J Supercomput (2013) 65:949–960
DOI 10.1007/s11227-013-0873-3

Cryptanalysis of the RNTS system

**Pablo Picazo-Sanchez · Lara Ortiz-Martin ·
Pedro Peris-Lopez · Julio Cesar Hernandez-Castro**

Published online: 24 January 2013

Abstract Internet of Things is a paradigm that enables communication between different devices connected to a local network or to Internet. Identification and communication between sensors used in Internet of Things and devices like smart-phones or tablets are established using radio frequency identification technology. However, this technology still has several security and privacy issues because of its severe computational constraints. In 2011, Jeong and Anh proposed the combined use of an authentication radio frequency identification protocol together with a ticket issuing system for bank services (in J. Supercomput. 55:307, 2011). In this paper we show that their message generation is weak, because it abuses the XOR operation and the use of a counter, which leaks too much secret protocol information. Our analysis shows important security faults that ruin most of the security properties claimed in the original paper. More precisely, information privacy (via a disclosure and leakage attack) and location privacy (traceability attack) are both compromised. Moreover, an attacker can disrupt the proper working of the system by exploiting the fact that message integrity is not properly checked.

Keywords RFID · Authentication · Banking services · Cryptanalysis

P. Picazo-Sanchez (✉) · L. Ortiz-Martin
Department of Applied Mathematics, University School of Computer Science (UPM) of Madrid, Madrid, Spain
e-mail: ppicazo@eui.upm.es

L. Ortiz-Martin
e-mail: lara.ortiz@eui.upm.es

P. Peris-Lopez
Computer Security Lab (COSEC), Carlos III University of Madrid, Madrid, Spain
e-mail: pperis@inf.uc3m.es

J.C. Hernandez-Castro
School of Computing, University of Kent, Canterbury, UK
e-mail: Julio.Hernandez-Castro@kent.ac.uk

1 Introduction & related work

Internet of Things (IoT) is a novel growing wireless communication. Its components are sensors which are usually called things. That things sense the environment retrieving different data of temperature, humidity, motion, wind velocity or ultrasound among others. These things that compose the IoT have some characteristics in common like high computational constraints and its necessity for saving battery. So taken into account those constraints, to guarantee a minimum security and privacy level is a new challenge for researchers [1, 7, 15, 18, 19].

Radio Frequency IDentification (RFID) is a technology based on radio frequency to identify objects, animals or humans. An RFID system typically consists of three main components: tags, readers and a database. The tag is generally a very constrained microchip with an antenna that connects to items that we want to authenticate or track. Readers are responsible for communicating with RFID tags through the radio channel and can read or modify the information stored on tags' memory. The last component is a centralized or distributed back-end system that is often connected to readers through a secure link. RFID systems have many applications in libraries, e-passports, access control, food traceability, inventory control, supply chain management, e-Health and many other fields.

Leakage of private information and traceability are the two main problems linked to RFID technology [10]. In fact, one of the critical requirements for RFID authentication protocols is that these should offer protection against traceability. This is particularly relevant when tags are embedded within items (e.g, clothing) and its operation is completely transparent and ubiquitous. Aside from mounting traceability attacks, stronger and more challenging attacks can be run by attackers such as the recovery of the long-term secrets stored on tags, compromising the privacy of tags' holder. A few other possible attacks are: eavesdropping, replay attacks, data integrity attacks and Denial-of-Service (DoS) attacks.

RFID security has to deal with tag footprint, storage and power consumption constrains. There are a number of lightweight proposals based on triangular functions (e.g. bitwise XOR and modular addition) and rotations [4, 22]. On the other hand, there are some other proposals that rely on the more hardware demanding hash functions [13, 17].

Join of RFID and IoT is widely studied on the literature [5, 14, 15, 19–21]. In the following, the RNTS (RFID Number Ticket Service) system proposed by Jeon and Ahn in [9] for automated ticketing in a bank using a user's smart-phone is presented. We analyse that system and then our security analysis shows a number of important security pitfalls.

This paper is structured as follows. In Sect. 2 RNTS system is introduced and its security and privacy risk are analysed in Sect. 3. This paper ends with some conclusions in Sect. 4.

