Passive Cryptanalysis of the UnConditionally Secure Authentication Protocol for RFID Systems

Mohammad Reza Sohizadeh Abyaneh

Department of Informatics, University of Bergen
{reza.sohizadeh@ii.uib.no}

Abstract. Recently, Alomair et al. proposed the first *UnConditionally Secure* mutual authentication protocol for low-cost RFID systems(UCS-RFID). The security of the UCS-RFID relies on five dynamic secret keys which are updated at every protocol run using a fresh random number (nonce) secretly transmitted from a reader to tags.
Our results show that, at the highest security level of the protocol (security parameter= 256), inferring a nonce is feasible with the probability of 0.99 by eavesdropping(observing) about 90 runs of the protocol. Finding a nonce enables a passive attacker to recover all five secret keys of the protocol. To do so, we propose a three-phase probabilistic approach in this paper. Our attack recovers the secret keys with a probability that increases by accessing more protocol runs. We also show that tracing a tag using this protocol is also possible even with less runs of the protocol.

Key words: : RFID, Authentication Protocol, Passive Attack

1 Introduction

As of today, RFID (Radio Frequency Identification) is referred to as the next technological revolution after the Internet. A typical RFID system involves a *reader*, a number of *tags*, which may range from the battery-powered, to the low-cost ones with even no internal power, and a *database*. RFID systems enable the identification of objects in various environments. They can potentially be applied almost everywhere from electronic passports[20,21], contactless credit cards[19], to supply chain management[22,23,24].
Keeping RFID systems secure is imperative, because they are vulnerable to a number of malicious attacks. For low-cost RFID systems, security problems become much more challenging, as many traditional security mechanisms are inefficient or even impossible due to resource constraints. Some existing solutions utilize traditional cryptographic primitives such as hash or encryption functions, which are often too expensive to be implemented on low-cost RFID tags.
Another method of securing RFID systems has been the lightweight approach. These solutions base themselves on mostly lightweight operations (e.g. bitwise

or simple arithmetic operations) instead of more expensive cryptographic primitives. The HB-family(HB$^+$,HB^{++}, HB*,etc.) [1,2,3,4,5,7,6,8] and the MAP-family(LMAP,EMAP,M2AP,etc)[9,10,11] authentication protocols, are some examples of this kind. However, proposed lightweight protocols so far have been targeted to various successful attacks and therefore, the search for a concrete lightweight solution for authentication in low-cost RFID tags still continues.
Recently, Alomair et al. embarked on the notion of UnConditionally Secure mutual authentication protocol for RFID systems (UCS-RFID)[17]. UCS-RFID's security relies mainly on the freshness of five secret keys rather than the hardness of solving mathematical problems. Freshness in the keys is guaranteed with a *key updating* phase at every protocol run by means of a fresh random number (nonce). This nonce is generated at the reader side due to low-cost tags constraints, and delivered to the tag secretly. This allows the tags to benefit from the functionalities of random numbers without the hardware to generate them.
Our Contribution. In this paper, we present a three-phase probabilistic passive attack against the UCS-RFID protocol to recover all the secret keys in the protocol. Our attack is mainly based on a weakness observed in the protocol(section 3). To put in a nutshell, the weakness implies that the more outputs we have from consecutive runs of the protocol, the more knowledge we will obtain on the nonces in these protocol runs. In other words, having more number of protocol run outputs observed, we are able to determine some of the nonces (*victim* nonces) with higher probability. It should be noted that this weakness has also been tackled by the authors in [17]. Nevertheless we will show that the security margin they expected from the protocol has been overestimated. Finding the victim nonce in the protocol paves the way toward adopting an attacking scenario to achieve all of the five secret keys in the system.
Outline. The remainder of this paper is organized as follows. In section 2, we briefly describe the UCS-RFID protocol. In section 3 the weakness of the protocol is investigated thoroughly. Section 4 and 5 describes our attacking scenario to recover the keys, and trace the tag in the protocol. Finally, section 6 concludes the paper.

