Algebraic Replay Attacks on Authentication in RFID Protocols

Noureddine Chikouche[1], Foudil Cherif[2], and Mohamed Benmohammed[3]

[1] Department of Computer Science, University of M'sila, Algeria
[2] LESIA Laboratory, University of Biskra, Algeria
[3] LIRE Laboratory, University of Constantine, Algeria

Abstract. One of the most important challenges related to RFID systems is the verification of security proprieties in RFID authentication protocols. Among the important attacks in RFID systems, we speak about the Algebraic Replay Attack on Authentication (ARA). Common characteristic between the verified protocols cannot resist algebraic replay attacks. Our work is articulated on the formal automatic verification of RFID protocols by two different tools, firstly, the Open-source Fixedpoint Model Checker (OFMC) tool, secondary, the Constraint Logic based Attack Searcher (Cl-Atse) tool. These tools sufficient for detecting the attach of type ARA.

1 Introduction

The radiofrequency identification (RFID) systems are steadily becoming paramount due to their vast applications such as supply chain management, mobile phone, health, automated payment systems, e-passport, access control etc. A typical RFID system consists of three entities: (1) the tag (or the label), a small electronic device, supplemented with an antenna that can transmit and receive data, (2) the reader, a device to read and write RFID tags by radio waves and (3) the backend system (or database, server), a centralized place that hosts all data regarding access permissions and which may be consulted by the reader.

One of the most important challenges related to RFID systems is security. The communication channel between the server and the reader is assumed to be secure while the wireless channel between the reader and the tag is assumed to be insecure since it makes it opened to logical attacks on authentication protocol. Among attacks studied in the last years by researchers, we quotes algebraic replay attacks (ARA). The main cause of these attacks is the abuse of the algebraic operator properties employed by the protocols. The operator or-exclusive (xor) is algebraic operator. This operation is used in many RFID protocols and has aroused a lot of interest during the last years; its implementation is low cost and requires some logical gates.

The phases of design and implementation of RFID authentication are important, but the phase of verification of the protocol is very important. To validate the security proprieties (secrecy, authentication, integrity,...etc) of authentication protocol, we use a formal tool of verification. There are several tools of automatic verification of

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 153–163, 2013.

cryptographic protocols. We chose OFMC [1, 2] (Open-source Fixedpoint Model Checker) tool and the CL-Atse (Constraint Logic based Attack Searcher) [3] for the following reasons: they are based on the same specification languages: HLPSL language [4] and AnB language [5]. These tools are the analyzer which models a big number of security protocols (more than 90 protocols). These tools are available using various techniques of validation (Model-checking, Horn Clause, resolution of constraints, rewriting technique).

Our Contribution is articulated around the verification of RFID authentication protocols by using the OFMC and CL-Atse tools after specifying these protocols in specification language. These analyses are based on the automatic verification of three security proprieties: secrecy, tag authentication and server authentication. The verified protocols require one-way function, xor-operator and pseudo-random number generator (PRNG).We prove which of the presented protocols cannot resist algebraic replay attacks.

The rest of this paper is structured as follow: Section 2 presents the specification language, the verification tools and the intruder model. Section 3 presents a RFID authentication protocols. In section 4, we show the verification results and we discuss these results in section 5. Finally, the paper is finished by a conclusion.

2 Formal Automatic Verification

The formal automatic verification of RFID protocols involves the following steps:

- Specification: specification the initial assumptions, the capacity of intruder, the protocol goals (secrecy, authentication, etc.), the roles (the tag and reader), the messages transmitted and the primitives (hash function, PRNG, xor-operator, concatenation, etc.),
- Verification: After verifying the protocol using a validation tool, it is confirmed that the protocol is either safe or it has failed. In case of failure, the tool presents message transmitted between an intruder, a reader and a tag, i.e. describe the trace of attack.

2.1 Intruder Model

Besides modelling security protocols, it is also necessary to model the intruder, that is to say, to define its behaviour and limit. For this, we assume an active Dolev-Yao attacker [6]. This intruder model is based on two important assumptions that are the *perfect encryption* and the *intruder is the network*.