2 RNTS protocol

In [9], the authors proposed a new system to provide privacy and anonymity in customer identification, and efficiency for banking services. RNTS pursues that cus-

Table 1 RNTS notation [9]

BK	A secret key shared between the bank and each customer
UID	An unique discrimination information saved in the tag
CN	A tagged card number identifying the customer
NT	The order-waiting tag transmitted to the customer in the order of entering the bank
PN_M	In a personal device number, corresponds to the middle digits
PN_P	In a personal device number, corresponds to the last four digits
r	A random number
cnt	A counter
V_x	Messages between the bank and the customer
$\mathcal{H}(x)$	Hash function
$\overset{?}{=}$	Compare two values and if they are the same then the protocol continues its execution
$\oplus$	XOR operator
$\|$	Concatenation operator

tomers do not have to wait for a long time—and sorting them on a transparent first-come first-served basis. In the RNTS system, when a customer equipped with a credit card (i.e., RFID tag) enters the bank, the tag is recognized and connected with the bank's queuing system (through the RFID reader). The are two different phases: authentication and ticket issuing. In the next subsections, these two schemes are introduced. In the following, we use the same notation as in the original paper [9] (see Table 1 for details).

2.1 Authentication phase

On the authentication phase when a customer visits the bank, her credit card is recognized by the reader at the entrance. An outline of the RNTS mutual authentication protocol is shown in Fig. 1. The steps carried out are:

1. Reader generates a random number r, and computes the XOR between r and the secret key shared with the bank: $V_1 = BK \oplus r$. It also computes the hash value of r ($\mathcal{H}_1 = \mathcal{H}(r)$) and sends both values $\{V_1 \| \mathcal{H}_1\}$ to the tag.
2. The tag recovers r' with the value of BK stored in its memory and message V_1: $r' = V_1 \oplus BK$. Then, it computes the hash value of r' and compares it with $\mathcal{H}_1$. If both values match, then the tag generates V_2 with its CN value and the random number r': $V_2 = CN \oplus r'$. Finally it computes the hash of CN and r' ($\mathcal{H}_2 = \mathcal{H}(CN \| r')$) and sends $\{V_2, \mathcal{H}_2\}$ to the reader.
3. The reader concatenates r with V_2 and $\mathcal{H}_2$ as transmitted from the tag, and forwards them to the database ($V_2 \| \mathcal{H}_2 \| r$).
4. The database gets CN from V_2 ($CN = V_2 \oplus r$), computes the hash value of $CN \| r$ and compares it with $\mathcal{H}_2$ ($\mathcal{H}_2 \overset{?}{=} \mathcal{H}(CN \| r)$). If both values match, then it generates $\mathcal{H}_3$ with the hash value of BK, UID and r ($\mathcal{H}_3 = \mathcal{H}(BK \| UID \| r)$). Finally, the database sends $\mathcal{H}_3$ to the reader.
5. The reader forwards $\mathcal{H}_3$ to the tag.

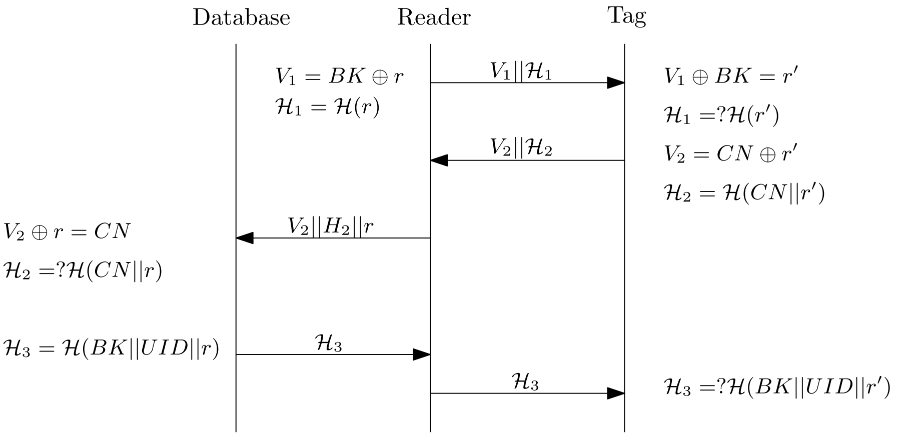

Database
Reader
Tag
$V_1 = BK \oplus r$
$\mathcal{H}_1 = \mathcal{H}(r)$
$V_1||\mathcal{H}_1$
$V_1 \oplus BK = r'$
$\mathcal{H}_1 =?\mathcal{H}(r')$
$V_2||\mathcal{H}_2$
$V_2 = CN \oplus r'$
$\mathcal{H}_2 = \mathcal{H}(CN||r')$
$V_2 \oplus r = CN$
$V_2||H_2||r$
$\mathcal{H}_2 =?\mathcal{H}(CN||r)$
$\mathcal{H}_3 = \mathcal{H}(BK||UID||r)$
$\mathcal{H}_3$
$\mathcal{H}_3$
$\mathcal{H}_3 =?\mathcal{H}(BK||UID||r')$