2 Description of the UCS-RFID Protocol

The UCS-RFID authentication protocol consists of two phases: the *mutual authentication phase* and the *key updating phase*. The former phase mutually authenticates an RFID reader and a tag. In the latter phase both the reader and the tag update their dynamic secret keys for next protocol runs.
In this protocol, first the security parameter, N, is specified and a $2N$-bit prime integer, p, is chosen. Then, each tag T is loaded with an N-bit long identifier, $A^{(0)}$, and five secret keys, $k_a^{(0)}, k_b^{(0)}, k_c^{(0)}, k_d^{(0)}$ and $k_u^{(0)}$ chosen independently and uniformly from $\mathbb{Z}_{2^N}, \mathbb{Z}_p, \mathbb{Z}_p\backslash\{0\}, \mathbb{Z}_{2^N}$ and $\mathbb{Z}_p\backslash\{0\}$ respectively.
Notation
- N: security parameter.
- p: a prime number in $\mathbb{Z}_{2^N}$

- A^x, B^x, C^x, D^x: observable outputs of x^{th} protocol run
- $n = n_l||n_r$: random number in $\mathbb{Z}_{2^N}$
- n_l, n_r: left and right *half-nonces*

2.1 Mutual Authentication Phase

Figure 1 shows one instance run of the mutual authentication phase in the UCS-RFID protocol. The reader starts the interrogation with a "Hello" message which

Specifications
- Public parameters: p, N.
- Secret parameters(shared between R and T): $k_a^{(0)}, k_b^{(0)}, k_c^{(0)}, k_d^{(0)}, k_u^{(0)}$.
Mutual Authentication Phase
(1) $R \Rightarrow T : Hello$
(2) $T \Rightarrow R : A^{(i)}$
(3) $R \Rightarrow T : B^{(i)}, C^{(i)}$
(4) $T \Rightarrow R : D^{(i)}$

Fig. 1. i^{th} run of the mutual authentication phase in the UCS-RFID protocol

is responded by tag's dynamic identifier $A^{(i)}$. The reader then looks up in the database for a set of five keys$(k_a, k_b, k_c, k_d, k_u)$ which corresponds to $A^{(i)}$. If this search is successful, it means that the tag is authentic. Having the tag authenticated, the reader generates a $2N$-bit random nonce $n^{(i)}$ uniformly drawn from $\mathbb{Z}_p^*$, calculates messages $B^{(i)}$, $C^{(i)}$ by (2),(3) and sends them to the tag.

$$A^{(i)} \equiv n_l^{(i-1)} + k_a^{(i)} \ mod \ 2^N \tag{1}$$

$$B^{(i)} \equiv n^{(i)} + k_b^{(i)} \ mod \ p \tag{2}$$

$$C^{(i)} \equiv n^{(i)} \times k_c^{(i)} \ mod \ p \tag{3}$$

The tag first checks the integrity of the received messages by (4):

$$(B^{(i)} - k_b^{(i)}) \times k_c^{(i)} \equiv C^{(i)} \ mod \ p \tag{4}$$

This check implies the authenticity of the reader as well. Then, the tag extracts the nonce $n^{(i)}$ by (5.)

$$n^{(i)} \equiv (B^{(i)} - k_b^{(i)}) \ mod \ p \tag{5}$$

To conclude the mutual authentication phase, the tag transmits $D^{(i)}$ as a receipt of obtaining $n^{(i)}$.

$$D^{(i)} = n_l^{(i)} \oplus k_d^{(i)} \tag{6}$$

2.2 Key Updating Phase

After a successful mutual authentication, both the reader and the tag update their keys and dynamic identifier ($A^{(i)}$) for the next protocol run.

$$k_a^{(i+1)} = n_r^{(i)} \oplus k_a^{(i)} \tag{7}$$
$$k_b^{(i+1)} \equiv k_u^{(i)} + (n^{(i)} \oplus k_b^{(i)}) \ mod \ p \tag{8}$$
$$k_c^{(i+1)} \equiv k_u^{(i)} \times (n^{(i)} \oplus k_c^{(i)}) \ mod \ p \tag{9}$$
$$k_d^{(i+1)} = n_r^{(i)} \oplus k_d^{(i)} \tag{10}$$
$$k_u^{(i+1)} \equiv k_u^{(i)} \times n^{(i)} \ mod \ p \tag{11}$$
$$A^{(i+1)} \equiv n_l^{(i)} + k_a^{(i+1)} \ mod \ 2^N \tag{12}$$

It should be noted that the dynamic values have been proved to preserve their properties of independency and uniformity after updating[17].