Perfect encryption ensures in particular that: *(1)* an intruder can decrypt a message m encrypted with key k if it has the opposite of that key, *(2)* a key cannot be guessed (during the period of its validity), *(3)* and Given m, it is not possible to find the corresponding ciphertext for any message containing m without knowledge of the key.

The intruder is the network: the intruder has complete control over the network, i.e. it can impersonate a tag, impersonate a reader, obtain any message passing through the network, block or modify messages and it can also derive new ones messages from its initial knowledge and the messages that are received from honest principals during protocol run. The communication between the tag and reader is not assured and based on radio frequencies waves. In this paper, our particular verification gets transmissions on the canal reader - tag only.

For the authentication protocols required or-exclusive operator, other important assumption, an intruder that can exploit the algebraic properties of the XOR operator, which are:

$x \oplus 0 \rightarrow x$	(neutral element)	(1)
$x \oplus x \rightarrow 0$	(nilpotence)	(2)
$x \oplus y \rightarrow y \oplus x$	(commutativity)	(3)
$x \oplus (y \oplus z) \rightarrow (x \oplus y) \oplus z$	(associativity)	(4)

2.2 Protocol Goals

Security proprieties, such as: secrecy, tag authentication, and reader authentication.

- Secrecy: or confidentiality, the verification of secret data so that they are never passed on clearly to air on the radio frequency interface which can be spied on.
- Tag authentication: A reader has to be capable of verifying a correct tag to authenticate and to identify this tag in complete safety.
- Reader authentication: A tag has to be capable of confirming that it communicates with the legitimate reader (we assume the communication between the server and reader is assured).

2.3 Specification

In this paper, we use two specification languages, HLPSL [4] and AnB [5]. These languages are the input languages of the OFMC and the CL-Atse verification tools. Alice and Bob (AnB) notation is a high-level and straight-forward language for describing security protocols. It describes how messages are exchanged between honest agents acting in the different protocol roles. The novel features of AnB are its support for protocols that require algebraic properties for the protocol execution, as well as a notion of several types of communication channels that can be used both as assumptions and as goals of a protocol.

High Level Protocol Specification Language (HLPSL) is a modular, expressive, formal, role-based language. Protocol specification consists of two types of roles, basic roles and composed roles. Basic roles serve to describe the actions of one single agent in the run of the protocol. Others instantiate basic roles to model an entire protocol run, a session of the protocol between multiple agents, or the protocol model itself.

2.4 Verification Tools

The OFMC and CL-Atse are developed in the framework of the AVISPA European Project[1] and the AVANTSSAR European Project[2]. These tools can verify the protocols requiring the operator exclusive or (XOR). The first tool, The Open-source Fixedpoint Model Checker (which extends the on the-Fly Model Checker, the previous OFMC) [1, 2] consists of two modules. The classical module performs verification for a bounded number of transitions of honest agents using a constraint-based representation of the intruder behavior. The fixed point module allows verification without restricting the number of steps by working on an over-approximation of the search space that is specified by a set of Horn clauses using abstract interpretation techniques and counterexample-based refinement of abstractions. Running both modules in parallel, OFMC stops as soon as the classic module has found an attack or the fixed point module has verified the specification, so as soon as there is a definitive result.

The second tool, CL-AtSe [3] is a Constraint Logic based Attack Searcher for the security protocols and services takes as an input a service specified as a set of rewriting rules, and applies rewriting and constraint solving techniques to model all states that are reachable by the participants and decides if an attack exists with respect to the Dolev-Yao intruder.

3 RFID Authentication Protocols

In this section, we describe an authentication protocols in RFID system, the common characteristic between these protocols: (i) they use or-exclusive operator and one-way function in transmitted messages and (ii) the vulnerabilities of these protocols are of type algebraic replay attacks on authentication (ARA).

