Attacks on Singelée and Preneel's protocol

Jorge Munilla and Alberto Peinado

Dpt. Ingeniería de Comunicaciones. E.T.S.I.Telecomunicación.
University of Málaga, Spain

Abstract. Singelée and Preneel have recently proposed a enhancement of Hancke and Kuhn's distance bounding protocol for RFID. The authors claim that their protocol offers substantial reductions in the number of rounds, though preserving its advantages: suitable to be employed in noisy wireless environments, and requiring so few resources to run that it can be implemented on a low-cost device. Subsequently, the same authors have also proposed it as an efficient key establishment protocol in wireless personal area networks. Nevertheless, in this paper we show effective relay attacks on this protocol, which dramatically increase the success probability of an adversary. As a result, the effectiveness of Singelée and Preneel's protocol is seriously questioned.

1 Introduction

Contactless smart cards are nowadays more and more used in applications which require security, like payment or access-control applications [1]. Although many solutions have been proposed to secure these RFID (radio frequency identification) systems, most of them are still vulnerable to relay attacks. This attack is conceptually depicted in Fig.1. It is a kind of man-in-the-middle attack where the genuine reader interacts with a rogue card, that manages to fool the reader into thinking that it is directly communicating with the genuine card [2, 3]. For example, to open a vehicle, an adversary with a rogue card placed near the vehicle, establishes contact with the legitimate reader, while an accomplice with a rogue reader, placed near the owner, powers up his card. Then both rogue parties readily forward each other all the messages. The electronic protection is thus breached, and both genuine parties, reader and card, remain unaware.

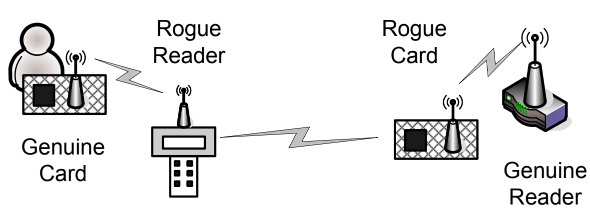

Fig. 1. Sketch of a relay attack

Relay attacks require simpler technical resources than tampering or cryptanalysis and they cannot be prevented by ordinary security protocols that operate in the high layers of the protocol stack. The main countermeasure against them is the use of so called distance bounding protocols, which are tightly integrated into the physical layer. Such protocols combine cryptographic and physical properties to determine an upper bound on the distance between the verifier and the prover (generally by measuring the round trip time). This way, they verify not only that the prover knows a cryptographic secret, but also that the prover is within a certain distance.

The most used cards are the "proximity" cards (ISO 14443 [4]), which operate at 13.56 MHz and are passive; they operate without any internal battery and receive the power that they need from the reader. This offers a long lifetime but results in short read ranges (of up to 10cm), and limited processing power. According to these particular characteristics Hancke and Kuhn proposed the first distance bounding protocol specifically designed for RFID devices [5]. This protocol has been used as a reference point by Singelée and Preneel [6] to propose a protocol which tries to outperform it. More precisely, this protocol seeks to reduce the probability of an adversary successfully impersonating a legitimate card; i.e. the false acceptance probability. A modification of this protocol has been also proposed by the same authors as an efficient key establishment protocol in wireless personal area networks [7].

This paper, however, proposes two effective attacks against the protocol of Singelée and Preneel: the first being when it is implemented on RFID devices, and the second for a more general case. It is structured as follows. Section 2 describes comprehensively Hancke and Kuhn's protocol, which we will also refer to as HKP from now on. In Sect. 3 the protocol of Singelée and Preneel is described, which we will also refer to as SPP. Sect. 4 presents the two modified relay attacks, which dramatically increase the adversary's success probability of impersonating a card. Finally, Sect. 5 concludes the paper.

2 Hancke and Kuhn's Protocol

In HKP, the reader uses time measurements of single bit round trips combined with a symmetric-key identification mechanism to authenticate the cards. The reader sends out a challenge and starts a timer; the card receives the challenge, computes the response, and sends it back to the reader, that stops the timer. The reader uses the round trip time, Δt_i, to extract the propagation time and determine the distance between them. In order to reliably extract the propagation time, the processing time must be as short and invariant as possible. Fig.2 depicts the protocol; it starts by having reader and card exchange random nonces, N_r and N_c, that will never be used again. With these nonces and the shared key K, the parties use a hash function to compute an unpredictable string H of length $2n$ bits, and split it into two n-bit strings, v_0 and v_1. Then, a rapid n-round challenge-response phase begins. For the ith round, the ith bit of v_0 is answered if the ith challenge is zero ($C_i = 0$), and the ith bit of v_1 otherwise

($C_i = 1$). The reader checks that the received response is correct and also that it has been received within a certain period of time ($\Delta t_i < t_{max}$).