Fig. 1 Authentication protocol [9]

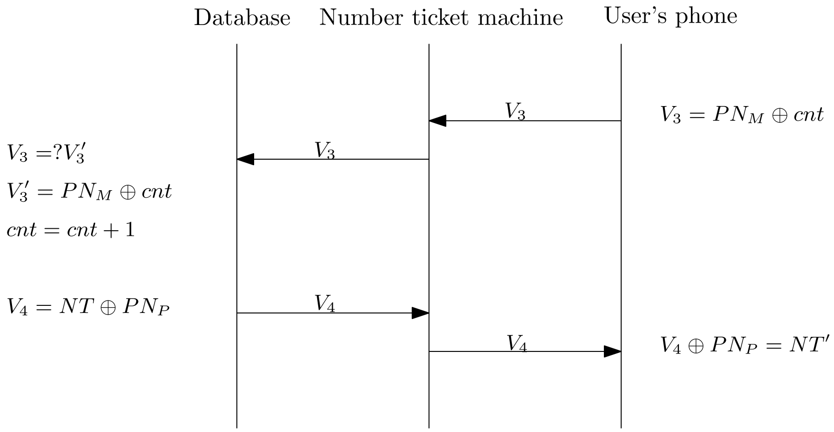


Fig. 2 Ticket issuing protocol [9]

6. The tag receives $\mathcal{H}_3$ that certifies the database and provides mutual authentication by hashing *BK*, *UID* and r': $\mathcal{H}_3 \overset{?}{=} \mathcal{H}(BK\|UID\|r')$.

2.2 Ticket issuing phase

If the customer is recognized at the entrance, then the ticket issuing phase starts. In this phase the customer receives her waiting/queue number from the bank. In Fig. 2 we sketch the RNTS ticket issuing protocol. The steps are described below:

1. User's phone generates a new message V_3 with an XOR operation between PN_M and *cnt* ($V_3 = PN_M \oplus cnt$) and then sends this value to the number ticket machine.
2. The number ticket machine forwards V_3 to the database.

3. The database computes its own version of $V_3^{'} = PN_M \oplus cnt$ corresponding to the previously identified customer, and compares it with the message received. If both match, $V_3 \overset{?}{=} V_3^{'}$ then cnt is incremented by 1 and the database is updated. Finally, a new message V_4 is generated ($V_4 = NT \oplus PN_P$) and sent to the ticket machine.
4. The ticket machine forwards V_4 to the user's phone.
5. The user's phone receives V_4 and gets from it NT', which is the number ticket received by the customer in the bank: $NT = V_4 \oplus PN_P$.

3 Vulnerabilities of RNTS protocol

In this section we describe the main security weakness found on the RNTS system: (1) Full disclosure; (2) Information Leakage; (3) Traceability and (4) Data integrity attacks.

3.1 Full disclosure attack

The attacker can exploit the excessive use of the XOR operation for the generation of messages. Moreover, the use of an incremental counter leaks too much information. An attacker ($\mathcal{A}$) can use both of the above mentioned weaknesses to mount a disclosure attack. More precisely, $\mathcal{A}$ follows the steps described below:

1. In the mth session, $\mathcal{A}$ eavesdrops on the insecure radio channel at the ticket issuing phase (see Fig. 2) capturing message V_3^m:

$$V_3^m = PN_M \oplus cnt^m \tag{1}$$

2. In the next session, $\mathcal{A}$ eavesdrops again at the ticket issuing phase capturing message V_3^{m+1}:

$$V_3^{m+1} = PN_M \oplus cnt^{m+1} \tag{2}$$

Note that after each successful authentication, the counter is incremented by one. That is, $cnt^{m+1} = cnt^m + 1$.