To describe informally many RFID authentication protocols, we afterward, use the following notations:

T	RFID tag or transponder
R	RFID reader or transceiver
H	One-way hash function
‖	Concatenation of two inputs
ID	The unique identifier of a tag
⊕	Or-exclusif
RID	The unique identifier of a reader
S,x,y	Secret value
RH	Right-half of the message
LH	Left-half of the message
Nr,Nt, N_{db}	Random number
CRC	Cyclic Redundancy Check

[1] http://www.avispa-project.org
[2] http://www.avantssar.eu

Table 1. RFID Authentication Protocols

PROTOCOL	**Auth_Tag**	**Auth_Reader**	**Secret Data**	**α**	***f***
LAK [7]	H(Nr⊕Nt⊕K)	H(H(Nr⊕Nt⊕K)⊕K⊕Nr)	K	K	H
CH [8]	LH(RH(ID)⊕h(Nr ⊕Nt⊕K))	RH(RH(ID)⊕h(Nr⊕Nt⊕K))	ID, K	K	H
YL [9]	x⊕h(h(K)⊕nt), h(y⊕Nr⊕ Nt)	y^*⊕h(x^*⊕y), h(x^*⊕y^*)	h(k),y,x,K	y	H
QYY [10]	CRC(ID⊕Nt⊕Nr), CRC(ID⊕ Nt⊕Nr) ⊕ x	CRC(ID⊕Nt), CRC(ID⊕Nt) ⊕ x	ID	ID	CRC
WHC [11]	H(Nr⊕Nt⊕S)	H(ID⊕N_{db})	S, ID	S	H

We can describe the transmitted messages in RFID mutual authentication protocols in form:

```
R → T : Nr
T → R : Nt, Auth_Tag
R → T : Auth_Reader
```

The transmitted messages of `Auth_Tag` and `Auth_Reader` are presented in table 1. The `Auth_Tag` comprises of $f(\alpha \oplus N_t \oplus N_r)$, with α is secret data shared between the tag and reader and f is one-way function such as hash function and CRC function. The following is a detailed description of each step of these protocols:

- The reader RFID produces a nonce Nr and sends it and a request to the tag.
- After receiving Nr, a tag generates a random number Nt and computes the function Auth_Tag, then sends Nt. The Auth_Tag is back to the reader (server).
- After receiving authentication message from the tag, the reader would search whether there exists certain ά in table α of the database, which could make f(ά ⊕Nt⊕Nr)= f(α⊕Nt⊕Nr). If it is found, the tag crosses the authentication of the tag and is considered as legitimate, and then the reader calculates Auth_Reader, then sends Auth_Reader to the tag.
- The tag computes Auth_Reader‘, If the outcome equals to the received Auth_Reader, the authentication of the reader is successful, otherwise, the authentication has failed.

Our paper verifies five protocols, as following:

- LAK [7]: Lee et al. propose an authentication protocol. The reader R and tag T share secrets k. at finish authentication, reader and tag updates k to h(k).
- CH [8]: The CH protocol is proposed by Chien and Huang in 2008. It uses hash function and primitives non-cryptographic (Left, Right and Rotate). It uses these primitives for increase the security of protocols.

- YL [9]: The author Yanfei Liu provides a detailed security analysis of the protocol and claims that YL achieves a list of security properties, including resistance to tag impersonating, denial of service, replay and compromising attacks.
- QYY [10]: The authors of this protocol claim that this protocol is secure because of the use CRC (Cyclic Redundancy Check) and uses random nonces to encrypt messages.
- WHC: Wei et al. [11] proposed an authentication protocol (WHC protocol) in 2011. The server and tag share secrets value S and Identifier ID, this protocol proposed for application RFID-Mobile.

4 Results of Verification

This section is articulated around the verification of LAK protocol (as an example) by using CL-Atse and OFMC tools after having specified this protocol in HLPSL and AnB languages respectively.

