TELKOMNIKA, Vol. 11, No. 2, February 2013, pp. 681~688
ISSN: 2302-4046

Desynchronization Attacks on RFID Security Protocols

Miaolei Deng*[1,3], Weijun Zhu[2,4]
[1]College of Information Science and Engineering, Henan University of Technology,
[2]College of Information Engineering, Zhengzhou University
[3]Lianhua street, Zhengzhou High-tech Zone, [4]100 Science Road, Zhengzhou High-tech Zone
*Corresponding author, e-mail: dmlei2003@163.com

Abstract

The characteristics of radio frequency identification (RFID) systems introduce growing security and privacy concerns. RFID systems need security protocols to provide confidentiality, user privacy, mutual authentication and etc. Many security protocols for the RFID system have been presented. This paper analyze several of the newest RFID security protocols which proposed by Niu et al., Fu et al. and Habibi et al. respectively from the security viewpoint of data desynchronization attack. These lightweight protocols were expected to proposed security protections for the RFID system and safeguard against almost all major attacks. However, we found that these RFID security protocols were vulnerable to the attack of data desynchronization. Based on strand spaces model, data desynchronization attacks on these protocols were analyzed and described. Furthermore, improvements to overcome the security vulnerabilities of two protocols presented by Niu et al. and Fu et al. were given.

***Keywords**: RFID, protocol, security, desynchronization attack*

1. Introduction

Radio frequency identification (RFID) technology with an increasing popularity in manufacturing, retail trade, supply chain management, inventory control, etc. is regarded as the main drive behind the pervasive computing. Because of its low production costs and tiny size, RFID are considered as a replacement technology for barcodes. With the extensive application of this technology, its security and privacy have to be concerned. The main security issues of RFID systems are confidentiality of information, traceability, counterfeit of tags, data desynchronization, etc. Now security issues and privacy problems has become a central concern, which is viewed as a primary barrier to the widespread adoption of RFID technology. Security protocol which is an important RFID security and protection way is the current hot issue of research in this field.

There have been many researches on the security protocol for the RFID system, respectively proposed new ultralightweight authentication protocols for low-cost RFID tags [1] [2]. Kara et al. [3] and Kardas et al. [4] respectively provided novel RFID distance bounding protocols which satisfied the expected security requirements. Avoine et al. [5] introduce a unified framework that aims to improve analysis and design of distance bounding protocols. Cao et al. [6] proposed a universally composable search protocol for RFID, and Zuo et al. [7] gived a set of protocols for secure and private search for tags based on their identities or certain criteria they must satisfy. Eurecom et al. [8] introduced a new protocol for counterfeit detection in RFID-based supply chains through on-site checting. Niu et al. [9] and Fu et al. [10] respectively designed lightweight security protocols for low-cost RFID lately, and they claimed that their protocols achieved in resisting privacy leakage, spoofing and replaying attack etc. However, our security analysis showed that these two RFID protocols were vulnerable to the attacks of data desynchronization. These attacks destroyed the availability of these RFID protocols.

There are several interconnected standards for RFID systems, and among them EPC global has played a major role. The Electronic Product Code Class-1 Generation-2 specification (EPC-C1G2) was announced in 2004 by EPC Global. Recently, Researchers [11-18] have proposed different schemes that complied with these standards. Among them, one of the most recent proposals is a protocol presented by Habibi et al. [18], which has been found by Julio et al. [19] that it was vulnerable to attacks of secret information disclosure, tag impersonation and traceability.

Received August 30, 2012; Revised December 23, 2012; Accepted January 11, 2013

This paper analyzed the attacks of data desynchronization on three RFID protocols which proposed by Niu et al., Fu et al. and Habibi based on the strand spaces model [20-21]. Furthermore, improvements to overcome the security vulnerabilities of protocols proposed by Niu et al. and Fu et al. were presented. The notations in Table 1 are used throughout this paper.

