as wormhole in wireless sensor networks (WSNs) [4].

One of the first solution to this threat are distance bounding protocols as called by Brand and Chaum [5]. Let consider a two-party communication protocol between Alice and Bob over an insecure channel. The goal of a distance bounding protocol is for Alice to ensure that Bob is inside her neighborhood at given time t. Such a protocol has an authentication side and a distance bounding side (the distance between Alice and Bob is upper-bounded).

Ultra-wideband (UWB) communication is a promising candidate for the implementation of distance bounding protocols [6], [7]. An UWB system offer a fine resolution when it comes to evaluate distance or location [8], [9]. The communications between the transmitter and the receiver require in an UWB system precise synchronization which can be exploited for distance or position measurements. An accurate synchronization is critical for the design of distance bounding protocols based on the round trip time (RTT).

In this paper, we adapt distance bounding protocols to the context of time-hopping UWB radios. The main characteristics of a time-hopping UWB radio are the time-hopping sequence S defined for spreading the spectrum and the mapping code C defined to increase the resiliency to noise. In our first protocol, the time-hopping sequences used during the communication are unknown to the attacker. In the second protocol, the mapping code is unknown to the adversary.

II. PRELIMINARIES

A. Time-Hopping UWB

UWB is a wireless technology which consists in the emission of very short temporal pulses. In UWB systems, time-hopping (TH) is a means to smooth the radiated spectrum, so as to optimize the transmission power. It also provides a way to isolate concurrent links, which can be used to share the communication medium [10]. In TH-UWB, the time unit is the frame of duration T_f, which is composed of N_s time slots. The time-hopping code S is a sequence of integers over Z/N_sZ that determines which time slot is occupied by the pulse in each frame. In Fig. 1, we have N_s = 3. To increase immunity against noise, N_f pulses are transmitted per symbol: the duration T_s of a TH-UWB symbol is T_s = N_f.T_f. Different binary modulations are possible for a TH-UWB system, in this paper, PPM or BPSK modulations are used [11]. The mapping code C is a binary sequence of length N_f that determines the N_f modulated pulses for the binary symbol 0 and 1. A repetition mapping code is frequently used, i.e. the N_f pulses are the same [12]. However, other mapping codes are possible such as the one proposed in [13] (N_f = 4):

\[ C = \begin{cases} 0, 1, 0, 1 & \text{if the symbol is equal to 0,} \\ 1, 0, 1, 0 & \text{if the symbol is equal to 1.} \end{cases} \]

Fig. 1. Structure of a TH-UWB symbol with N_s = 3, N_f = 4, PPM modulation and repetition code

Prior to data transmission, a precise synchronization should be acquired between the transmitter and the receiver for detecting the short pulses. Synchronization is acquired thanks to a preamble composed of unmodulated symbols with predefined time-hopping code known to the receiver [14]. The
latter compares temporal distances between the received pulses and those predicted by the predefined time-hopping code. Synchronization is declared when the TH sequence is fully identified [15]. Synchronization is a critical phase in UWB systems: the number of required pulses to acquire synchronization is independent of the payload size. This implies that the energetic cost of the synchronization becomes preponderant in the consumption required for receiving a packet, when dealing with short packets.

B. Distance Bounding protocols

Distance Bounding protocols have been introduced by Brand and Chaum [5] to defeat certain classes of man in the middle attack (MITM) described by Desmedt et al. [1] and known as mafia frauds. A distance bounding protocol allows a verifier V to check that a legitimate user, the prover P, is within its neighborhood, i.e. the Euclidean distance between the verifier and the prover is upper bounded.

A distance bounding protocol is said to be secure if the verifier rejects a prover with overwhelming probability when the prover is not legitimate and/or it is outside of the neighborhood. The verifier accepts the prover if the latter is legitimate and within the neighborhood. Many solutions are available to measure the distance between two radio devices: GPS, RTT, RSSI, AoA, ... The reader can consult [16] for more details on these techniques. For low cost embedded devices, the RTT is the most popular solution, and the UWB technology is so far the most promising radio for the implementation of distance bounding protocols based on the measure of the RTT (see [6], [7], [4]): UWB provides a very accurate synchronization between the verifier and the prover.