4.1 OFMC Result

OFMC tool detects the trace of attack on RFID tag authentication (see Fig. 1 (a)). In this trace result, `i` represents the intruder, `(x501, 1)` the reader (server), and `(x502,1)` the tag. The posted information such as: `NR(1)` is the instance of the nonce `NR`, `X2624` and which is a variables related to the internal workings of the OFMC tool (in this trace is the instance of the nonce `NR`), `NT(2)` is the instance of the nonce `Nt` and `sk(x502,x501)` is a symmetric key `K`.
We symbolize: `NR(1)` by `Nr, X2624` by `Nr'`, `NT(2)` by `Nt and sk(x502,x501)` by `K`. We Summarizes this trace as the following:

```
(1) R → I : Nr
(2) I → T : Nr⊕Nr'
(3) T → I : Nt, H(Nr'⊕Nt⊕K)
(4) I → R : Nr'⊕Nt, H(Nr'⊕Nt⊕K)
(5) R → I : H (H(Nr⊕Nr'⊕Nt⊕K)⊕K⊕Nr)
```

Several comments can be drawn from the trace:

- *Msg1:* The reader generates a nonce Nr and the intruder captures and stores the nonce in the course of the communication.
- *Msg2*: The intruder generates another nonce Nr' and sends Nr⊕Nr' to the tag.
- Msg3: The tag generates an instance of the nonce Nt and sends it with the hash function h(K⊕Nr⊕Nr'⊕Nt) to the intruder.
- Msg4: The intruder returns the received function to the reader with Nr'⊕Nt.
- Msg5: The reader sends the message h(h(K⊕Nr⊕Nr'⊕Nt)⊕K⊕Nr)to the tag. This message does not depend on the discovered attack (Impersonation of tag).

The attack on RFID tag authentication is realised in Msg4. We will describe the principle of this attack in the section of discussion.

```
% Open-Source Fixedpoint Model-Checker
   version 2012c
INPUT
   Lak.AnB

SUMMARY
  ATTACK_FOUND
GOAL:  weak_auth
DETAILS
BACKEND  OFMC
STATISTICS
  TIME 2184 ms
  parseTime 0 ms
  visitedNodes: 9 nodes
  depth: 2 plies

ATTACK TRACE
(x501,1) -> i: NR(1)
i -> (x502,1): NR(1) XOR x2624
(x502,1) -> i: NT(2),hash(sk(x502,x501)
   XOR NR(1) XOR x2624 XOR NT(2))
i -> (x501,1): x2624 XOR
   NT(2),hash(sk(x502,x501) XOR NR(1)
   XOR x2624 XOR NT(2))

(x501,1) -> i: hash(hash(sk(x502,x501)
   XOR NR(1) XOR x2624 XOR NT(2)) XOR
   sk(x502,x501) XOR NR(1))
% Reached State:
request(x501,x502,pRTNThashXorXorskTRNRN
   T,x2624 XOR NT(2),hash(sk(x502,x501)
   X
OR NR(1) XOR x2624 XOR NT(2)),1)
% state_rR(x501,2,sk(x502,x501),
   hash,x502,NR(1),hash(sk(x502,x501)
   XOR NR(1) XOR
 x2624 XOR NT(2)),x2624 XOR NT(2),x2624
   XOR NT(2),hash(sk(x502,x501) XOR
   NR(1) X
OR x2624 XOR
   NT(2)),hash(hash(sk(x502,x501) XOR
   NR(1) XOR x2624 XOR NT(2)) XOR x
2624 XOR NT(2)),1)
% state_rT(x502,1,hash,
   sk(x502,x501),x501,NR(1) XOR
   x2624,NT(2),NT(2),hash(sk(x5
02,x501) XOR NR(1) XOR x2624 XOR
   NT(2)),1)
%
   witness(x502,x501,pRTNThashXorXorskTR
   NRNT,NT(2),hash(sk(x502,x501) XOR
   NR(1) X
OR x2624 XOR NT(2)))
```