From Eqs. (1) and (2) and performing an XOR operation, $\mathcal{A}$ gets

$$V_3^m \oplus V_3^{m+1} = cnt^m \oplus cnt^{m+1}$$

Thanks to these data, an attacker can obtain PN_M because in two consecutive sessions the XOR operation between V_3^m and V_3^{m+1} modifies only a tiny bit of PN_M. More precisely, we can exploit the fact that $V_3^m \oplus V_3^{m+1}$ is a Mersenne number [8]:

$$V_3^m \oplus V_3^{m+1} = cnt^m \oplus cnt^{m+1} = 2^n - 1 \quad \text{for some integer } n \tag{3}$$

Knowing that $cnt^{m+1} = cnt^m + 1$ (i.e., $cnt^{m+1} > cnt^m$) and using Eq. (3), we know that the nth bit of cnt^m is 0. So, message V_3, transmitted over the insecure radio channel, leaks the nth bit of PN_M.

$$PN_{M_{(n-1)}} = V^m_{3_{(n-1)}} \oplus cnt^m_{(n-1)} = V^m_{3_{(n-1)}} \oplus 0 = V^m_{3_{(n-1)}} \tag{4}$$

Fig. 3 Full disclosure attack (length is 16 bits)

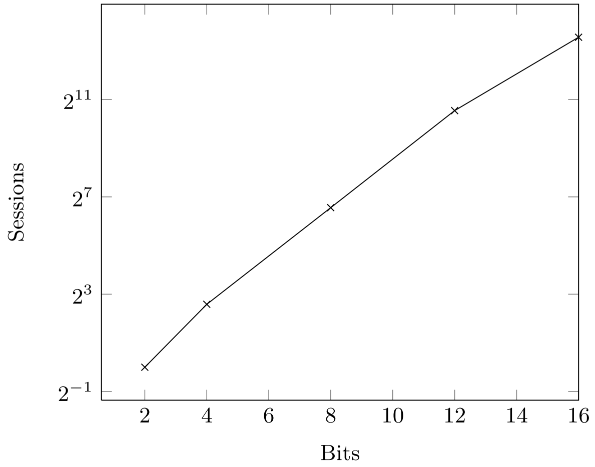
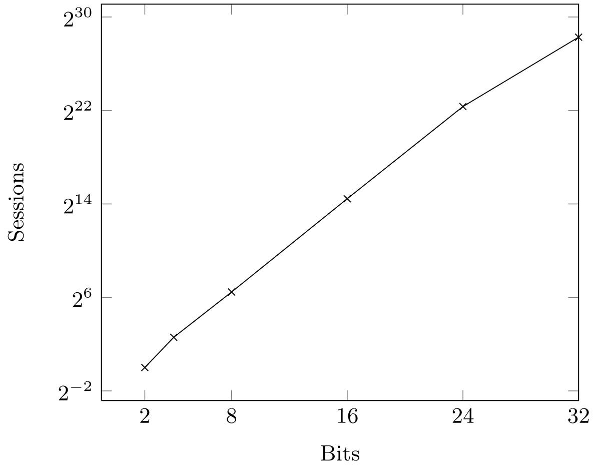


Fig. 4 Full disclosure attack (length of variables sets to 32 bits)



The attacker can repeat the above passive attack to disclose all secret bits of PN_M. As a rule of thumb, the average number of sessions an adversary needs to eavesdrop in order to recover k bits is $2^{\frac{2k-3}{2}}$, which is better (if only slightly) than any brute force approach. Therefore the number of required sessions goes up exponentially (base 2) when the bit length of the secret value PN_M is incremented. Regarding this bit length, the authors do not provide any particular value in the original protocol. In Figs. 3, 4, and without losing generality, we show some simulation results considering 16 and 32 bit length for PN_M.

In the following we provide an example in order to clarify the proposed attack. The bit length of variables is set to 16 for simplicity. We randomly choose[1] values for PN_M and cnt variables (e.g., $PN_M = 374F$ and $cnt = 67A7$).

[1] Corresponding to the first hexadecimal digits of π.

We display V_3^m values eavesdropped by an attacker during six consecutive sessions (i.e. $m = 0, 1, \ldots, 5$):

$$V_3^0 = 374F \oplus 67A7 = 50E8$$
$$V_3^1 = 374F \oplus 67A8 = 50E7$$
$$V_3^2 = 374F \oplus 67A9 = 50E6$$
$$V_3^3 = 374F \oplus 67AA = 50E5$$
$$V_3^4 = 374F \oplus 67AB = 50E4$$
$$V_3^5 = 374F \oplus 67AC = 50E3$$