Table 1. Notations

Notation	Interpretation
P	Malicious penetrator
T	The legitimate RFID tag
R	The legitimate RFID reader
B	The legitimate back-end server
D	A database of back-end server
IDT	The static tag-identification number
$IDTA$	An alias
$h()$	One-way hash function
k	Shared symmetric-key between R and B
$E_k()$	Symmetric-key encryption function with the key k
$D_k()$	Symmetric-key decryption function with the key k
K	Shared random secret between T and R (or B)
$DATA$	All application related data of T
RID	The reader identification number
$PRNG$	A 16-bit pseudo-random number generator
EPC_s	The 96 bits of EPC code are divided into six 16-bit blocks, and these blocks are XORed to form EPCs.
P_i	The 16-bit access key stored in T to authenticate B at the $(i+1)^{th}$ phase of authentication
K_i	The 16-bit authentication key stored in T to be authenticated by B at the $(i+1)^{th}$ phase
C_i	The 16-bit index of the record of the i^{th} tag's information in B
P_{old} and P_{new}	The old and new access keys, respectively
K_{old} and K_{new}	The old and new authentication keys, respectively
C_{old} and C_{new}	The old and new indexes for the i^{th} tag, respectively
N_T and N_R	Random numbers generated by T and R respectively
X	The value kept as either *new* or *old* to show which key in the record of B is matched

2. Basal Definition of Strand Spaces Model

In this section, we will introduce the basic ideas of the strand spaces model. In strand spaces model an execution of a protocol includes a set of actions. Send and receive actions are used to represent send message and receive message respectively. For simplicity, 〈send a〉 and 〈receive a〉 are denoted as signed terms 〈+a〉 and 〈-a〉 respectively. Then the set of a finite sequence of signed terms as $(\pm A)^*$ describes the event sequences in the execution of the protocol. A strand is a sequence of transmission and reception events local to a particular run of a principal. If this principal is honest, it is a regular strand. If it is dishonest, it is a penetrator strand.

A bundle C is a causally well-founded directed graph containing the transmission and reception events of a number of strands. It represents a global execution possible for a given protocol. A node m in the graph precedes a node n (written $m \not\Rightarrow n$) if n is accessible from m via 0 or more edges of the graph. Likewise, $m \prec n$ means it is accessible via 1 or more edges.

3. Data Desynchronization Attack on the NIUWU Protocol

3.1. The NIUWU Protocol

Niu et al. proposed a lightweight RFID authentication protocol (NIUWU protocol) in [9]. The steps of their RFID protocol are as following.

1) RFID reader R generates a fresh random nonce r, randomizes it with the one-way hash function, $S=h(r)$. R sends S to the queried tag T. S is used to authenticate the validity of R.

2) When queried, RFID tag T sends M and N to R, where $M = h(IDT||S) \oplus K$ and $N = h(M||S)$. M is to verify the legitimate R.

3) R simply forwards M, N, S and r to B.

4) Firstly, B verifies whether the forwarded r is valid or not by comparing S with $h(r)$. If r is valid, for each tuple (IDT, K) in D, B verifies that $M \oplus K$ equals $h(IDT||h(r))$ and N equals

$h(M||h(r))$. If no tuple is found, the tag is rejected. If *B* successfully finishes the authentication process, *B* generates *C*, where $C=h(K)$. Secondly, *B* encrypts the corresponding DATA using the key *k*, then replies *C* and E_k(DATA). Finally, *B* makes its shared secret, *K*, randomized simply by Xoring with *C*.

5) *R* forwards *C* to *T*. *T* verifies the forwarded *C*, calculates $h(K)$ and compares it with *C*. If matched, the mutual authentication is finally succeeded, and *T* updates the shared secret *K*. Otherwise, *T* will not update it.