(a) OFMC Result

```
%% Constraint Logic-based ATtack SEarcher
(CL-ATSE) Version 2.5-18 (2012-septembre-
26).
   ---------------
     AtSe Summary
   ---------------
     Protocol file:
     Attack found : YES
     Analysed   : 6 states
     Reachable  : 3 states
     Translation: 0.00 seconds
     Computation: 0.00 seconds
   ---------------------------------------
                  Attack Description
(the list of protocol steps followed by
cl-atse)
   ---------------------------------------
   Short attack description :
   -------------------------
    Kind of attack:  Authentication on
(r,t,aut_tag,xor(X4064,n1(Nr),n5(Nt)))
       UnivQ. Vars:   false
      Substitution:  [
Nt(2)=xor(X4064,n1(Nr),n5(Nt))  Nr(5)=X4064
]
     Compact trace:   (r,3) (t,4) (r,3)
   Detailed attack description :
   ----------------------------
    i -> (r,3):  start
    (r,3) -> i:  n1(Nr)
                 &
Witness(r,t,aut_reader,n1(Nr));
                 & Built from step_0
    i -> (t,4):  X4064
    (t,4) -> i:
n5(Nt).{xor(X4064,k,n5(Nt))}_h
                 & Secret(k,(),set_68);
Witness(t,r,aut_tag,n5(Nt));
                 & Built from step_2
    i -> (r,3):
xor(X4064,n1(Nr),n5(Nt)).{xor(X4064,k,n5(N
t))}_h
    (r,3) -> i:
{xor(k,n1(Nr),{xor(X4064,k,n5(Nt))}_h)}_h
                 & Secret(k,(),set_60);
                 &
Request(r,t,aut_tag,xor(X4064,n1(Nr),n5(Nt
)));
                 & Built from step_1
   %% Job terminated successfully.
```

(b) CL-Atse Result

Fig. 1. Traces attacks on the LAK protocol

4.2 CL-Atse Result

Fig 1. (b) and Fig 2. shows the trace of attack on LAK protocol with the CL-Aste tool. We Summarizes this trace as the following:

(1) I → R : start
(2) R → T : Nr

(3) I → T : Nr‘
(4) T → I : Nt, H(Nr‘⊕Nt⊕K)
(5) I → R : Nr‘⊕Nr⊕Nt, H(Nr‘⊕Nt⊕K)
(6) R → I : H (H(Nr‘⊕Nt⊕K)⊕K⊕Nr)

The principle of the detected attack on LAK protocol by OFMC and CL-Atse is the same. The only difference is that intruder in CL-Atse generates a Nr’ nonce and sends it to the tag, but the intruder in OFMC generates the same nonce Nr and computes the xoring of Nr‘ with Nr (e.g. Nr⊕Nr‘), and sends the result to tag.

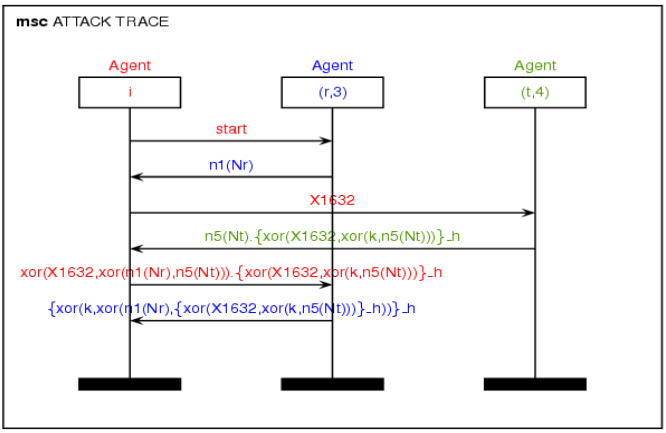

Fig. 2. Message Sequence Chart of ARA (Cl-Atse)

5 Discussion

In this section, we analyze the results of RFID authentication protocols and we quote the implementation and the countermeasure of ARA attacks.

Our results are based on the automatic verification of the authentication properties of each RFID authentication protocol. Concerning the message of tag authentication Auth_tag, the difference between these protocols is the type of one-way function (hash function and CRC) and the secret data which shared between the tag and the reader (server).