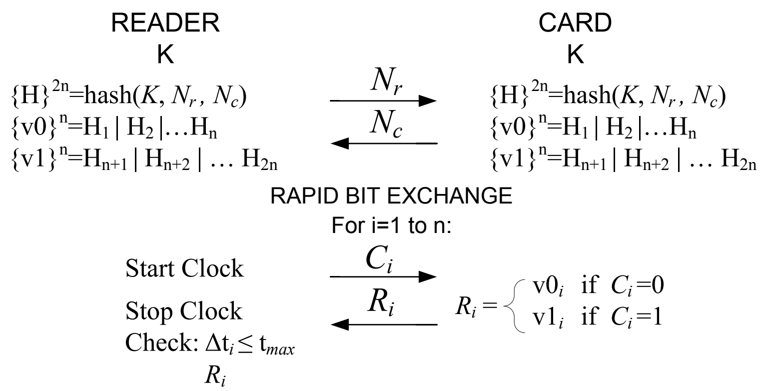

Fig. 2. Hancke and Kuhn's protocol

The probability that an adversary successfully impersonates a legitimate card is $(3/4)^n$. It is $(3/4)^n$, and not $(1/2)^n$, because the adversary can query the card with any value (1 or 0) before receiving the challenge, obtaining the right response for this value. When the actual challenge is sent by the reader, the adversary knows the response whether this challenge coincides with the value that was previously queried. In case it does not coincide, the adversary randomly answers one of the two possible values. This problem is avoided in [8] by sending, at the end of the protocol, a signed message on the challenges and responses received. However, this solution cannot be applied here due to the high probability of bit errors occurring during the rapid bit exchange. This high bit error rate (BER) is mainly caused by two factors. First, the communication method used for the rapid bit exchange is very sensitive to background noise, because an ultra wide band (UWB) link is used as a low-latency channel to achieve a high timing resolution. And second, no detector and corrector mechanisms are used since it would mean additional and variable cycles of processing.

Distance bounding protocols must cope with this problem (high BER). HKP handles communication errors simply by tolerating some bit errors during the rapid bit exchange; i.e. the reader will accept a card as valid even if some, at most x, of the responses are not correct. Unfortunately in the worst case, where no bit is corrupted due to noise, the adversary's success probability increases since he can fail up to x responses,

$$p_{HKP} = \sum_{t=0}^{t=x} \binom{n}{t} \cdot \left(\frac{3}{4}\right)^{n-t} \cdot \left(\frac{1}{4}\right)^{t}. \tag{1}$$

3 Singelée and Preneel's Protocol

3.1 Description of SPP

Singelée and Preneel seek to reduce the probability that an adversary successfully impersonates a legitimate card or, in other words, reduce the number of rounds for the same adversary's success probability. They combine the MAD protocol of Čapkun et al. [9], where both parties authenticate each other and in each round an adversary has only probability 1/2 of replying correctly, with the HKP, which can cope with bit errors during the rapid bit exchange.

This protocol, which is shown in Fig.3, is carried out as follows. Firstly, both parties, Alice and Bob, which share a key K, agree on an $(n,k)ECC$ (Error Correcting Code) capable of correcting at least x bit errors during the rapid bit exchange. The minimal Hamming distance d_{min} of the binary code must be such that $x = 0.5 \cdot (d_{min} - 1)$. For more details about ECC we refer to [10, 11].

Next, Alice and Bob generate k random bits ($r_1, ..., r_k$ and $s_1, ..., s_k$ respectively). These k bits are extended to n-bit strings ($r_1, ..., r_n$ and $s_1, ..., s_n$) by applying the ECC, and a commitment to this string is sent to the other party.

During the n 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 βi to Alice where $\beta_i = s_i \oplus \alpha_i$.

In each round, the time between sending α_i and receiving β_i (or sending β_i and receiving α_{i+1}) is measured to determine an upper bound on the distance between Alice and Bob. After the 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 $s_1, ..., s_k$ and $r_1, ..., r_k$ 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 at the beginning of the protocol. If the MAC and the commitment are correct, the protocol is successful.