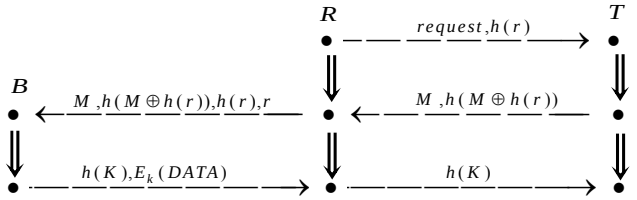

Figure 1. The bundle C_{NIU} of the NIUWU protocol

3.2. The Data Desynchronization Attack on the NIUWU Protocol

In [9], Niu et al. deemed that t5heir proposed RFID protocol could guarantee data confidentiality, tag anonymity, data integrity, forgery resistance, and mutual authentication, and resist to replay attack. However, the NIUWU protocol can not offer any protection against data desynchronization attack: a penetrator *P* can easily force an honest tag to fall out of synchronization with the reader so that it can no longer authenticate itself successfully. The data desynchronization attack on the NIUWU protocol can be described in Figure 2.

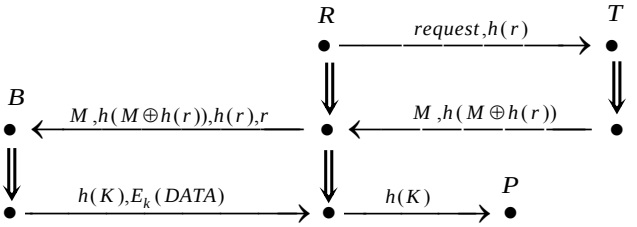

Figure 2. The data desynchronization attack on the NIUWU protocol

In the attack, the penetrator *P* easily destroys the synchronization of the *K* updating between the tag *T* and the back-end server *B*. *P* can intercept the message $h(k)$ form *R* to *T*. Therefore, *B* has refreshed the secret *K* while *T* will not do it. Thus the shared random secret between *B* and *T* may not be the same. After a successful data desynchronization attack, because *P* makes *B* and *T* share the different secrets, *R* (and *B*) will not be authorized by *T* and *T* will not be authorized by *R* (and *B*) yet. Thus, the availability of the NIUWU protocol is destroyed.

3.3. Improvement of the NIUWU Protocol

To solve the data desynchronization attack problem described above, *B* ought to keep the history of the entire shared secret update. That is, *B* will keep the current records and the previous records of the secret update process. While *B* (and *R*) fails to authenticate a tag because of the data desynchronization attack, it recovers the old shared secret from the previous secret update record to complete the authentication.

4. Data Desynchronization Attack on the FUWU Protocol

4.1. The FUWU Protocol

In [10], Fu et al. designed an RFID security protocol (FUWU protocol). In their protocol, each tag shares the static tag-identification number *IDT* and the secret *K* with the reader. Each

tag has an *IDTA* which is an alias, and the reader has a symmetric-key encryption function $E_k()$ where k is known only by it. *IDTA* is encrypted and decrypted by R with $E_k()$. In each execution *IDTA* is updated as $IDTA = E_k(IDT||r)$, where r is a random number generated by the reader. The steps of the FUWU protocol can be described as follows.

1) R generates a random number r_1, and sends it to the queried tag T.

2) When queried, T generates a random number r_2, computes $h(K||r_1)$ and sends *IDTA*, $h(K||r_1)$, r_2 to R.

3) Firstly, R decrypts *IDTA* to get *IDT* of T, and then retrieves the shared secret K between R and T by *IDT*. Secondly, R computes $h(K||r_1)$ and verifies whether the computed value equals to the received one or not. If it matches, T is authenticated. Otherwise the authentication failed. If T is authenticated successfully, R generates a new random number r', computes $IDTA'$ as $IDTA' = E_k(IDT||r')$ and stores $IDTA'$ as the new alias. Then R computes the values A and B as $A = IDTA' \oplus h(K||r_1||r_2)$, $B = IDTA' \oplus h(K||r_2||r_1)$. Finally, R computes $h(K||r_2)$ and sends $h(K||r_2)$, $IDTA' \oplus h(K||r_1||r_2)$, $IDTA' \oplus h(K||r_2||r_1)$ to T.