For tag impersonation of the studies protocols, an intruder can store all the messages exchanged in a protocol run. To tag impersonate, the adversary could replay f(α⊕nr⊕nt) if he ensures that f(α⊕nr⊕nt) = f(α⊕nr’⊕nt’). The activate intruder can generate a new none and make an algebraic calculate of the type xor operation between numbers. Then, to satisfy this condition the intruder sets nt’ to nr⊕nr’⊕nt. Here is the detail:

f(α⊕nr⊕nt) =? f(α ⊕nr’⊕nt’)

f(α⊕nr⊕nt) =? f(α ⊕nr’⊕nr⊕nr’⊕nt) ➔ replace nt’

$f(\alpha \oplus nr \oplus nt) = ?\ f(\alpha \oplus nr' \oplus \underline{nr' \oplus nr \oplus nt})$ ➔ commutativity

$f(\alpha \oplus nr \oplus nt) = ?\ f(\alpha \oplus 0 \underline{\oplus nr \oplus nt})$ ➔ nilpotence

$f(\alpha \oplus nr \oplus nt) = f(\alpha \oplus nr \oplus nt)$ ➔ neutral element

All the studied protocols cannot resist RFID tag authentication attack, and therefore an intruder can impersonate the tag. This type of attack is based on algebraic properties of algebraic operators (or, and, xor). The paper [12] aims to identify the algebraic problems which enable many attacks on RFID protocols. Toward this goal, three emerging types of attacks on RFID protocols, concerning authentication, untraceability, and secrecy are discussed. The common theme in these attacks is the fact that the algebraic properties of operators (e.g. *xor* operator) employed by the protocols are abused. The methods used to find algebraic replay attacks are sufficiently straight-forward. The algebraic replay attacks in RFID authentication protocols are described in some works such as [13, 14, 15, 16, 17, 18].

The relay attack system can use two transponders in order to relay the information that a reader and a token exchange during a cryptographic challenge response protocol. A proxy-token device is placed near the real reader and a proxy-reader device is placed near the real token, possibly unknown to its holder. Information can therefore be forwarded over a great distance if a suitable communication medium is chosen between the proxy-token and proxy-reader. As a result, the reader will report that it has verified the presence of a remote token and provide access to the attacker [19]. Practically, the ARA system is based on relay attack system. The difference between this system and relay attack system is: this system supports Dolev-Yao attack model (see section 2). Therefore, the proxy system can generate a random number and compute xor operation between numbers. The process of attack system for LAK protocol as following (see figure 3):

1. Legitimate reader generates a nonce Nr and sends it to the proxy-token.
2. Proxy-token receives it and blocks it; the proxy-token generates a nonce Nr' and forwards this nonce to the proxy-reader through the fast communication channels.
3. Proxy-reader fakes the real reader, and sends Nr' to the legitimate tag.
4. Legitimate tag computes a new nonce Nt and computes hash function H(Nr'⊕Nt⊕K) and transmits it to the proxy-reader.
5. Proxy-reader receives it and calculates the new nt' = nr⊕nr'⊕nt and forwards this message and hash function received to the proxy-token through the fast communication channel.
6. Proxy-token forwards nt' and H(Nr'⊕Nt⊕K) to the real reader.

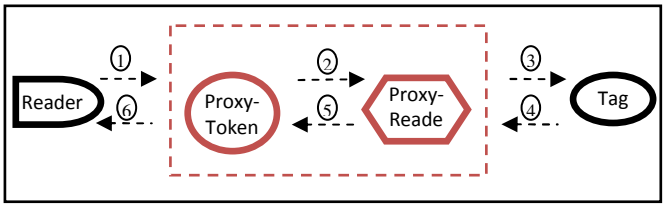

Fig. 3. Attack System

The principal vulnerability in studies protocols in use of xor operator in one-way function. Consequently, the solution is to change the primitive XOR (⊕) between the values of one-way function (α, Nr,Nt) by the concatenation (‖). Therefore, the new one-way function is: $f((\alpha \oplus Nr) \| Nt)$ or $f(\alpha \| (Nr \oplus Nt))$.

6 Conclusion

We have presented in this paper different protocols using xor-operator and one-way functions. The one-way functions in studying protocols are: hash function and CRC function. Our security analysis of these RFID authentication protocols by automatic formal tools. We showed that the verified protocols cannot resist RFID tag authentication attack therefore; an intruder can impersonate the tag.