Two families of RTT based distance bounding protocols have been proposed in the literature: Brand and Chaum [5], and Hancke and Khun [6] protocols. The Hancke and Khun distance bounding protocol is composed of two steps: (1) an initialization phase in which the prover and the verifier exchange values to compute a shared state, and (2) a fast phase during which the RTT is measured several times and the authentication process is carried out. Then, the verifier can conclude if the legitimate user is inside its neighborhood. In comparison with Hancke and Khun, Brand and Chaum protocol requires an additional phase because the measure of the RTT and the authentication are two independent processes. The core of our protocols is based on the work of Hancke and Khun [6].

In this paper, the verifier and the prover are two UWB devices with identical capabilities. The context of applications is for instance the verification of the neighborhood of a node in a WSN.

III. NEW DISTANCE BOUNDING PROTOCOLS

We propose two alternatives to introduce distance bounding in TH-UWB radios. In the protocol A, the TH codes used by the verifier and the prover (respectively $S^V$ and $S^P$) are unknown to the adversary while the mapping code $C$ is public. In the protocol B, the mapping codes of the verifier and of the prover are unknown to the adversary while the TH code is public.

A. Protocol A: secret TH sequences

The protocol is described in Fig. 2 and detailed here. We assume that synchronization is performed between the verifier $V$ and the legitimate prover $P$ before the distance bounding protocol begins. The synchronization is maintained during the protocol execution. This hypothesis is realistic in the sight of the number of bits exchanged during the protocol. Moreover, the mapping code used by the verifier and the prover is public. It can be for instance a simple repetition code.

\[
\begin{array}{ll}
\text{Prover } P & k \in \{0,1\}^m, f \\
\text{Verifier } V & k \in \{0,1\}^m, f \\
\text{Picks } N_P & \text{Picks } N_V \\
\text{Picks } z \in \{0,1\}^n & \text{Picks } c \in \{0,1\}^n \\
\text{For } i = 1 \cdots n & \text{Sends } c_i \text{ with } S^V_i \\
\text{If correct time slot then} & \\
\text{else} & \\
r_i = R^e_i & \\
t = S^P_i & \\
\text{Sends } r_i \text{ with } t & \text{Measure of RTT: } \delta t_i \\
\end{array}
\]

Fig. 2. Protocol A: secret TH sequences.

1) Protocol requirements: $P$ and $V$ share a secret key $k \in \{0,1\}^m$. They can both compute a pseudo-random function $f$ and they have an access to a random number generator. The pseudo-random function can be implemented with a cryptographic hash function such as SHA-256.

2) Initialization phase: The protocol begins as follow: the prover picks a nonce $N_P$ (number used once) and sends it to $V$. Reciprocally, the verifier picks a nonce $N_V$ and sends it to $P$. From the values $N_V, N_P$ and the key $k$, $V$ and $P$ compute a share state $H = f(k, N_P, N_V)$. $H$ is a bit string of length $2n(pN_f + 1)$ where $n$ is the number of rounds in the fast phase and $p = \log_2 N_s$. For an ease of implementation, we assume that $N_s$ the number of slots is a power of two. $H$ is split into four parts:

- The time-hopping code $S^V$ of the verifier is defined by:
  \[
  S^V = H_1 \cdots H_{p+1}, H_{p+1} \cdots H_{2p} \cdots , H_{N_f}, H_{N_f} \cdots H_{nN_f}.
  \]

  $S^V_i$ of length $(pN_f)$ bits defines the integers over $\mathbb{Z}/N_s \mathbb{Z}$ corresponding to time slots used to emit a $i$-th symbol, $i \in \{1, \cdots, n\}$.

- The time-hopping code of the prover $S^P$ is defined by:
  \[
  S^P = H_{N_f+1} \cdots H_{nN_f+1}, H_{nN_f+1} \cdots H_{nN_f}.
  \]
- A first register $R^0 = H_{2(n.N_f .p)+1} \cdots H_{n(2.N_f.p+1)}$ containing $n$ bits.
- A second register $R^1 = H_{n(2.N_f.p+1)+1} \cdots H_{2n(2.N_f.p+1)}$ which contains also $n$ bits.

In addition, the verifier and the prover pick respectively an $n$-bit random vector $r$ and $z$. The prover also picks randomly an $n.p.N_f$-bit vector $q$. This vector is decomposed into sequences $q_i$ of $(p.N_f)$ bits as $S^V$ and $S^P$.