4) T verifies $h(K||r_2)$ to authenticate R. If it matches, R is authenticated, otherwise the authentication failed. If R is authenticated successfully, T computes $h(K||r_1||r_2)$ and $h(K||r_2||r_1)$. Then T computes $IDTA_1 = A \oplus h(K||r_1||r_2)$ and $IDTA_2 = B \oplus h(K||r_2||r_1)$. If $IDTA_1 = IDTA_2$, T stores $IDTA_1$ (= $IDTA'$) as the new alias and sends OK to R.

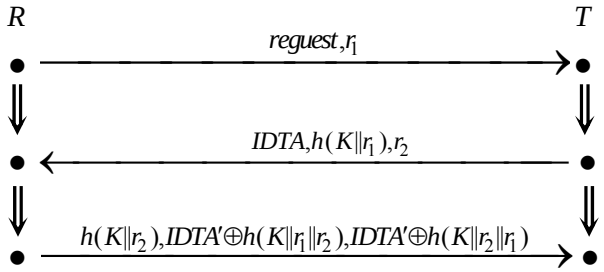

Figure 3. The bundle C_{FU} of the FUWU protocol

4.2. The Data Desynchronization Attack on the FUWU Protocol

The FUWU protocol can't resist the data desynchronization attack. The process of the attack is shown as follows. Firstly, P can eavesdrop a valid session between R and T. Secondly, P lets R and T send the first and the second message normally, but it modifies the third message. $IDTA' \oplus h(K||r_1||r_2)$ is modified to $IDTA'' \oplus IDTA' \oplus h(K||r_1||r_2)$ and $IDTA' \oplus h(K||r_2||r_1)$ to $IDTA'' \oplus IDTA' \oplus h(K||r_2||r_1)$, where $IDTA''$ is an arbitrary random number. Then P sends $h(K||r_2)$, $IDTA'' \oplus IDTA' \oplus h(K||r_1||r_2)$, $IDTA'' \oplus h(K||r_2||r_1)$ to T. Finally, T verifies $h(K||r_2)$ and regards P as R. T also computes $IDTA_1 = IDTA'' \oplus IDTA' \oplus h(K||r_1||r_2) \oplus h(K||r_1||r_2) = IDTA'' \oplus IDTA'$ and $IDTA_2 = IDTA'' \oplus h(K||r_2||r_1) \oplus h(K||r_2||r_1) = IDTA'' \oplus IDTA'$. For $IDTA_1 = IDTA_2$, T stores $IDTA'' \oplus IDTA'$ as the new alias and sends OK to R. Thus P makes R and T share the different alias, T will not be authorized by R in the next sessions. The data desynchronization attack on the FUWU protocol can be described based on the strand spaces model in Figure 4.

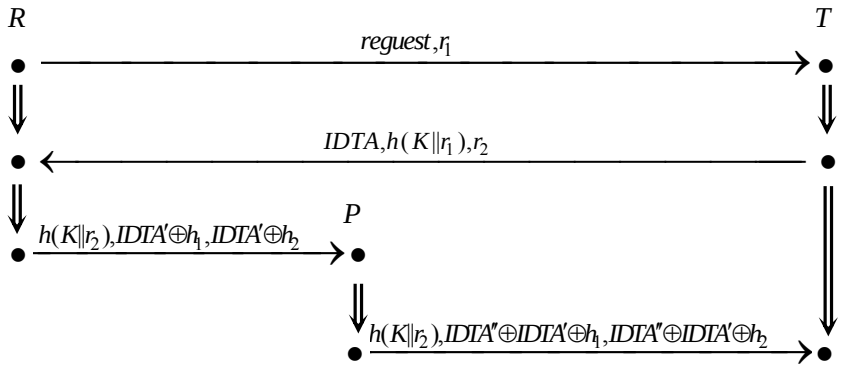

Figure 4. The data desynchronization attack on the NIUWU protocol (where $h_1=h(K||r_1||r_2)$ and $h_2=h(K||r_2||r_1)$)