3 Observation

In this section, we shed more light on a weakness in the UCS-RFID protocol which becomes the origin of our proposed attack presented in the subsequent section.
By xoring (7) and (10), we have:

$$k_a^{i+1} \oplus k_d^{i+1} = k_a^i \oplus k_d^i \tag{13}$$

Equation (13) shows that the difference between k_a and k_d remains the same for two consecutive runs of the protocol. This statement can also be generalized for every r arbitrary run of the protocol the as following:

$$k_a^{r+1} \oplus k_d^{r+1} = k_a^r \oplus k_d^r = \ldots = k_a^0 \oplus k_d^0 = L \tag{14}$$

By using (14), for outputs A and D in m consecutive runs of the protocol, we have:

$$A^{(i)} \equiv n_l^{(i-1)} + k_a^{(i)} \ mod \ 2^N \tag{15}$$
$$D^{(i)} = n_l^{(i)} \oplus (k_a^{(i)} \oplus L) \tag{16}$$
$$A^{(i+1)} \equiv n_l^{(i)} + (k_a^{(i)} \oplus n_r^{(i)}) \ mod \ 2^N \tag{17}$$
$$D^{(i+1)} = n_l^{(i+1)} \oplus (k_a^{(i)} \oplus L \oplus n_r^{(i)}) \tag{18}$$
$$\vdots$$
$$A^{(i+m-1)} \equiv n_l^{(i+m-2)} + (k_a^{(i)} \bigoplus_{j=i}^{i+m-2} n_r^{(j)}) \ mod \ 2^N \tag{19}$$
$$D^{(i+m-1)} = n_l^{(i+m-1)} \oplus (k_a^{(i)} \oplus L \bigoplus_{j=i}^{i+m-2} n_r^{(j)}) \tag{20}$$

It is apparent that we have a set of $2m$ equations with $2m+2$ variables. These variables can be divided into two groups:
1. $2m$ half-nonces: $n_l^{(i-1)}, \ldots, n_l^{(i+m-1)}, n_r^{(i)}, \ldots, n_r^{(i+m-2)}$
2. L and $k_a^{(i)}$.

So, if we fix the value of variables L and $k_a^{(i)}$, we end up with $2m$ equations and $2m$ half-nonce variables. This implies that the $2m$ half-nonces can not be chosen independently and fulfil the above equations simultaneously. In other words, if we observe the outputs of m consecutive runs of the protocol, it is only necessary to search over all possible sequences of $k_a^{(i)}$ and L, which is 2^{2N}, and then it will be possible to find all $2m$ half-nonces uniquely. As we will see, this weakness is the result of introduction of a tighter bound for the half-nonces while we keep observing more runs of the protocol.
By the randomness nature of the generated half-nonces, the total number of possible sequences for them(2^{2N}) is uniformly distributed over them. This implies that each of the $2m$ half-nonces is expected to have a bound of $\sqrt[2m]{2^{2N}}$ possible values (comparing to its previous bound which was N). Therefore, for m consecutive protocol runs, the total number of possible values distributed over the $2m$ half-nonces is $2m\sqrt[2m]{2^{2N}}$ [17].
Now, if we exclude the value which half-nonces has taken already($2m\sqrt[2m]{2^{2N}} - 2m$), we can calculate the probability that at least one half-nonce does not receive another possible value (remains constant). To do so, we utilize the well-known problem in probability theory(i.e. Given r balls thrown uniformly at random at b bins, the probability that at least one bin remains empty which is calculated by (21))[18]:

$$\Pr(\text{at least one bin remains empty}) = 1 - \frac{\binom{r-1}{b-1}}{\binom{b+r-1}{b-1}} \tag{21}$$

Now, it only requires to substitute $b = 2m$ and $r = 2m.\sqrt[2m]{2^{2N}} - 2m$ in (21) and then we will have (22). The result is plotted in Figure 2.

$$P_h = \Pr(\text{at least one half-nonce remains constant}) = 1 - \frac{\binom{2m.\sqrt[2m]{2^{2N}} - 2m - 1}{2m-1}}{\binom{2m.\sqrt[2m]{2^{2N}} - 1}{2m-1}} \tag{22}$$

Figure 2 shows the probability of inferring at least one half-nonce in terms of the number of consecutive runs of the protocol required to be observed to do so. For example, if we observe 35 runs of the protocol runs with N=256, we will be able to determine at least one of the 70 transmitted half-nonces with the probability of more than 0.99.
We will use the term "*victim* half-nonce" for inferred half-nonce and notation m_h instead of m for the number of consecutive runs of the protocol required to infer one half-nonce hereafter.