Authors point out that their protocol only requires low-cost cryptographic primitives, and hence it is perfectly suitable to be employed in resource constrained wireless networks.

3.2 Performance Analysis (in accordance with the authors)

Since the first k bits of r_i and s_i are independent and uniformly distributed in {0,1}, the two sequences α_i and β_i are independent up to the point where the index is k, and by consequence the first k rounds of the rapid bit exchange are independent. If the commitments sent at the beginning of the protocol is chosen properly [12], it is infeasible for a computationally bounded attacker to determine these bits in advance. The last $(n-k)$ bits of r_i and s_i depend on the first k bits and can be easily computed by applying the $(n,k)ECC$. In the worst case scenario (no bit error occurs), the last $(n-k)$ bits of the sequence α_i and β_i can be computed in advance (from the moment the first k rounds are conducted) and do not offer extra security. To be successful, an adversary hence

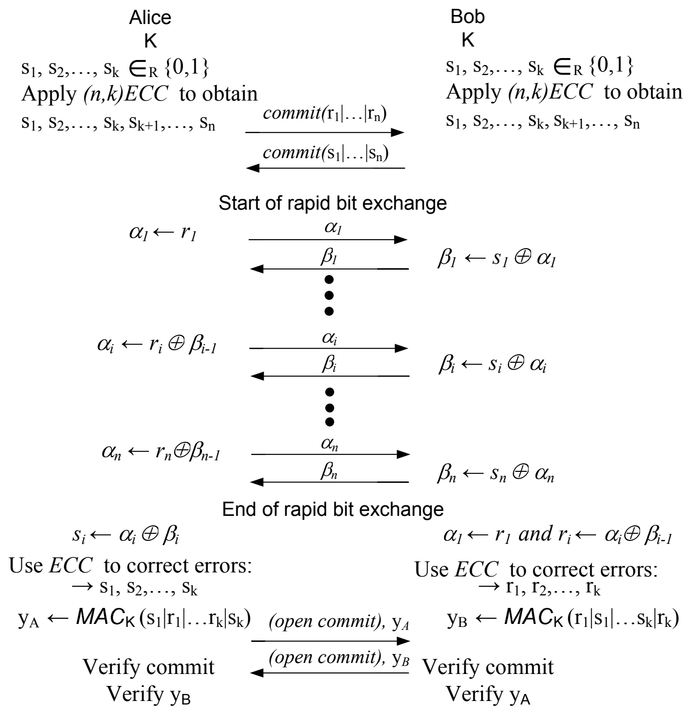

Fig. 3. Singelée and Preneel's protocol

has to correctly guess the first k bits r_i (or s_i). Therefore, the false acceptance probability equals

$$p_{SPP} = \left(\frac{1}{2}\right)^k. \tag{2}$$

4 Attacks on Singelée and Preneel's Protocol

4.1 Attack against RFID Implementation

We firstly analyze here the implementation of SPP on RFID devices. For this case, we have assumed that:

- The reader initiates the protocol (the reader is Alice). As aforementioned, cards are passive and the communication is always initiated by the reader.
- Timing measurements are carried out by the reader (the time between sending α_i and receiving β_i is measured). Cards do not have built-in high precision time base and they generate their internal clocking signal from the carrier frequency of the reader's field.

The key to reduce the false acceptance probability in this SPP with respect to HKP is the mutual authentication. This way, authors assume that an adversary cannot ask the card in advance since he would be detected. Nevertheless, Fig.4 shows an example of a slightly modified relay attack where the adversary asks the card in advance, and makes up to $2x$ failures without being detected. The adversary splits the run of the protocol in three separate phases. He first guesses the challenges α_i until he made x errors. The received responses from the card are to be changed ($\oplus 1$), before relaying them to the reader, when the card has detected an odd number of errors up until that point. From that moment on, in the second phase, he forwards (modifying if necessary) the challenges α_i and guesses the responses β_i until the kth round. The challenges are modified (or not) to be consistent from the point of view of the card, with the errors that it has detected in the first phase. This way, the challenges are changed ($\oplus 1$) when the sum of x (errors detected by the card in the first phase), plus the number of errors detected by the reader up until that point is an odd number. In this second phase, he is again allowed to make x errors. Finally, for the last $(n-k)$ rounds he computes the responses. These steps are described in more detail next,