4.3. Improvement of the FUWU Protocol

For the above flaw, we give the following improvements to the FUWU protocol to resist data desynchronization attacks. In step 3, *R* also computes $h(K||IDTA')$ beside *A* ($=IDTA'\oplus h(K||r_1||r_2)$), *B* ($=IDTA'\oplus h(K||r_2||r_1)$) and $h(K||r_2)$. Finally, *R* sends $h(K||r_2)$, $IDTA'\oplus h(K||r_1||r_2)$, $IDTA'\oplus h(K||r_2||r_1)$ and $h(K||IDTA')$ to *T*. In step 4, after affirming of $IDTA_1=IDTA_2$, *T* also computes $h(K||IDTA_1)$ and checks whether the computed value equals to the received $h(K||IDTA')$. If it matches, *T* just stores $IDTA_1$ as the new alias and sends OK to *R*. The improved bundle C'_{FU} of the FUWU protocol is shown in Figure 5.

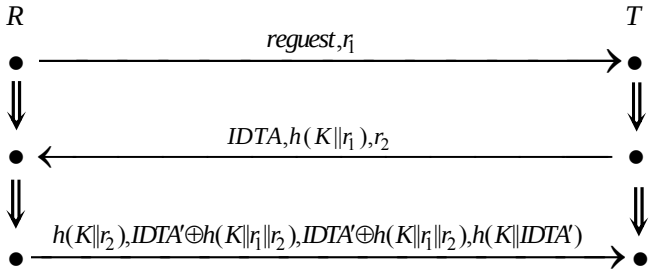

Figure 5. The improved bundle C'_{FU}

5. Data desynchronization attack on the HAA protocol

5.1. The HAA Protocol

In [18], Habibi et al. proposed an EPC-compliant scheme (HAA protocol). Their protocol contains two phases: an initialization phase and an $(i+1)^{th}$ authentication phase. In the initialization phase, the manufacturer generates random values for K_0, P_0 and C_0 respectively and sets the values of the record in the tag, i.e., $K_i = K_0$, $P_i = P_0$, $C_i = C_0$ and the corresponding record in the back-end server $K_{old} = K_{new} = K_0$, $P_{old} = P_{new} = P_0$, $C_{old} = C_{new} = 0$.

The authentication phase (in its $(i+1)^{th}$ run) is described as follows.

1) The reader *R* generates a random number N_R and sends it to the tag *T*.

2) *T* receives N_R, generates a random number N_T, computes M_1, *Q*, *E* and finally sends M_1, *Q*, *E* and C_i to *R*, where $M_1 = PRNG(EPC_s\oplus N_R\oplus N_T)\oplus K_i$ and $Q = N_T\oplus K_i$ and $E = N_T\oplus PRNG(C_i\oplus K_i)$.

3) When *R* receives the message, it computes $V = h(RID\oplus N_R)$ and forwards M_1, *Q*, C_i, *E*, N_R, *V* to the back-end server *B*.

4) After B receiving M_1, Q, C_i, E, N_R, and V, it proceeds as follows.
- For each RID stored in the database D, it computes $h(RID \| N_R)$ and compares it with the received V to verifies R legitimacy.
- If $C_i = 0$, which means that it is the first access to the tag, it proceeds as follows, iteratively: (a) Picks up an entry (K_{old}, P_{old}, C_{old}, K_{new}, P_{new}, C_{new}, RID, EPC_s, $DATA$) stored in database. (b) Verifies whether $M_1 \| K_{old} = PRNG(EPC_s \oplus N_R \oplus Q \oplus K_{old})$ or $M_1 \| K_{new} = PRNG(EPC_s \oplus N_R \oplus Q \oplus K_{new})$, and marks X as *old* or *new* provided that the verification process is satisfied based on the new record or the old record.
- Otherwise, B uses C_i as an index to find the corresponding record in the database and verify whether $PRNG(EPC_s \oplus N_R \oplus Q \oplus K_X) \| K_X = M_1$. If "No" the protocol aborts.
- Verify whether $N_T \| PRNG(C_i \oplus K_X) = E$. If "No" the protocol aborts.
- Computes M_2 and $Info$ and forwards them to R, where $M_2 = PRNG(EPC_s \oplus N_T) \| P_X$ and $Info = DATA \| RID$.
- If $X = new$, updates the database as follows: $K_{old} \leftarrow K_{new}$, $K_{new} \leftarrow PRNG(K_{new})$, $P_{old} \leftarrow P_{new}$, $P_{new} \leftarrow PRNG(P_{new})$, $C_{old} \leftarrow C_{new}$, $C_{new} \leftarrow PRNG(N_T \oplus N_R)$.
- Else, $C_{new} \leftarrow PRNG(N_T \oplus N_R)$.