3) **Fast phase:** The fast phase consists in $n$ rounds in which the verifier sends a challenge bit $c_i$ to $P$. This challenge is transmitted using $N_f$ pulses according to the public mapping code. Each pulse is in the time slot defined by $S^V_i$. The prover replies with $r_i = R^c_i$ the $i^{th}$ bit of the register $R^c$, with the time hopping sequence $S^P_i$, if the challenge $c_i$ is received in the correct time slots. Otherwise, the prover detects an attack and replies randomly with $z_i$ from the register $z$ with the TH sequence $q_i$ from the vector $q$. Reciprocally, the verifier also assumes an attack if it receives an impulse in the wrong time slot and stops the protocol. Unless an attack is detected, the verifier computes in each round the RTT, denoted $\delta t_i$, between the emission of the last impulse and the reception of the first impulse. The RTT includes the propagation time $t_p$, the processing delay of the prover $t_d$ and the time-hopping sequence $S^P_i$ by the equation:

$$\delta t_i = 2 \cdot t_p + t_d + S^P_i \cdot T_c. \quad (1)$$

4) **Verification:** The protocol succeeds if all the responses $r_i$ sent by the prover are correct and $\forall i, \delta t_i \leq t_{max}$ where $t_{max}$ is an upper-bound.

**B. Protocol B: secret mapping code**

In the protocol B, the verifier and the prover are coding and decoding with a side information. The prover and the verifier use a public codebook. The codebook consists in the cosets of an $(N_f, 1, d)$ repetition code. Let consider the $(4, 1, 4)$ repetition code. The $2^{N_f-1}=3$ cosets of this code are $K_1 = \{0000, 1111\}$, $K_2 = \{0001, 1110\}$, $K_3 = \{0010, 1101\}$, $K_4 = \{0111, 1100\}$, $K_5 = \{0100, 1011\}$, $K_6 = \{0110, 1001\}$, $K_7 = \{0111, 1000\}$. The Hamming distance between codewords of each coset is $d = 4$. For each bit transmitted during the protocol, a coset is chosen to define how the 0 and the 1 are encoded. For instance, 0 can be coded with minimum of the coset.

The requirements of the protocol B are the same than the protocol A with the addition of the codebook.

1) **Initialization phase:** The prover and the verifier exchange the nonce $N_P$ and $N_V$. Then, they both compute the share state $H = f(k, N_P, N_V)$ of length $2.N_f.n$ bits. Let consider $N_f = 4$, the length of $H$ is $8n$ and the $H$ value is split into four:

- The register $C^V$ of length $3n$ defines the sequence $C^V_i$ of cosets used during the protocol by the verifier:
  \[ C^V = H_1, H_2, H_3, H_4, H_5, H_6, \cdots H_{3n-2}, H_{3n-1}, H_{3n} \]

- In the same vain, the register $C^P$ of length $3n$ defines the sequence $C^P_i$ of cosets used by the prover:
  \[ C^P = H_{3n+1}, H_{3n+2}, H_{3n+3}, \cdots H_{6n-2}, H_{6n-1}, H_{6n} \]

- A register $R^0 = H_{6n+1} \cdots H_{7n}$ containing $n$ bits.
- A register $R^1 = H_{7n+1} \cdots H_{8n}$ which contains $n$ bits.

In addition, the verifier and the prover pick respectively an $n$-bit random vector $r$ and an $3n$-bit vector $q$. This vector $q$ has the same purpose than $C^V$ and $C^P$. It is composed of symbols of 3 bits.

2) **Fast phase:** At round $i$, $1 \leq i \leq n$, the verifier sends a challenge bit $c_i$ coded with a word from the coset $C^V_i$ to $P$. The prover checks that the challenge corresponds to a word from the coset $C^V_i$: the prover computes the Hamming distance between the mapping code of received challenge and the two words from the coset $C^V_i$. If this Hamming distance $\leq \Delta$, then the prover responds with $r_i = R^c_i$ coded from the coset $C^P_i$. $\Delta$ must be chosen such that:

$$\Delta \leq \left[ \frac{d-1}{2} \right]. \quad (2)$$

So, in our example $\Delta \leq 1$. The interest of $\Delta$ is to make a tradeoff between security and resiliency to noise. Taking $\Delta = 0$ is more beneficial to security while taking $\Delta$ such as in (2) with equality is more beneficial to error correction. If the Hamming distance condition is not satisfied, the prover detects an attack and responds with a random mapping from the vector $q$. The verifier computes the RTT in each round such (1).

3) **Verification:** The protocol succeeds if all the responses $r_i$ are distant at most with $\Delta$ from the codeword of coset $C^P_i$ and $\forall i, \delta t_i \leq t_{max}$ where $t_{max}$ is the upper-bound.