From Eq. (3), we have

$$n = \log_2\big((V_3 \oplus V_3') + 1\big) \tag{5}$$

Using Eqs. (4) and (5) on the eavesdropped values $\{V_3^m\}_{m=0}^5$, an adversary gets the four least significant bits of PN. We provide the details below:

$$\begin{aligned}
V_3^0 \oplus V_3^1 = 2^4 - 1 \quad &\Rightarrow \quad PN_{M_{(3)}} = V^0_{3_{(3)}} = 1\\
V_3^1 \oplus V_3^2 = 2^1 - 1 \quad &\Rightarrow \quad PN_{M_{(0)}} = V^1_{3_{(0)}} = 1\\
V_3^2 \oplus V_3^3 = 2^2 - 1 \quad &\Rightarrow \quad PN_{M_{(1)}} = V^2_{3_{(1)}} = 1\\
V_3^4 \oplus V_5^1 = 2^3 - 1 \quad &\Rightarrow \quad PN_{M_{(2)}} = V^4_{3_{(2)}} = 1
\end{aligned}$$

So, the passive attack is successful and $\mathcal{A}$ recovers the secret information (i.e., $PN_{M_{(3:0)}} = 1111_2 = F_{16}$).

3.2 Leakage information attack

The attacker can use the information eavesdropped in the ticket issue protocol to disclose several bits of PN_M. Similarly as in Sect. 3.1, the attacker exploits the use of XOR and the incremental counter. Nevertheless, the procedure followed here is completely different. The adversary starts by initializing the list X to null (i.e. $X = \{\}$). Then, $\mathcal{A}$ repeats the following procedure to fill the list. Assuming that $\mathcal{A}$ starts observing the mth session at the ticket issuing phase (see Fig. 2) and the maximum length of list X is K (i.e., $|X| < K$):

1. $\mathcal{A}$ eavesdrops over the insecure radio channel and captures message V_3^m:

$$V_3^m = PN_M \oplus cnt^m \tag{6}$$

2. In the next session, $\mathcal{A}$ eavesdrops again at the ticket issuing phase, and captures message V_3^{m+1}:

$$V_3^{m+1} = PN_M \oplus cnt^{m+1} \tag{7}$$

3. $\mathcal{A}$ updates the list X and increments index m:

$$\begin{aligned} X &= \left\{X, V_3^m \oplus V_3^{m+1}\right\} \\ m &= m + 1 \end{aligned} \tag{8}$$

4. If $|X| < K$, we jump back to Step 1. Otherwise, we abort the algorithm.

Once this phase of the attack is completed, the adversary has a list of K elements:

$$X = \left\{V_3^m \oplus V_3^{m+1}, V_3^{m+1} \oplus V_3^{m+2}, \ldots, V_3^{m+(K-1)} \oplus V_3^{m+K}\right\} \tag{9}$$

At this point, $\mathcal{A}$ finds a family of counters ($c \in C$) that can generate the same list X. That is, each of these counters satisfies

$$X = \left\{c \oplus (c+1), (c+1) \oplus (c+2), \ldots, \big(c + (K-1)\big) \oplus (c+K)\right\} \tag{10}$$

If the length of variables is L bits (i.e., $|PN_M| = L$), $\mathcal{A}$ does a loop for $c = 0$ to N ($N < 2^L$) to check whether c conforms to Eq. (10). If so, the value is stored in a list C. Otherwise, a new value is checked:

$$C = \left\{c, c', \ldots, c''\right\} \quad ! \, C(i) \in X$$

The adversary obtains the four Least Significant Bits (LSBs) of the counter, and discloses the same amount of bits from PN_M, by picking up the minimum (min) value of C:

$$\begin{aligned} cnt'_{(\mathrm{LSB})} &= \min\{C\} \\ PN_{M_{(\mathrm{LSB})}} &= V^0_{3_{(\mathrm{LSB})}} \oplus \min\{C\} = V^0_{3_{(\mathrm{LSB})}} \oplus cnt'_{(\mathrm{LSB})} \end{aligned} \tag{11}$$

The information disclosed in this way is useful to mount straightforward traceability attack [4]. But, even worse, one of the counters in C is the counter used by the tag. That is,

$$\exists c^* \in C \quad ! \, PN_M = V_3^0 \oplus c^* \tag{12}$$

To discover c^* a key-search attack can be run thanks to the fact that the search space is reduced markedly in comparison to the one we should explore in a brute-force attack. In fact, the search space depends on the bit-length of variables ($|PN_M| = L$) and the number of sessions eavesdropped (S). Using the same bit length that in previous analysis, in Table 2 we show an upper bound of the search space for variables of 16 and 32 bits. We can conclude that for a given bit length, the search space quickly goes down when there is an increment in the number of sessions (i.e. exponential decrement).