5) Once R receives the message, it extracts $DATA$ as $Info \oplus RID$ and forwards M_2 to T.

6) When T receives the message, it verifies whether $PRNG(EPC_s \oplus N_T) = M_2 \oplus P_i$. If "No" the protocol aborts. Else T authenticates B and updates the contents kept inside as $K_{i+1} \leftarrow PRNG(K_i)$, $P_{i+1} \leftarrow PRNG(P_i)$, $C_{i+1} \leftarrow PRNG(N_T \oplus N_R)$. Figure 6 gives the bundle C_{HAA} of the HAA protocol.

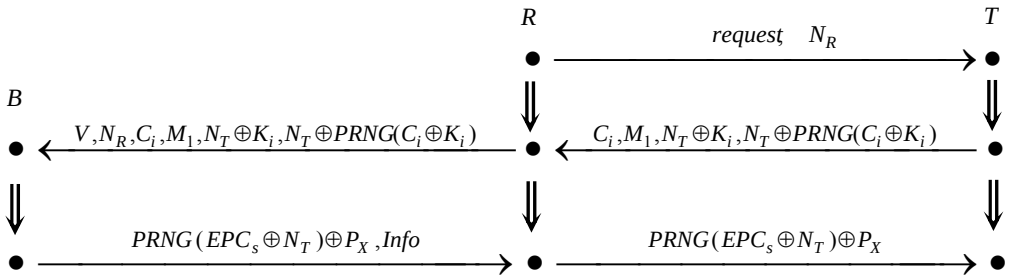

Figure 6. The bundle C_{HAA} of the HAA protocol (where $V = h(RID \oplus N_R)$ and $M_1 = PRNG(EPC_s \oplus N_R \oplus N_T) \oplus K_i$)

5.2. The Data Desynchronization Attack on the HAA Protocol

The HAA protocol can't resist the data desynchronization attack either. Before the implementation of the data desynchronization attack, P needs to carry out a secret information disclosure attack. Julio et al. [19] presented an efficient secret information disclosure attack on the HAA protocol. The Julio's information disclosure attack can be described as follows.

1) P eavesdrops one session of the HAA protocol and stores all the exchanged messages: N_R, C_i, $M_1 = PRNG(EPC_s \oplus N_R \oplus N_T) \oplus K_i$, $Q = N_T \oplus K_i$, $E = N_T \oplus PRNG(C_i \oplus K_i)$ and $M_2 = PRNG(EPC_s \oplus N_T) \| P_X$.

2) $\forall\ i = 0, ..., 2^{16}-1$ does as follows: $K_i \leftarrow i$ and $N_T \leftarrow Q \| K_i$, If $E = N_T \| PRNG(C_i \| K_i)$ then return K_i and N_T.

3) For the returned values of K_i and N_T and $\forall\ i = 0, ..., 2^{16}-1$ does as follows: $EPC_s \leftarrow i$, If $M_1 = PRNG(EPC_s \oplus N_R \oplus N_T) \oplus K_i$ then return EPC_s.