Note: the asterisk, $\alpha*$ or $\beta*$, indicates that the communication, sending or receiving, takes place with the card. Without the asterisk, the communication takes place with the reader. Algorithm assumes $(x \geq 1)$

Variables

n, k, x parameters of the protocol $(x \geq 1)$.
$count = 0$; *variable to count the number of bit errors*
m=[0,...,0]; *array to store the rounds when the card detects the bit errors*
p=[0,...,0]; *array to store the rounds when the reader detects the bit errors*
$i = 1$; *round*
Z= 0; *auxiliary variable to complement the challenges/responses when necessary*

Step 1

1.1. *if* $(i = k + 1)$: go to Step 3
1.2. Ask the card in advance with a random α_i*
1.3. Receive the corresponding response β_i* from the card
1.4. Receive the actual challenge from the reader, α_i
1.5.a. *if*$(\alpha_i = \alpha_i * \oplus Z)$: Send $\beta_i = \beta_i * \oplus Z$, $i = i + 1$, return to Step 1.1.
1.5.b. *else*: $Z = Z \oplus 1$, Send $\beta_i = \beta_i * \oplus Z$, $count = count + 1$, m($count$)= i,
if $(count = x)$: $count = 0$, $i = i + 1$, go to Step 2.
else: $i = i + 1$, go to Step 1.1.

Step 2

2.1. *if* $(i = k + 1)$: go to Step 3
2.2. Wait until receiving α_i from the reader
2.3. Send $\alpha_i* = \alpha_i \oplus Z$ to the card
2.4. Send at random β_i to the reader
2.5. Receive β_i* from the card
2.6.a. *if*$(\beta_i* = \beta_i \oplus Z)$: $i = i + 1$, return to Step 2.1
2.6.b. *else*: $count = count + 1$, p($count$)= i
if $(count > x)$: FINISH "Attack has failed"

else: $Z = Z \oplus 1$, $i = i + 1$, return to Step 2.1

Step 3

3.1. Obtain the first k bits of the bitstring s
3.2. By using $(n, k)ECC$, compute the last $(n - k)$ bits of s
3.3. Receive α_i from the reader
3.4. Send $\alpha_i* = \alpha_i \oplus Z$ to the card
3.5. Send $\beta_i = \alpha_i \oplus s_i$ to the reader by using the computed s_i
3.6.a. *if* $(i < n)$: $i = i + 1$, return to Step 3.3.
3.6.b. *else*: FINISH "Attack has been successful"

In Step 3.1, to obtain the first k bits of s, the adversary can apply,

$$s_i = \begin{cases} \alpha_i \oplus \beta_i & \text{if } i \neq p(t) \text{ for t=1 to x} \\ \\ \alpha_i \oplus \beta_i \oplus 1 & \text{if } i = p(t) \text{ for t=1 to x} \end{cases} \tag{3}$$

The parties assume that (up to) x bit errors due to noise have occurred, in the rounds $m(1)...m(x)$ from the point of view of the card, and in the rounds $p(1)...p(x)$ from the point of view of the reader. The parties correct these bit errors and compute properly the signed message y_A and y_B. Adversary relays the messages y_A, y_B and *opencommits*, and thus both parties remain unaware. We call the attention to the fact that when the adversary goes from Step 1 to Step 2, he has to wait for the actual challenge from the reader, and thus the time, T_a, between the response of the card (β_2* in Fig.4) and the next challenge (α_3*) is longer than usual ($T_a > T$).

The success probability of this adversary can be calculated as the probability of failing up to $2x$ out of (first) k rounds,

$$p_{a1} = \left(\frac{1}{2}\right)^k \cdot \sum_{t=0}^{t=2x} \binom{k}{t} \tag{4}$$

which is dramatically higher than that estimated in Sect. 3.2.