**IV. SECURITY AND PRACTICAL CONSIDERATIONS**

In this section, the security of our two protocols is considered against the mafia fraud as done in [17]. Two cases are considered: an idealistic case without noise and a more practical case in which errors occur. We discuss also the energy consumption of the two protocols.

We are interested by the resiliency of our protocols against two types of adversary strategies. Both have to be considered when analyzing a protocol based on Hancke and Kuhn [6]:

- **No-ask strategy:** this is the classical impersonation attack. The adversary attempts to answer the challenges of the verifier by itself only. It can not be strictly considered as a form of mafia fraud. However, it is critical to know the security of our protocols against this attack.
- **Pre-ask strategy:** the adversary tries to gain some information by querying the prover before the beginning of the fast phase.

A. Analysis with noise-free communication

1) **Protocol A:** In the protocol A, the radio interface of the prover and of the verifier is always turned on for detecting
an attack. We compute the probability of succeeding an attack against the protocol A with the previous strategies.

**No-asking strategy** - To succeed the no-ask strategy, the adversary needs to emit the pulses in the correct time slots $s_i^P$ and should also guess the correct answer $r_i$ for each round $i$. The probability of choosing the true time slots in a round is $(1/N_a)^{N_f}$. Let define $x = N_a^{-1}$. The adversary’s success probability with the no-ask strategy $P_{na,A}$ against protocol A is given by:

$$P_{na,A} = \left(\frac{1}{2x}\right)^n.$$  \hspace{1cm} (3)

**Pre-asking strategy** - During the attack, the adversary queries the prover with challenges $\hat{c}_i$ emitted in time slots chosen randomly, and it obtains the answers $r_i$ from $P$. Then, the adversary succeeds if $\hat{r}_i = r_i$ in the same time slots as received. To compute the adversary’s probability of success with the pre-asking strategy, we define the following events:

- $A_i$ the event that $\hat{c}_i = c_i$ in the $i^{th}$ round,
- $B_i$ the event that $\hat{c}_i$ is emitted in time slots $s_i^V$ in the $i^{th}$ round,
- $C_i$ the event that $r_i$ is emitted in time slots $s_i^P$ in the $i^{th}$ round,
- $D_i$ the event that $r_i = R_i^c$ in the $i^{th}$ round.

The attack succeeds in the $i^{th}$ round if the event $(C_i$ and $D_i$) is realized. We compute the probability of this event:

$$P(C_i) = P(C_i|B_i) \cdot P(B_i) + P(C_i|\bar{B_i}) \cdot P(\bar{B_i}),$$

$$= 1 \cdot \frac{1}{x} + \frac{1}{x} \cdot \left(1 - \frac{1}{x}\right).$$

$$P(D_i) = P(D_i|(A_i$ and $B_i)) \cdot P(A_i$ and $B_i) + P(D_i|(\bar{A_i}$ or $\bar{B_i})) \cdot P(\bar{A_i}$ or $\bar{B_i}),$$

$$= 1 \cdot \frac{1}{2} \cdot \frac{1}{x} + \frac{1}{2} \cdot \left(1 - \frac{1}{x}\right).$$

Thus, the success probability of the pre-asking strategy $P_{pa,A}$ against protocol A is:

$$P_{pa,A} = \left(\frac{4x^2 - 1}{4x^3}\right)^n.$$  \hspace{1cm} (4)

**Comparing strategies** - From Equations (3) and (4), one can see that the best strategy for the adversary is the pre-asking. Table I compares the success probability of the pre-asking strategy between protocol A, Hancke and Kuhn [6] (HK) and MUSE-pHK described in [18].

<table>
<thead>
<tr>
<th>Protocol</th>
<th>$P_{pa}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HK</td>
<td>$(3/4)^n$</td>
</tr>
<tr>
<td>MUSE-pHK</td>
<td>$(2p - 1)/p^2)^n$</td>
</tr>
<tr>
<td>A</td>
<td>$((4x^2 - 1)/(4x^3))$</td>
</tr>
</tbody>
</table>

**TABLE I**  
SECURITY COMPARISON OF DISTANCE BOUNDING PROTOCOLS.

Protocol A improves the HK protocol and outperforms one of the HK enhancement, i.e. the MUSE-pHK protocol, when we consider $N_a$ the number of slots in protocol A equivalent to $p$ the number of states in MUSE-pHK. Our protocol is still better when $N_f = 1$. In Fig. 3, we represent the adversary’s probability of success of the three protocols with $N_a = p = 8$ (practical value for UWB systems) in the case of $N_f = 1$ (to have a fair comparison). For a given security level, protocol A divides by approximately $2/3$ the number of required rounds needed by MUSE-8HK.