4) For the returned values of K_i and N_T from Step 2 and EPC_s from Step 3 assigns $M_2 \| PRNG(EPC_s \oplus N_T)$ to P_i and returns the following values: $P_{old} = P_i$, $P_{new} = PRNG(P_i)$, $K_{old} = K_i$, $K_{new} = PRNG(K_i)$, $C_{old} = C_i$.

Thus P can disclose all the secret parameters of T, including EPC_s, K_i and P_i. Then P can easily launch the data desynchronization attack. The process of the data desynchronization

attack is shown as follows. Firstly, *P* launches the secret information disclosure attack and retrieves any secret information in *T*, including EPC_s, K_i and P_i. Secondly, *P* eavesdrops the random number N_R generated by *R* and values C_i, M_1, *Q*, *E* generated by *T* in the following protocol run, and it intercepts the message C_i, M_1, *Q*, *E* from the tag to the reader. Thirdly, *P* Computes $N_T = Q \oplus K_i$, $M_2 = PRNG(EPC_s \oplus N_T) \square P_i$, and forwards M_2 to *T*. Once *T* receives M_2, it authenticates *B* and updates the contents kept inside as $K_{i+1} \leftarrow PRNG(K_i)$, $P_{i+1} \leftarrow PRNG(P_i)$, $C_{i+1} \leftarrow PRNG(N_T \oplus N_R)$. Therefore, the tag has refreshed the secret K_i, P_i, C_i while the back-end server will not do it. Thus, the shared secret between the tag and the back-end server may not be the same, which can bring system to a mess. After a successful data desynchronization attack, because *P* makes *B* and the valid tag *T* share the different secrets, *B* will not be authorized by *T* and *T* will not be authorized by *B* yet.

The data desynchronization attack on the HAA protocol can be described based on the strand spaces model in Figure 7.

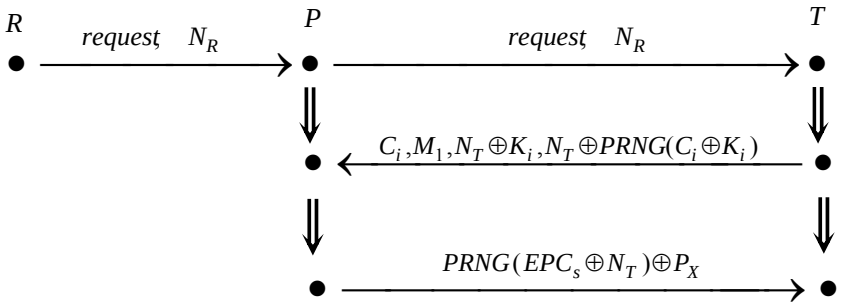

Figure 7. The data desynchronization attack on the HAA protocol (where $M_1 = PRNG(EPC_s \oplus N_R \oplus N_T) \oplus K_i$)

6. Conclusion

In a data desynchronization attack, the penetrator forces the RFID tag and the RFID reader (or the back-end server) to update their common values to different values. If the penetrator can succeed in forcing the tag and the reader (or the back-end server) to do so, the tag will not be authenticated in future transactions. This destroys the availability of RFID security protocols. This paper discusses data desynchronization attacks on some RFID protocols which proposed recently by Niu et al., Fu et al. and Habibi et al. in the strand spaces model. We found that both the lightweight protocol proposed by Niu et al. and the scalable protocol proposed by Fu et al. were vulnerable to data desynchronization attacks. In addition, improvements to overcome the vulnerabilities of these two protocols were given. For the HAA protocol − an EPC-compliant scheme proposed by Habibi et al., we presented an efficient data desynchronization attack which based on a passive secret information disclosure attack.

References

[1] Lee Yung-Cheng. Two Ultralightweight Authentication Protocols for Low-Cost RFID Tags. *Applied Mathematics & Information Sciences*. 2012; 6(2S): 425S-431S.

[2] Tian Y, Chen G, Li J. A New Ultralight weight RFID Authentication Protocol with Permutation. *IEEE Communications Letters*. 2012; 16(5): 701-705.