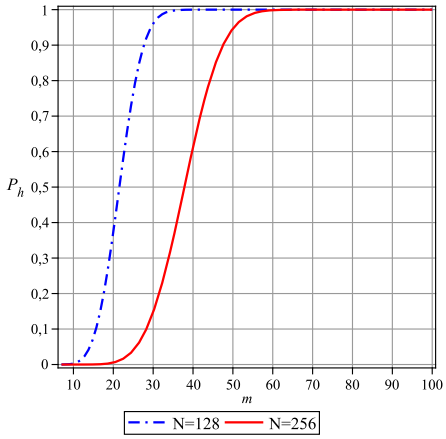

Fig. 2. The number of consecutive protocol runs an adversary must observe(m) in order to infer at least one half-nonce for $N = 128, 256$

4 Our Attack Scenario

In the previous section, we presented a probabilistic approach to find the number of consecutive runs of the protocol to infer one half-nonce. But in our attack, we need to have a complete nonce(left and right corresponding half-nonces) to recover all secret keys. To achieve this goal, we propose an attacking scenario which consists of the three following phases:

1. Finding the total number of necessary consecutive runs of the protocol to find a complete victim nonce (m_t).
2. Finding the victim nonce.
3. Recovering the secret keys.

4.1 Phase I: Finding m_t

In section 3, we proposed a probabilistic way to calculate the number of consecutive runs that must be observed by an adversary to infer a half-nonce(m_h). It is obvious that if we keep observing more runs of the protocol(i.e. more than m_h), after each extra observation, another half-nonce can be inferred. This is simply possible by eliminating the two equations which contain the first victim half-nonce and adding two newly observed equations to the set of equations (15-20) and then, we again have $2m_h$ equations and $2m_h + 2$ variables which yield another half-nonce inference.
If we intend to find a complete nonce, we must continue observing the runs of the protocol until we infer two corresponding victim half-nonces to form a complete nonce. To do so, we should first calculate the probability that the inferred half-nonce at $(m_e + m_h)^{th}$ run matches one of the previously victim half-nonces.

As we know, after m_h runs of the protocol, we accomplish to find one victim half-nonce, after m_e extra runs of the protocol, we have $\beta = 2m_h + 2m_e$ equations and β half-nonces which $m_e + 1$ of them can be inferred. The probability that none of these $m_e + 1$ half-nonces match is:

$$\begin{aligned}\Pr(\text{Having no pair after } m_h + m_e \text{ runs}) &= \frac{(\beta-1)}{\beta} \times \frac{(\beta-2)}{\beta} \times \ldots \times \frac{(\beta-m_e)}{\beta} \\ &= \frac{\prod_{i=1}^{m_e}(\beta-i)}{\beta^{(m_e)}} \end{aligned} \tag{23}$$

Consequently, the probability of having at least one pair after observing m_e runs is simply calculated by (24).

$$\begin{aligned} P_e &= \Pr(\text{Having at least one pair of matching half-nonces after } m_h + m_e \text{ runs}) \\ &= 1 - \frac{\prod_{i=1}^{m_e}(\beta-i)}{\beta^{(m_e)}} \end{aligned} \tag{24}$$

By using (22) and (24) the total number of protocol runs to have at least one complete victim nonce ($m_t = m_h + m_e$) can be calculated by (25) and is plotted in Figure 3.

$$\begin{aligned} P_t &= \Pr(\text{Having at least one complete nonce after } m_t \text{ runs}) \\ &= (P_e | m_h = h) \times Pr(m_h = h) = (P_e | m_h = h) \times P_h(h) \end{aligned} \tag{25}$$

Remark The authors of [17] have also calculated m_t by using some other protocol outputs (B and C). Figure 3 compares our results with what the authors "Expected". This comparison has been conducted for two different security parameters N=128,N=256 which are plotted on the left and right respectively. The results show that the security margin of the protocol in terms of the number of consecutive runs that must be observed to infer one nonce is less than what the designers of the protocol expected. In other words, we need less number of protocol runs to infer at least one nonce. For example a passive adversary is able to infer a complete nonce with high probability of 0.99 by eavesdropping less that 60 and 90 runs of the protocol for the key size of 128 and 256 bits respectively. These numbers were expected to be 110 and 200 respectively.