![Fig. 3. Adversary success probability against protocol A with $N_a = 8$, $N_f = 1$.](image)

2) **Protocol B**: We provide the security analysis of protocol B with an arbitrary value $\Delta$ accepted by both the verifier and the prover. Then, we adapt the obtained formula to our particular context with no noise, i.e. $\Delta = 0$.

**No-asking strategy** - Let define:

- $E_i$ the event that the mapping code of $\hat{r}_i$ emitted by the adversary is at Hamming distance less than $\Delta$ from the mapping code of the correct answer $R^c_i$.

The probability that the adversary succeeds a no-asking attack at round $i$ corresponds to the probability of event $E_i$. Let define $z = \sum_{j=0}^{\Delta} \binom{N_f}{j}/2^{N_f}$. Thus, the probability of succeeding the no-asking attack $P_{na,B}$ is given by:

$$P_{na,B} = (z)^n.$$  \hspace{1cm} (5)

**Pre-asking strategy** - The adversary queries the prover with a challenge $c_i$ encoded randomly from all the combinations. Then, the adversary obtains the encoded form of $r_i$ from the prover. Here, the time-hopping sequences are predefined, so the adversary knows in which time slots it should transmit the pulses.

In addition, we define:

- $G_i$ the event that the Hamming distance between the mapping code of $\hat{c}_i$ and one of the two words of coset $C^V_i$ is less than $\Delta$.

The adversary succeeds its attack at round $i$ if the event...
(E_i) is realized. We compute the probability of this event:
\[ P(E_i) = P(E_i | G_i) \cdot P(G_i) + P(E_i | G_i^c) \cdot P(G_i^c) \]
\[ = \frac{3}{4} \cdot P(G_i) + \frac{\sum_{i=0}^{\Delta} (N_f^i)}{2^{N_f}} \cdot (1 - P(G_i)). \]

To determine the probability of success at round i, \( P(G_i) \) must be computed. The choice of \( \Delta \) such (2) gives that any binary word of length \( N_f \) can not be at an Hamming distance less or equal than \( \Delta \) from the two words of any coset simultaneously. So, \( P(G_i) = 2z \). In conclusion, the adversary’s probability of success with the pre-asking strategy \( P_{pa,B} \) is given by:
\[ P_{pa,B} = \left( z \left( \frac{5}{2} - 2z \right) \right)^n. \] (6)

### Comparing protocols
Again, the best strategy is the pre-asking. The previous equation is valid for any \( \Delta \), but in the case of noise-free communication we take \( \Delta = 0 \). In Fig. 4, we compare the probabilities of success of the pre-asking strategy for protocols A, B and MUSE-pHK with parameters \( N_f = 4 \), \( N_c = 2 \), \( p = 2^{N_f} = 16 \) and \( \Delta = 0 \). Clearly, protocol A insures a higher security level than protocol B. We will see in IV-C the counterpart of this high level security. Protocol B guarantees a security level comparable to MUSE-16HK with less memory consumption. Indeed, with MUSE-16HK we need 64 \( n \) bits memory [18] and we need only 11 \( n \) bits memory with protocol B.

![Adversary success probability against protocol B with \( N_f = 4 \). \( \Delta = 0 \).](image)

Fig. 4: Adversary success probability against protocol B with \( N_f = 4 \), \( \Delta = 0 \).

### B. Analysis with noisy communication

In practical systems, bits exchanged between V and P may be erroneous due to noise. This have an impact on the security level of distance bounding protocols because some errors must be tolerated, increasing the success probability of the adversary.

1) Protocol A: V tolerates \( (\ell - 1) \) errors during the verification phase. If more errors are noticed, the protocol fails.

Noise is modeled by a symmetric binary channel with the parameter \( \varepsilon_{bdec} \) for the bit error probability of the links \( V \rightarrow P \) and \( P \rightarrow V \) before channel decoding. We consider the same error probability for the two links since \( V \) and \( P \) have same capabilities. The mapping code provides a coding gain. We give an upper-bound on the bit error probability after channel decoding \( \varepsilon_{dec} [19] \):
\[ \varepsilon_{dec} \leq \frac{N_f}{j} \varepsilon_{bdec} (1 - \varepsilon_{bdec})^{N_f - j}, \] (7)

where \( t = \left[ \frac{N_f - 1}{2} \right] \) is the capacity of error correction of the repetition code.