[3] Kara O, Kardas S, Bingol MA, Avoine G. *Optimal Security Limits of RFID Distance Bounding Protocols*. Proceedings of the 6th International Workshop on RFID Security- RFIDSec'10. Istanbul. 2010: 220-238.

[4] Kardas S, Kiraz MS, Binglo MA, Demirci H. *A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions*. Proceedings of the 7th International Workshop on RFID Security-RFIDSec'11. Amherst. 2011: 78-93.

[5] Avoine G, Bingol MA, Kardas S, Lauradoux C, Martin B. A framework for Analyzing RFID Distance Bounding Protocols. *Journal of Computer Security*. 2011; 19(2): 289-317.
[6] Cao Z, Deng ML. Universally Composable Search Protocol for RFID. *Journal of Huazhong University of Science and Technology*. 2011; 39(4): 56-59.
[7] Zuo YJ. Secure and Private Search Protocols for RFID Systems. *Information Systems Frontiers*. 12(5): 507-519.
[8] Eurecom KE, Blass EO, Molva R. *CHECKER: On-site Checking in RFID-based Supply Chains*. WiSec'12. Arizona. 2012: 80-92.
[9] Niu ZH, Wu XH. A New Lightweight Authentication Protocol for the RFID System. *Chinese Journal of Engineering Mathematics*. 2012; 27(5): 939-942.
[10] Fu J, Wu C, Chen X, Fan R, Ping L. *Scalable pseudo random RFID Private Mutual Authenticaiton*. Proceedings of the 2nd IEEE International Conference on Computer Engineering and Technology. Chengdu. 2010: 497-500.
[11] Chen CL, Chien CF. An Ownership Transfer Scheme Using Mobile RFIDs. *Wireless Personal Communications*. 2012: 1-27.
[12] Doss R, Zhou W, Sundaresan S, Yu S, Gao L. A Minimum Disclosure Approach to Authentication and Privacy in RFID Systems. *Computer Networks*. 2012; 56(15): 3401-3416.
[13] Joan MS, Joaquin GA, Jordi HJ. A Practical Implementation Attack on Weak Pseudorandom Number Generator Designs for EPC Gen2 Tags. *Wireless Personal Communications*. 2011; 59(1): 27-42.
[14] Pedro PL, Julio CH, Juan MET, Jan CAL. Cryptanalysis of an EPC Class-1 Generation-2 Standard Compliant Authentication Protocol. *Engineering Applications of Artificial Intelligence*. 2011; 24: 1061-1069.
[15] Safkhani M, Bagheri N, Naderi M. *Cryptanalysis of AZUMI: an EPC Class-1 Generation-2 Standard Compliant RFID Authentication Protocol*. Proceedings of IACR Cryptology ePrint Archive. 2011: 1-7.
[16] Yoon EJ. Improvement of the Securing RFID Systems Conforming to EPC Class 1 Generation 2 standard. *Expert Systems with Applications*. 2012; 39: 1589-1594.
[17] Moessner M, Khan GN. Secure Authentication Scheme for Passive C1G2 RFID Tags. *Computer Networks*. 2012; 56: 273-286.
[18] Habibi MH, Alagheband MR, Aref MR. *Attacks on a Lightweight Mutual Authentication Protocol under EPC C-1 G-2 Standard*. Proceedings of WISTP. 2011; 6633: 254-263.
[19] Julio CHC, Pedro PL, Masoumeh S, Nasour B, Majid N. *Another Fallen Hash-Based RFID Authentication Protocol*. Proceedings of WISTP. 2012; 7322: 29-37.
[20] Thayer F, Herzog JC, Guttman JD. Strand Spaces: Proving Security Protocols Correct. *Journal of Computer Security*. 1999; 7(2): 191-230.
[21] Guttman JD. Authentication Tests and Disjoint Encryption: A Design Method for Security Protocols. *Journal of Computer Security*. 2004; 12(3): 409-433.