4.2 Phase II: Finding the constant nonce

Having m_h consecutive runs of the protocol observed, we have one constant half-nonce or one half-nonce with only one possible value. In order to find this half-nonce, we adopt the following algorithm.

Algorithm Inputs :$A^{(i)}, \ldots, A^{(i+m_t-1)}, D^{(i)}, \ldots, D^{(i+m_t-1)}$

1. Determine a level of confidence(probability) for the final results.

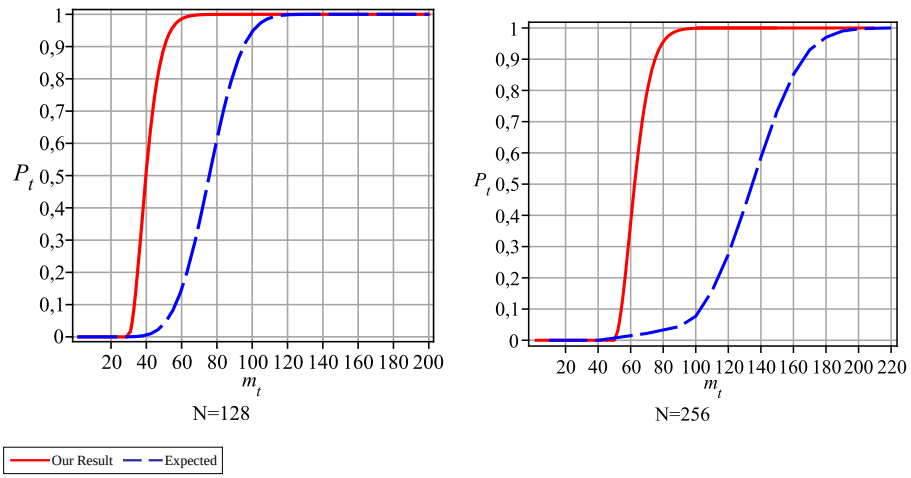

Fig. 3. Comparison of expected security margin of the UCS-RFID protocol and our results in terms of the number of consecutive protocol runs an adversary must observe in order to infer at least one nonce.

2. Find the m_h, m_t related to the determined probability from Figures 1,2 respectively.
3. Calculate $m_e = m_t - m_h$
4. Choose two random numbers from $\mathbb{Z}_{2^N}$ and assign them to L,$k_a^{(i)}$ respectively.
5. Find $2m$ nonces $(n_l^{(i-1)}, \ldots, n_l^{(i+m_h-1)}, n_r^{(i)}, \ldots, n_r^{(i+m_h-2)})$ as follows.
   - Find $n_l^{(i-1)}$ from (15) i.e. $n_l^{(i-1)} \equiv A^{(i)} - k_a^{(i)} \ mod\ 2^N$.
   - Find $n^{(i)}$ from (16) i.e. $n_l^{(i)} = D^{(i)} \oplus (k_a^{(i)} \oplus L)$.
   - Find $n_r^{(i)}$ from (17) i.e. $n_r^{(i)} \equiv (A^{(i+1)} - n_l^{(i)} \ mod\ 2^N) \oplus k_a^{(i)}$.
   - $\vdots$
   - Find $n_r^{(i+m_h-2)}$ from (19)i.e. $n_r^{(i+m_h-2)} \equiv (A^{(i+m_h-1)} - n_l^{(i+m_h-2)} \ mod\, 2^N) \oplus (k_a^{(i)} \bigoplus_{j=i}^{i+m_h-2} n_r^{(j)})$.
   - Find $n_l^{(i+m_h-1)}$ from (20) i.e. $n_l^{(i+m_h-1)} = D^{(i+m_h-1)} \oplus (k_a^{(i)} \oplus L) \bigoplus_{j=i}^{i+m_h-2} n_r^{(j)}$.
6. Repeat 4 and 5 as many times as we observe that only one half-nonce keeps its value for all of the repetitions.
7. Save the constant(victim) half-nonce.
8. Observe another run of the protocol.
   - $A^{(i+m_h)} \equiv n_l^{(i+m_h-1)} + (k_a^{(i)} \bigoplus_{j=i}^{i+m_h-1} n_r^{(j)}) \ mod\ 2^N$
   - $D^{(i+m_h)} = n_l^{(i+m_h)} \oplus (k_a^{(i)} \oplus L \bigoplus_{j=i}^{i+m_h-1} n_r^{(j)})$.
9. Replace the equations corresponding to the found victim half-nonce with two newly observed equations in the equation set (15-20).