Table 2 Size of the search space for leakage information

Eavesdropped sessions (S)	Bit length (L)	
	16 bits	32 bits
4	$14536 < 2^{14}$	$65026 < 2^{16}$
8	$7828 < 2^{13}$	$48320 < 2^{16}$
16	$1797 < 2^{11}$	$17872 < 2^{15}$
24	$1425 < 2^{11}$	$13580 < 2^{14}$
32	$851 < 2^{10}$	$11872 < 2^{14}$
64	$502 < 2^{9}$	$6920 < 2^{13}$
128	$160 < 2^{8}$	$4050 < 2^{12}$
256	$88 < 2^{7}$	$1462 < 2^{11}$
512	$35 < 2^{6}$	$704 < 2^{10}$
1024	$18 < 2^{5}$	$474 < 2^{9}$

We sketch an example to facilitate the understanding of the proposed attack. For simplicity, the bit length of variables is set to 16 (i.e. $L = 16$). We randomly choose values for the PN_M and cnt variables (e.g., $PN_M = 374F$ and $cnt = 67A7$).

The adversary eavesdrops values of V_3^m during K consecutive sessions (i.e. $m = m, m+1, \ldots, m+K$):

$$V_3^m = 374F \oplus 67A7 = 50E8$$

$$V_3^{m+1} = 374F \oplus 67A8 = 50E7$$

$$V_3^{m+2} = 374F \oplus 67A9 = 50E6$$

$$V_3^{m+3} = 374F \oplus 67AA = 50E5$$

$$V_3^{m+4} = 374F \oplus 67AB = 50E4$$

$$V_3^{m+5} = 374F \oplus 67AC = 50E3$$

$$V_3^{m+K} = 374F \oplus 67AD = 50E2$$

$\mathcal{A}$ computes the list X:

$$X = \{F, 1, 3, 1, 7\}$$

$\mathcal{A}$ looks for the family of counters $c \in C$ that conforms to Eq. (10).

$$C = \{7_{16}, 17_{16}, 27_{16}, \ldots, 67A7, \ldots, FFE7_{16}, FFF7_{16}\}$$

$\mathcal{A}$ reveals the last four bits of PN_M:

$$PN_{M_{(4-\mathrm{LSB})}} = V^0_{3_{(4-\mathrm{LSB})}} \oplus \min\{C\} = 0x8 \oplus 0x7 = F_{16} = 1111_2$$

3.3 Traceability attack and user anonymity

Traceability is an attack where an attacker can track users carrying an RFID tag, and could make an associations between tags and owners. For a successful attack $\mathcal{A}$ does not need to reveal all secret information stored on tags' memory. In fact, eavesdropping of some data transmitted over the insecure radio channel is generally enough in weak protocols.

We follow the untraceability model proposed by Juels and Weis, and later formalized by Phan in a style commonly used to define security protocol models—the reader is recommended to consult [3, 11] for details. Specifically, $\mathcal{A}$ performs the following steps:

Phase 1 (Learning) $\mathcal{A}$ sends an Execute($\mathcal{R}$, $\mathcal{T}$, m) query. This phase models a passive attacker. $\mathcal{A}$ eavesdrops on the channel, and gets read access to the exchanged messages between $\mathcal{R}$ and $\mathcal{T}$ in session mth of a genuine authentication protocol execution. More precisely, $\mathcal{A}$ acquires the following messages:

$$Z1 = \left\{V_1^{\mathcal{T}_0} = BK^{\mathcal{T}_0} \oplus r, V_2^{\mathcal{T}_0} = CN^{\mathcal{T}_0} \oplus r\right\} \tag{13}$$

Phase 2 (Challenge) $\mathcal{A}$ chooses two fresh tags whose associated identifiers are UID_0 and UID_1. Then she sends a Test($q, \mathcal{T}_0, \mathcal{T}_1$) query. When this query is invoked in session qth, a random bit is generated $b \in \{0, 1\}$. As result, $\{V_1^{\mathcal{T}_b}, V_2^{\mathcal{T}_b}\}$ tuple is given depending on the chosen random bit:

$$Z2 = \begin{cases} \{V_1^{\mathcal{T}_0} = BK^{\mathcal{T}_0} \oplus r^*, V_2^{\mathcal{T}_0} = CN^{\mathcal{T}_0} \oplus r^*\} & \text{if } b = 0 \\ \{V_1^{\mathcal{T}_1} = BK^{\mathcal{T}_1} \oplus r^\dagger, V_2^{\mathcal{T}_1} = CN^{\mathcal{T}_0} \oplus r^\dagger\} & \text{if } b = 1 \end{cases} \tag{14}$$

Phase 3 (Guessing) A outputs a bit d ($d \in \{0, 1\}$) as her guess of the value b. In particular, we propose the following procedure to obtain value d.