4.2 Modified Protocol and More General Attack

In order to thwart the attack described above, SPP could be modified in the following ways:

- Both parties inform each other (in a secure way) of the rounds where the bit errors have occurred. This information can be included in the final MAC; i.e. y_a and y_b. If a bit error is caused by noise, it should be detected by the parties in the same or consecutive rounds (we neglect the probability that two consecutive messages are corrupted by noise).
- The cards are equipped with trusted internal clocks which allow them to carry out precise timing measurements. This way, they can measure the time between sending β_i and receiving α_{i+1}, and thus detect that the interval T_a is longer than usual. As aforementioned, this possibility is difficult to be

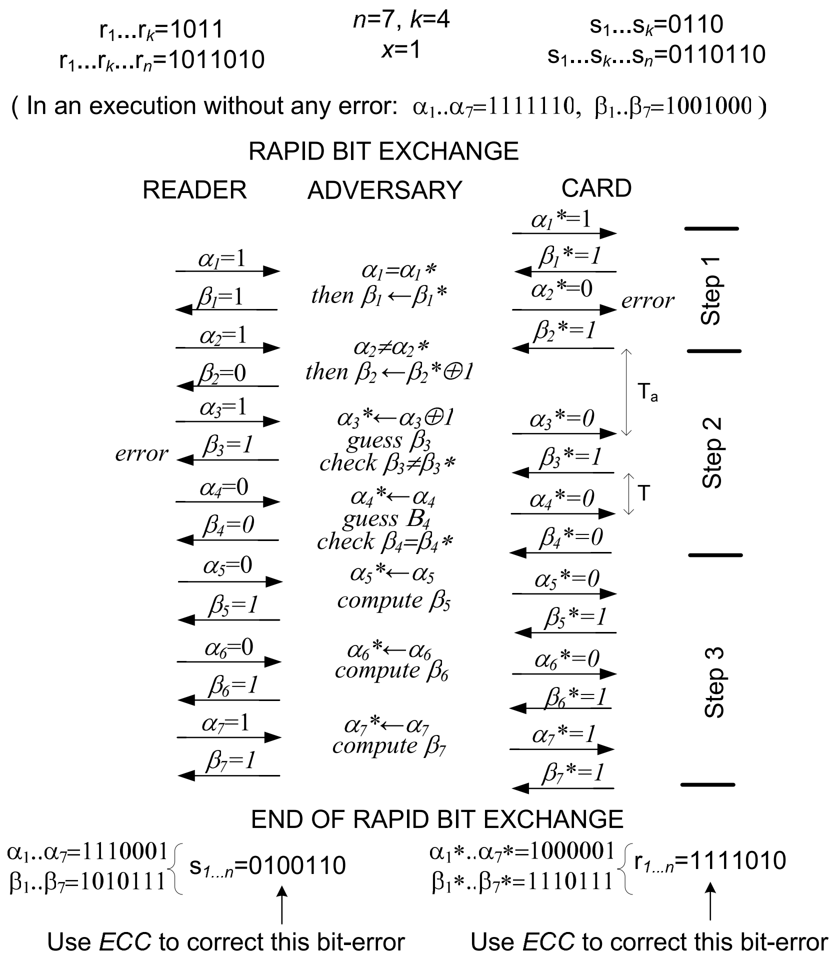

Fig. 4. Example of an attack on RFID implementation of SPP

implemented on RFID devices because the clock signal is externally supplied, and therefore not trusted; an adversary even could accelerate or decelerate it [13].

However, in spite of these modifications, the adversary can still carry out another more general attack, valid for (almost) any wireless system. An example of such an attack is shown in Fig.5. It is similar to the previous one but it has only two phases, and the adversary only can make up to x errors,

Variables

n, k, x *parameters of the protocol*
$count = 0$; *variable to count the number of bit errors*
i=1; *round*

Step 1

1.1. *if* $(i = k + 1)$, goes to Step 2
1.2. Receive α_i from the reader
1.3. Send $\alpha_i* = \alpha_i$ to the card
1.4. Send a random β_i to the reader
1.5. Receive the corresponding response β_i* from the card
1.6.a. *if* $(\beta_i = \beta_i*)$: $i = i + 1$, and returns to Step 1.1.
1.6.b. *else*: $count = count + 1$
if $(count > x)$: FINISH " Attack has failed"
else: $i = i + 1$, return to Step 1.1.