10. Repeat 4,5,6,7,8 for m_e times.
11. Match two corresponding victim half-nonces(e.g. $n_l^{(j)}, n_r^{(j)}$).
12. Output the victim nonce ($n^{(j)} = n_l^{(j)} || n_r^{(j)}$).

4.3 Phase III: Key Recovery

In the previous two phases of our attack, we accomplished to find a complete victim nonce $n^{(j)}$,with a certain probability, by observing m_t consecutive runs of the protocol. Now, we present how an adversary is able to recover all five secret keys of the protocol. To find $k_a^{(j)}, k_b^{(j)}, k_c^{(j)}$ and $k_d^{(j)}$, we should follow(26-29).

$$k_a^{(j)} \equiv (A^{(j+1)} - n_l^{(j)}) \oplus n_r^{(j)} mod\ 2^N \tag{26}$$

$$k_b^{(j)} \equiv B^{(j)} - n^{(j)}\ mod\ p \tag{27}$$

$$k_c^{(j)} \equiv (\frac{1}{n^{(j)}}\ mod\ p) \times C^{(j)}\ mod\ p \tag{28}$$

$$k_d^{(j)} = n_l^{(j)} \oplus D^{(j)} \tag{29}$$

To recover $k_u^{(j)}$, we need to find the nonce in the next run ($n^{(j+1)}$), thus we should calculate the updated keys for the $(j+1)^{th}$ run using (7) and (10).

$$k_a^{(j+1)} = k_a^{(j)} \oplus n_r^{(j)} \tag{30}$$

$$k_d^{(j+1)} = k_d^{(j)} \oplus n_r^{(j)} \tag{31}$$

Then we have:

$$n_l^{(j+1)} = D^{(j+1)} \oplus k_d^{(j+1)} \tag{32}$$

$$k_a^{(j+2)} = A^{(j+2)} \oplus n_l^{(j+1)} \tag{33}$$

Using (30) and (33), we can write:

$$n_r^{(j+1)} = k_a^{(j+2)} \oplus k_a^{(j+1)} \tag{34}$$

Finally, by using (27),(32) and,(34) we can find $k_u^{(j)}$.

$$k_u^{(j)} \equiv B^{(j+1)} - n^{(j+1)} - (k_b^{(j)} \oplus n^{(j+1)})\ mod\ p \tag{35}$$

The procedure above provides us with our objective to recover all of the secret keys with a certain probability(P_t). This probability can be increased by paying the price of having more protocol run outputs available.
Furthermore, as it can be seen from the (32) and (34), next nonce is also achievable. This implies that the secret keys of the next run can also be calculated by using (26-35) for the next run. This is an ongoing procedure which yields the keys of any arbitrary run of the protocol(r) which $r > j$. Being able to generate the future secret keys, an adversary is capable of either impersonating both the reader and the tag or tracing the tag.

5 On the Traceability of the UCS-RFID

In the previous section, we presented a probabilistic key recovery attack against the UCS-RFID protocol. We mentioned that according to Figure 3, we need to have about 90 runs of the protocol to be almost sure that our found keys are correct. But with less number of protocol run outputs, we still can apply an attack against the traceability of the protocol. In this section, we formally investigate the *untraceability* of the UCS-RFID based on the formal description in [12].

5.1 Adversarial Model

According to [12], the means that are accessible to an attacker are the following: We denote a tag and a reader in i^{th} run of the protocol by $\mathcal{T}_i$ and $\mathcal{R}_i$, respectively.

- $\mathsf{Query}(\mathcal{T}_i, m_1, m_3)$: This query models the attacker $\mathcal{A}$ sending a message m_1 to the tag and sending the m_3 after receiving the response.
- $\mathsf{Send}(\mathcal{R}_i, m_2)$: This query models the attacker $\mathcal{A}$ sending a message m_2 to the Reader and being acknowledged.
- $\mathsf{Execute}(\mathcal{T}_i, \mathcal{R}_i)$: This query models the attacker $\mathcal{A}$ executing a run of protocol between the Tag and Reader to obtain the exchanged messages.
- $\mathsf{Reveal}(\mathcal{T}_i)$: This query models the attacker $\mathcal{A}$ obtaining the information on the Tag's memory.