- $\mathcal{A}$ computes an XOR between the components of $Z1$:

$$X = V_1^{\mathcal{T}_0} \oplus V_2^{\mathcal{T}_0} = BK^{\mathcal{T}_0} \oplus r \oplus CN^{\mathcal{T}_0} \oplus r \ = BK^{\mathcal{T}_0} \oplus CN^{\mathcal{T}_0} \tag{15}$$

- $\mathcal{A}$ computes an XOR between the components of $Z2$:

$$Y = \begin{cases} V_1^{\mathcal{T}_0} \oplus V_2^{\mathcal{T}_0} = BK^{\mathcal{T}_0} \oplus r^* \oplus CN^{\mathcal{T}_0} \oplus r^* = BK^{\mathcal{T}_0} \oplus CN^{\mathcal{T}_0} & \text{if } b = 0 \\ V_1^{\mathcal{T}_1} \oplus V_2^{\mathcal{T}_1} = BK^{\mathcal{T}_1} \oplus r^\dagger \oplus CN^{\mathcal{T}_1} \oplus r^\dagger = BK^{\mathcal{T}_1} \oplus CN^{\mathcal{T}_1} & \text{if } b = 1 \end{cases} \tag{16}$$

- $\mathcal{A}$ utilizes the following simple decision rule:

$$d = \begin{cases} 0 & \text{if } X = Y \\ 1 & \text{if } X \neq Y \end{cases} \tag{17}$$

According to [3, 11], the advantage of $\mathcal{A}$ in distinguishing whether the adversary interacts with $\mathcal{T}_0$ or $\mathcal{T}_1$ is $Adv_{\mathcal{A}}^{UNT}(t, 1) = |\Pr[d == b] - \frac{1}{2}| = |1 - \frac{1}{2}| = 1/2$. So, the advantage in our case is maximum, and the use of random numbers does not prevent the attacker form associating the tags' answers with its holder. In other words, a passive attacker can track a tag with a success probability of 100 %.

3.4 Data integrity attack

Data integrity guarantees that the content of the message has not been tampered with. This sort of services are needed, especially when messages are transmitted over an insecure channel. The proposal under scrutiny here is open to man-in-the-middle attacks (i.e. replay, stop or modify messages) in the ticket issuing phase. Specifically, the problem lays in that the user's phone does not check the integrity of message V_4. This message is generated by computing a simple XOR operation, and this sort of operations (and triangular functions in general) are particularly vulnerable to active attacks [2]. Therefore, an attacker can easily modify or add a random value (RND) to message V_4 (see Fig. 2 for the protocol description):

$$V_4 = NT \oplus PN_P \oplus RND = (NT \oplus RND) \oplus PN_P = NT^* \oplus PN_P \tag{18}$$

After receiving this message and using its personal number, the user's phone gets a false number ticket:

$$NT^* = V_4 \oplus PN_P = (NT \oplus RND) \neq NT \tag{19}$$

The user cannot detect the attack and will wait her turn indefinitely in the bank. So, the proposed attack can be used to mount a DoS, equivalently to waste all tickets in a ticket machine.

Hash functions are commonly used to protect data integrity. For instance, to avoid the presented attack, the ticket machine could have sent $\{V_4 = NT \oplus PN_P, \mathcal{H}_4 = \mathcal{H}(NT)\}$. This is correct from the theoretical point of view, but the proposal is already very heavy in hardware requirements. Hash functions like SHA-256, SHA-1, MD5 and MD4 require 10,868, 8,120, 8,400 and 7,350 gates equivalents, respectively [6]. Alternatively, a cipher such as AES (i.e. 2,400 GE [16]) demands a footprint that is more tiny and realistic. Specifically, we can use this cipher in Accumulated Block Chaining mode [12] which combines verification of message integrity with encryption.