Step 2

2.1. Obtain the first k bits of the bitstring s; $s_i = \alpha_i * \oplus \beta_i*$
2.2. By using $(n, k)ECC$, compute the last $(n - k)$ bits of s
2.3. Receive α_i from the reader
2.4. Send $\alpha_i* = \alpha_i$ to the card
2.5. Send $\beta_i = \alpha_i \oplus s_i$ to the reader
2.6.a. *if* $(i < n)$: $i = i + 1$, and returns to Step 2.3.
2.7.b. *else*: FINISH "Attack has been successful"

In this attack the false acceptance probability reduces with respect to the attack described in the previous section, but it remains much higher than estimated by the authors (Sect. 3.2),

$$p_{a2} = \left(\frac{1}{2}\right)^k \cdot \sum_{t=0}^{t=x} \binom{k}{t} \tag{5}$$

4.3 Discussion

In accordance with the authors, HKP needs about twice as many rounds as SPP to obtain the same false acceptance ratio. It must be noticed that this reduction in the number of rounds should compensate the time consumed to carry out the additional tasks; i.e. computations (MAC and ECC) and sending data (commitment and final messages). However, the described attacks, for the RFID case in Sect. 4.1, and for a more general case in Sect. 4.2, increase the adversary's success probability. Table 1 compares some probabilities ($n = 37$) of false acceptance in HKP, p_{HKP}, calculated by Singelée and Preneel, p_{SPP}, and when the two versions of the attack are carried out, p_{a1} and p_{a2}. It is shown that when x increases, not only the protocol does not compensate the additional time that it needs, but its probabilities of false acceptance (if the attacks are performed) are even higher than those provided by HKP.

On the other hand, although the authors themselves consider in their analysis (Sect. 3.2) that an adversary can compute the last $(n - k)$ bits (a simple look-up table could be used to extend them), it must be noticed that even if the adversary was not able to compute them in time, and he needed more than one round, for instance q, the attack could still be applied. The adversary would

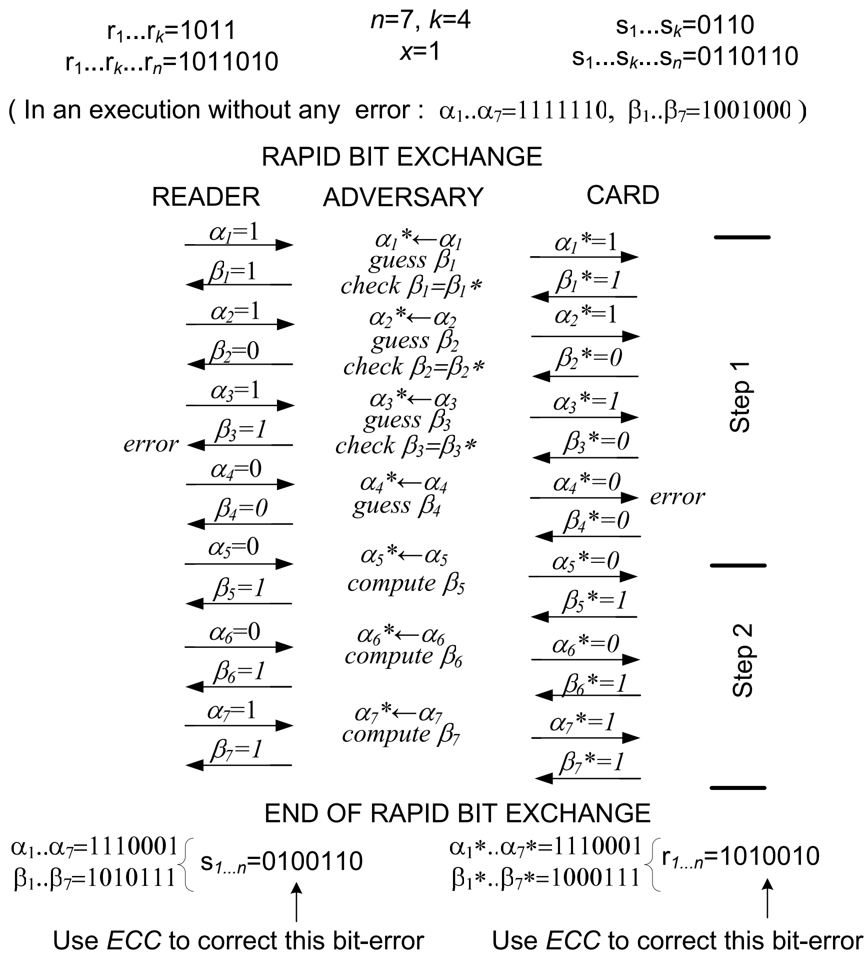

Fig. 5. Example of a more general attack on SPP

compute the last $(n - k - q)$ responses, and his success probability would equal the probability of making up to $2x$ (or x in the more general case) errors in the first $(k + q)$ rounds (not in the first k rounds as previously).