A *Passive Adversary*, $\mathcal{A}_\mathcal{P}$, is capable of eavesdropping all communications between a tag and a reader and accesses only to the $\mathsf{Execute}(\mathcal{T}_i, \mathcal{R}_i)$: .

5.2 Attacking Untraceability

The result of application of an oracle for a passive attack $\mathcal{O}_\mathcal{P} \subseteq \{Execute(.)\}$ on a tag T in the run i is denoted by $w_i(T)$. Thus, a set of I protocol run outputs, $\Omega_I(T)$, is:
$\Omega_I(T) = \{w_i(T) | i \in I\}\,; I \subseteq N$;($N$ denotes the total set of protocol runs).
The formal description of attacking scenario against untraceability of a protocol is as following:
1. $\mathcal{A}_\mathcal{P}$ requests the *Challenger* to give her a target T.
2. $\mathcal{A}_\mathcal{P}$ chooses I and calls $Oracle(T, I, \mathcal{O}_\mathcal{P})$ where $|I| \leq l_{ref}$ receives $\Omega_I(T)$.
3. $\mathcal{A}_\mathcal{P}$ requests the *Challenger* thus receiving her challenge T_1, T_2 ,$I_1$and I_2
4. $\mathcal{A}_\mathcal{P}$ calls $Oracle(T_1, I_1, \mathcal{O}_\mathcal{P})$, $Oracle(T_2, I_2, \mathcal{O}_\mathcal{P})$ then receives $\Omega_{I_1}(T_1)$, $\Omega_{I_2}(T_2)$.
5. $\mathcal{A}_\mathcal{P}$ decides which of T_1 or T_2 is T, then outputs her guess T'.

For a security parameter,k, if $Adv^{UNT}_{\mathcal{A}_\mathcal{P}}(k) = 2Pr(T' = T) - 1 > \epsilon$ then we can say that the protocol is traceable.
For UCS-RFID case, as Figure 3 implies, an adversary $\mathcal{A}_\mathcal{P}$ needs only to access to about 40 and 65 consecutive runs of the protocol to be able to determine $n^{(j)}$ with a probability of more than 0.5 (e.g. 0.6) for k =128 and 256 respectively and

then according to section 4.3, she will be able to recover the keys of subsequent runs. After, key recovery, the adversary can easily distinguish a target tag with any other challenge tag given by the challenger. So we have:
$\forall l_{ref} \geq 40$, $Adv^{UNT}_{\mathcal{A}_\mathcal{P}}(128) = 2Pr(T' = T) - 1 = 0.1 > \epsilon$.
$\forall l_{ref} \geq 65$,$Adv^{UNT}_{\mathcal{A}_\mathcal{P}}(256) = 2Pr(T' = T) - 1 = 0.1 > \epsilon$.

6 Conclusions

The design of suitable lightweight security protocols for low-cost RFID tags is still a big challenge due to their severe constraints. Despite of interesting proposals in the literature, this field still lacks a concrete solution.
Recently, Alomair *et al* have proposed the first authentication protocol based on the notion of unconditional security. Regardless of some inefficiencies in UCS-RFID authentication protocol, such as: large key sizes, using modular multiplication ,etc ,which makes this protocol an unsuitable nominate for low-cost RFID tag deployment, we presented a passive attack which showed that even the security margin which was expected to be yielded by UCS-RFID has also been overestimated.
In our attack, we showed that a passive adversary is able to achieve the all secret keys of the system with a high probability of 0.99 by eavesdropping less that 60 and 90 runs of the protocol for the key size of 128 and 256 bits respectively. Tracing the tag in the protocol is also feasible even by less number of runs of the protocol (e.g. 40, 65).
Our results suggest a major rethink in the design of the authentication protocols for RFID systems based on unconditional security notion. Drastic changes are necessary to fulfil both technological constraints and security concerns in RFID systems.