In absence of an attacker, the verifier \( V \) may reject a legitimate prover \( P \) if more \( \ell \) responses are affected by noise. This false-reject probability \( P_{FR} \) depends on the events:
- \( I_i \) the event that \( c_i \) is correctly decoded,
- \( I_i \) the event that received \( r_i \neq R_i^{c_i} \).

\[ P(I_i) = P(I_i | D_i) \cdot P(D_i) + P(I_i | \tilde{D}_i) \cdot P(\tilde{D}_i) \]

\[ P(D_i) = P(D_i | H_i) \cdot P(H_i) + P(D_i | \tilde{H}_i) \cdot P(\tilde{H}_i) \]

After computation, we find:
\[ P(I_i) = \varepsilon_{dec} \cdot \left( \frac{3}{2} - \varepsilon_{dec} \right). \]

We consider that \( \varepsilon = \varepsilon_{dec} \cdot (3/2 - \varepsilon_{dec}) \). The probability of false-reject \( P_{FR,A} \) is given by:
\[ P_{FR,A} = \sum_{j=\ell}^{n} \binom{n}{j} \varepsilon^j (1 - \varepsilon)^{n-j}. \] (8)

The fact that \( V \) tolerates some errors changes the adversary’s probability of success computed in IV-A1. The attacker needs to succeed in only \((n - j)\) times, \( 0 \leq j \leq \ell - 1 \) and the event \((\tilde{D}_i \text{ and } C_i)\) must be realized \( j \) times. So, the new probability of success called here the probability of false-accept \( P_{FA,A} \) is given by:
\[ P_{FA,A} = \sum_{j=0}^{\ell-1} \binom{n}{j} \cdot \frac{(4x^2 - 1)}{4x^3} \cdot \left( \frac{(2x - 1)^2}{4x^3} \right)^j. \] (9)

Here, we have assumed that the link \( V \rightarrow P \) is noisy while the link involving the adversary is perfect which corresponds to the worst case. In Fig. 5, we represent the probability of success without noise and \( P_{FA,A} \) the probability of success in noisy channel with parameters \( N_s = 4 \), \( N_f = 4 \) and the number of tolerated errors \( \ell - 1 = \left[ \varepsilon.n \right] \) such that \( \varepsilon_{bdec} = 10^{-1} \) and \( \varepsilon_{dec} \) is given by the bound (7) taken with equality. It is clear from Fig. 5 that noise does not affect a lot the security performance of protocol A with the parameters mentioned.

2) Protocol B: In noisy communication, \( V \) should tolerate \( (\ell - 1) \) erroneous responses and \( \Delta \) is ever chosen such (2). We propose to compute the changed adversary’s probability of success in presence of noise. In this case, the adversary must succeed in only \((n - j)\) rounds, \( 0 \leq j \leq \ell - 1 \) and the event \( \tilde{E}_i \) must be realized in \( j \) rounds. The event \( \tilde{E}_i \) is defined by:
- \( \tilde{E}_i \) the event that the mapping code of \( r_i \) is at Hamming distance \( \leq \Delta \) from the wrong codeword of the coset \( C_i^{R_i} \).
The probability of false-accept $P_{FA,B}$ is given by:

$$P_{FA,B} = \sum_{j=0}^{\ell-1} \binom{n}{j} \cdot X^{n-j} \cdot Y^j,$$

where $X$ is the probability of success in a round already computed in IV-A2 and $Y = P(E_i)$ given by:

$$Y = z \cdot \left(\frac{3}{2} - 2z\right).$$

We show in Fig. 5 the probability of success in noise-free channel and the probability of success in noisy channel $P_{FA,B}$ with parameters $N_t = 8$, $\Delta = 2$ and $\varepsilon_{\text{dec}} = 10^{-3}$. The number of tolerated errors $(\ell - 1)$ is taken like previously, but we should point out that $\varepsilon_{\text{dec}}$ is given with (7) where $t = \Delta$ here. In this case the capacity of error correction is $\Delta$. The loss in security performance caused by noise is much more important in protocol B than in protocol A. In fact, for a security level of $10^{-7}$, the number of required rounds $n$ is 4 for the noise-free case and 16 for the noisy case.

**REFERENCES**