The detected attack is the type of algebraic replay attacks (ARA) on tag authentication. The principal cause of the described attacks in our work is the abuse of the proprieties of xor-operator in the transmitted messages. The proposed solution for this attack is correcting the use of xor-operator and replacing it by concatenation operator.

References

1. Basin, D., Mödersheim, S., Viganò, L.: OFMC: A symbolic model checker for security protocols. International Journal of Information Security 4(3), 181–208 (2005)
2. Modersheim, S., Vigano, L.: The open-source Fixed-point model checker for symbolic analysis of security protocols. In: Aldini, A., Barthe, G., Gorrieri, R. (eds.) FOSAD 2007/2008/2009. LNCS, vol. 5705, pp. 166–194. Springer, Heidelberg (2009)
3. Turuani, M.: The CL-Atse Protocol Analyser. In: Pfenning, F. (ed.) RTA 2006. LNCS, vol. 4098, pp. 277–286. Springer, Heidelberg (2006)
4. HLPSL Tutorial: A Beginner's Guide to Modelling and Analysing Internet Security Protocols (2005), http://www.avispa-project.org/
5. Mödersheim, S.: Algebraic Properties in Alice and Bob Notation. In: Proceedings of Ares 2009, pp. 433–440. IEEE Xplore (2009); Extended version: Technical Report RZ3709, IBM Zurich Research Lab (2008)
6. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on Information Theory IT-29(2), 198–208 (1983)
7. Lee, S., Asano, T., Kim, K.: RFID mutual authentication scheme based on synchronized secret information. In: Symposium on Cryptography and Information Security (2006)
8. Chien, H.-Y., Huang, C.-W.: A lightweight RFID Protocol Using Substring. In: Embedded Ubiquitous Computing (EUC), pp. 422–431 (2007)
9. Liu, Y.: An Efficient RFID Authentication Protocol for Low-Cost Tags. In: Proceedings of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Shanghai, China, pp. 706–711 (2008)
10. Qingling, C., Yiju, Z., Yonghua, W.: A minimalist mutual authentication protocol for RFID system & BAN logic analysis. In: Proc. of CCCM 2008, pp. 449–453. IEEE Computer Society, Los Alamitos (2008)
11. Wei, C.-H., Hwang, M.-S., Chin, A.-Y.: A Mutual Authentication Protocol for RFID. IT Professional 3, 20–24 (2011)

12. van Deursen, T., Radomirović, S.: Algebraic Attacks on RFID Protocols. In: Markowitch, O., Bilas, A., Hoepman, J.-H., Mitchell, C.J., Quisquater, J.-J. (eds.) WISTP 2009. LNCS, vol. 5746, pp. 38–51. Springer, Heidelberg (2009)
13. Cao, T., Shen, P.: Cryptanalysis of Two RFID Authentication Protocols. International Journal of Network Security 9(1), 95–100 (2009)
14. Jannati, H., Falahati, A.: Cryptanalysis and Enhanced of Two Low Cost RFID Authentication protocols. International Journal of UbiComp 3(1), 1–9 (2012)
15. van Deursen, T., Radomirovic, S.: Attacks on RFID Protocols. Report 2008/310, Cryptology ePrint Archive (2008)
16. Chen, X., van Deursen, T., Pang, J.: Improving Automatic Verification of Security Protocols with XOR. In: Breitman, K., Cavalcanti, A. (eds.) ICFEM 2009. LNCS, vol. 5885, pp. 107–126. Springer, Heidelberg (2009)
17. Mihailescu, M.I.: Resreach on Solutions for Preventing Algebraic Attacks Against Biometric and RFID Protocols. ACTA Universitatis Apulensis (Special Issue), 371–386 (2011)
18. Chikouche, N., Cherif, F., Benmohammed, M.: Vulnerabilities of two Recently RFID Authentication Protocols. In: International Conference on Complex Systems, Agadir, Morocco (2012)
19. Hancke, G.P.: Practical Attacks on Proximity Identification Systems. In: Proceedings of IEEE Symposium on Security and Privacy, pp. 328–333 (May 2006) (short paper)