References

1. N.J. Hopper and M. Blum. : Secure Human Identifcation Protocols , in C. Boyd (ed.) Advances in Cryptology - ASIACRYPT 2001, Volume 2248, Lecture Notes in Computer Science, pp. 52-66, Springer-Verlag, (2001).
2. J. Bringer, H. Chabanne, and E. Dottax.: HB++: a Lightweight Authentication Protocol Secure Against Some Attacks, IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing SecPerU, (2006).
3. Julien Bringer and Herve Chabanne.: Trusted-HB: a low-cost version of HB+ secure against man-in-the-middle attacks. CoRR, abs/0802.0603, (2008).
4. Julien Bringer, Herve Chabanne, and Emmanuelle Dottax.: HB++: a lightweight authentication protocol secure against some attacks, In Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2006), pages 28-33. IEEE Computer Society, (2006).
5. Dang Nguyen Duc and Kwangjo Kim.: Securing HB+ against GRS man-in-the-middle attack, In Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security, Jan. 23-26, 2007, Sasebo, Japan, page 123, (2007).

6. Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin: HB♯: Increasing the security and effciency of HB+, Advances in Cryptology EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, Proceedings, volume 4965 of Lecture Notes in Computer Science, pages 361-378.Springer, (2008).
7. J. Munilla and A. Peinado.: HB-MP: A further step in the HB-family of lightweight authentication protocols. Computer Networks, (2007).
8. Mukundan Madhavan,Andrew Thangaraj,Yogesh Sankarasubramaniam and Kapali Viswanathan: NLHB : A Non-Linear Hopper Blum Protocol,IEEE National Conference on Communications (NCC),2010, CoRR abs/1001.2140 (2010).
9. Peris-Lopez, Hernandez-Castro,Estevez Tapiador, and Ribagorda: LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags , RFIDSec 06, (2006).
10. P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda: M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags, in International Conference on Ubiquitous Intelligence and Computing (UIC06), vol. 4159 of LNCS, pp.912-923 (2006).
11. P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda: EMAP: An Efficient Mutual-Authentication Protocol for Low-cost RFID tags , in OTM Federated Conferences and Workshop: IS Workshop, (2006).
12. Avoine G. :Adversarial Model for Radio Frequency Identification. Cryptology ePrint Archive, Report 2005/049, (2005).
13. M. Ohkubo, K. Suzuki, and S. Kinoshita: Cryptographic Approach to Privacy-Friendly Tags, in RFID Privacy Workshop, (2003).
14. D. Henrici, and P. Muller: Hash-based Enhancement of Location Privacy for Radio Frequency Identification Devices using Varying Identifiers, in Proceedings of PerSec04,IEEE PerCom, pp.149-153, (2004).
15. D. Henrici, and P. Muller: Providing Security and Privacy in RFID Systems Using Triggered Hash Chains, in PerCom'08, 50-59, (2008).
16. L.S. Kulseng: Lightweight Mutual Authentication, Owner Transfer, and Secure Search Protocols for RFID Systems , Master Thesis, Iowa State University,Ames, (2009).
17. B. Alomair, L. Lazos , R. Poovendran: Securing Low-cost RFID Systems: an Unconditionally Secure Approach , RFIDsec'10 Asia, Singapore, (2010).
18. W. Feller: An Introduction to Probability Theory and its Applications, Wiley India Pvt. Ltd., (2008).
19. T.S. Heydt-Benjamin, D.V. Bailey, K. Fu, A. Juels, and T. O'Hare: Vulnerabilities in First-Generation RFID-Enabled Credit Cards, Proc. 11th Int'l Conf. Financial Cryptography and Data Security (FC '07), pp. 2-14, (2007).
20. D.Carluccio, K.Lemke, C.Paar: E-passport: The Global Traceability or How to feel like a UPS package, Proceeding of WISA'06, LNCS 4298, Springer, pp.391-404, (2007).
21. J.-H. Hoepman, E. Hubbers, B. Jacobs, M. Oostdijk, and R.W. Schreur, Crossing Borders: Security and Privacy Issues of the European e-Passport, Proc. First Int'l Workshop Security (IWSEC '06), pp.152-167 (2006).
22. CASPIAN, Boycott Benetton: http://www.boycottbenetton.com (2007).
23. Mitsubishi Electric Asia Switches on RFID: www.rfidjournal.com/article/articleview/2644/ (2006).
24. Target, Wal-Mart Share EPC Data: http://www.rfidjournal.com/article/articleview/642/1/1/ (2005).