5 Conclusion

Distance bounding protocols are used to preclude relay attacks in proximity based authentication schemes. Hancke and Kuhn presented a suitable protocol to be employed in low cost, noisy wireless environments (RFID). This protocol is vulnerable to an attack where the adversary asks the card in advance without being detected, and this way his probability of guessing a response is not 1/2 but 3/4. Due to this high success probability, the number of rounds has to be

Table 1. Comparison of false acceptance probabilities for n=37

allowed errors	SPP				HKP
x	$(n,k)ECC$	p_{SPP}	p_{a1}	p_{a2}	p_{HKP}
0	(37,37)	$7.3 \cdot 10^{-12}$	$7.3 \cdot 10^{-12}$	$7.3 \cdot 10^{-12}$	$2.4 \cdot 10^{-5}$
1	(37,31)	$4.7 \cdot 10^{-10}$	$2.3 \cdot 10^{-7}$	$1.5 \cdot 10^{-8}$	$3.2 \cdot 10^{-4}$
2	(37,26)	$1.5 \cdot 10^{-8}$	$2.7 \cdot 10^{-4}$	$5.2 \cdot 10^{-6}$	$2.1 \cdot 10^{-3}$
3	(37,22)	$2.4 \cdot 10^{-7}$	$2.6 \cdot 10^{-2}$	$4.3 \cdot 10^{-4}$	$8.9 \cdot 10^{-3}$
4	(37,16)	$1.5 \cdot 10^{-5}$	0.5982	0.0384	0.0284

increased, especially when noise is taken into account and some failures must be tolerated. Singelée and Preneel have proposed a protocol which seeks to prevent this attack by applying mutual authentication, and thus reduces the number of rounds. This reduction in the number of rounds should compensate the time consumed to carry out the additional tasks needed by the protocol. From this point of view, the effectiveness of SPP is here questioned.

SPP is shown to be vulnerable to two attacks. The first being when it is implemented on RFID devices, and the second for a more general case, which dramatically increase the false acceptance probability. When the number of allowed failures increases, this probability can be even higher than that provided by HKP.

References

1. Finkenzeller, K.: RFID and Contactless Smart Card Applications. John Wiley & Sons (1993)
2. Hancke, G.: A Practical Relay Attack on ISO 14443 Proximity Cards. http://www.cl.cam.ac.uk/∼gh275/relay.pdf, (2005)
3. Kfir, Z., Wool, A.: Picking Virtual Pockets Using Relay Attack on Contactless Smartcard Systems. Cryptology ePrint Archive, Report 2005/052. `http://eprint.iacr.org`
4. ISO 14443. Identification Cards-Contactless Integrated Circuit Cards-Proximity cards. International Organization for Standardization, Geneva. `http://www.iso.org`
5. Hancke, G., Kuhn, M.: An RFID Distance Bounding Protocol. Proceeding of the IEEE, SecureComm (2005)
6. Singelée, D., Preneel, B.: Distance Bounding in Noisy Environments. In: F.Stajano et al. (Eds): ESAS 2007. LNCS, vol. 4572, pp. 101-115. Springer, Heidelberg (2007)
7. Singelée, D., Preneel, B.: Key Establishment Using Secure Distance Bounding Protocols. In: Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, 2007. IEEE-MobiQuitous 2007. August (2007).
8. Brands, S., Chaum, D.: Distance-Bounding Protocols. In: Helleseth, T. (ed) EUROCRYPT '93. LNCS, vol 765, pp. 344-359. Springer, Heidelberg (1994)

9. Čapkun, S., Buttyán, L., Hubaux, J.: SECTOR: Secure Tracking of Node Encounters in Multi-Hop Wireless Networks. In: Proceeding of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03), pp.21-32 (2003)
10. Jaffe, D.:Information about binary linear Codes, http://www.math.unl.edu/∼djaffe2/codes/webcodes/codeform.html
11. MacWilliams, F., Sloane, N.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1977)
12. Damgård, I.: Commitment Schemes and Zero-Knowledge Protocols. In: Damgård, I.B. (ed.) Lectures on Data Security. LNCS, vol. 1561, pp. 63-86. Springer, Heidelberg (1999)
13. Reid, J., González Nieto, J.M., Tang, T., Senadji, B.,: Detecting Relay Attacks with Timing-Based Protocools. Proceeding of the 2nd ACM Symposium on Information, Computer and Communication Security, pp. 204-213, (2007)