4 Conclusions

The RNTS system was designed with the useful purpose of providing authentication and a ticket service. Nevertheless, the security properties claimed in the proposal are not satisfied. We show how an adversary can disclose private information, recovering several bits or even the whole user's personal number. Anonymity is also compromised, and bank users can be tracked just by eavesdropping messages exchanged over the insecure radio channel. Exploiting the lack of integrity checking, an adversary can also generate incorrect number tickets, preventing the system from running adequately. Summarizing, we show important security faults in an new RFID hash-based authentication protocol that can be considered useful in a number of realistic scenarios but cannot be deployed in its current form due to these security shortcomings.

References

1. Atzori L, Iera A, Morabito G (2010) The Internet of things: a survey. Comput Netw 54(15):2787–2805
2. Avoine G, Carpent X, Martin B (2010) Strong authentication and strong integrity (SASI) is not that strong. In: Proceedings of RFIDSec, pp 50–64
3. Chien H-Y (2007) SASI a new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE Trans Dependable Secure Comput 4(4):337–340
4. Chien H-Y, Huang C-W (2007) Security of ultra-lightweight RFID authentication protocols and its improvements. Oper Syst Rev 41:83–86
5. Darianian M, Michael MP (2008) Smart home mobile rfid-based Internet-of-things systems and services. In: Proceedings of the 2008 international conference on advanced computer theory and engineering, ICACTE'08, Washington, DC, USA. IEEE Computer Society Press, Los Alamitos, pp 116–120
6. Feldhofer M, Rechberger C (2006) A case against currently used hash functions in RFID protocols. In: Proceedings of OTM. Lecture notes in computer science, vol 4277. Springer, Berlin, pp 372–381
7. Haller S, Karnouskos S, Schroth C (2009) Future Internet—fis 2008. In: The Internet of things in an enterprise context. Springer, Berlin, pp 14–28
8. Hardy GH, Wright EM (1979) An introduction to the theory of numbers, 5th edn. Clarendon Press, Oxford
9. Jeong C, Ahn K (2011) Efficient RNTS system for privacy of banking off-line customer. J Supercomput 55:307–319
10. Juels A (2006) RFID security and privacy: a research survey. IEEE J Sel Areas Commun 24(2):381–394
11. Juels A, Weis SA (2007) Defining strong privacy for RFID. In: Proceedings of PerCom, pp 342–347
12. Knudsen LR (2000) Block chaining modes of operation. Reports in Informatics N0. 207, Department of Informatics, University of Bergen, Norway (ISSN 0333-3590), October 2000
13. Lee K (2010) A two-step mutual authentication protocol based on randomized hash-lock for small RFID networks. In: Proceedings of NSS, September 2010, pp 527–533
14. Michael MP, Darianian M (2008) Architectural solutions for mobile rfid services for the Internet of things. In: Proceedings of the 2008 IEEE congress on services—part I, SERVICES '08, Washington, DC, USA. IEEE Computer Society Press, Los Alamitos, pp 71–74
15. Miorandi D, Sicari S, Pellegrini FD, Chlamtac I (2012) Internet of things: vision, applications and research challenges. Ad Hoc Netw 10(7):1497–1516
16. Moradi A, Poschmann A, Ling S, Paar C, Wang H (2011) Pushing the limits: a very compact and a threshold implementation of AES. In: Proceedings of EUROCRYPT'11, pp 69–88
17. Syamsuddin I, Dillon T, Chang E, Han S (2008) A survey of RFID authentication protocols based on hash-chain method. In: Proceedings of ICCIT, vol 2. IEEE Press, New York, pp 559–564
18. Tan L, Wang N (2010) Future Internet: the Internet of things. In: 3rd international conference on advanced computer theory and engineering (ICACTE), vol 5, pp V5–376–V5–380
19. Weber RH (2010) Internet of things new security and privacy challenges. Comput Law & Secur Rev 26(1):23–30
20. Welbourne E, Battle L, Cole G, Gould K, Rector K, Raymer S, Balazinska M, Borriello G (2009) Building the Internet of things using rfid: the rfid ecosystem experience. IEEE Internet Comput 13(3):48–55
21. Yan T, Wen Q (2011) Building the Internet of things using a mobile rfid security protocol based on information technology. In: Jin D, Lin S (eds) Advances in computer science, intelligent system and environment. Advances in intelligent and soft computing, vol 104. Springer, Berlin, pp 143–149
22. Yeh K-H, Lo N, Winata E (2010) An efficient ultralightweight authentication protocol for RFID systems. In: Proceedings of RFIDSec Asia, pp 49–60