A Secure Lightweight Mutually Authenticated Radio Frequency Identification (RFID) Protocol

Varinder Kumar[1] and Sujata Mohanty[2]

Abstract— **Radio Frequency Identification (RFID) devices are made of small electronic chips and antenna. The proposed scheme designed for lightweight authentication protocol using simple HMAC or SHA-1 hash operation. The scheme is mutually authenticated and it also achieves security against possible known attacks. The proposed RFID scheme is formally and informally analyzed using Proverif and AVISPA tool. The security analysis shows that the scheme achieves efficiency and is applicable for lightweight devices.**

Keywords: RFID, Replay Attack, Mutual Authentication

I. INTRODUCTION

Nowadays the usage of RFID device is becoming very popular. RFID technology is now used in many fields such as in healthcare, smart cards, inventory control systems etc. This technology uses the radio frequencies for communication, which has three main components namely, tag, reader and a back-end server[1].

Tags are the devices which hold the identification information, basically of two type: passive and active tags. The active tags have their own power source, such as a battery. The passive tags do not have their own power of source and they rely upon the power they get, in the form of radio signals from the reader as a request.

Readers are devices which are used for scanning the tags and receiving identification information from the tags and transmit the information to the back-end server. It acts as a mediator between the back-end server and the tag.

The back-end server contains a database in which it stores the data related to each tag and this data is later used for the identification of the tag by the server. Whenever the reader sends the request for identification of any tag, the server matches the received information with each tag information stored in the database until any match is found. When any match is found, it is considered that the tag is authenticated by the server otherwise the authentication process is considered to be failed.

RFID technology is a promising replacement for some of the popular applications, such as barcode and magnetic strip system. In the barcode systems, each device needs to be scanned by the reader device one by one, whereas, in RFID system multiple devices can be scanned at a time. The magnetic strips are found on the back of the credit and debit cards need to be swiped for transfer of identification information, whereas, the RFID is a contactless technology and there is no need to swipe. The swiping mechanism does not work if any scratch is found on the magnetic strip. But RFID does not contain such type of issues.

As the usage of RFID devices has increased, several weakness and vulnerabilities in RFID devices are being found these days. There are many security challenges such as tag impersonation attack, reader impersonation attack, replay attack, desynchronization attack and forgery attacks are found in RFID technology.To solve these types of security challenges, several security protocols are developed for RFID devices which can provide security in the RFID devices [1]. Till date, not a single RFID protocol is full proof as it lacks one or more essential security features.

In this paper, we proposed a mutually authenticated RFID protocol, which is devoid of computationally intensive operations. Also, simple hash operations, such as HMAC or SHA1 are used in the proposed scheme, instead of using complex hash chains. The formal and informal security analysis of the proposed scheme is resistant to desynchronization attack, impersonation attack, and replay attack. Moreover, it achieves mutual authentication with lightweight operations. We validated the proposed scheme with widely accepted AVISPA tool, also verified using ProVerif tool.

The remaining part of the paper is designed as follows. Section II gives a formal overview of related work done in this area. In section III, we demonstrate our protocol. Section IV presents security analysis of the proposed scheme. In section V, we validate the scheme using AVISPA and ProVerif tool. Finally, we conclude in section VI.

II. RELATED WORK

As the usage of this technology increased, RFID scheme needs to be secure against several types of security attacks. Many authors have proposed protocols, that can provide security to this scheme. But it is found that these security protocols also suffer from some of the security challenges.

Cho et al.[1] proposed a hash-based scheme that provided mutual authentication, forward security, and confidentiality but due to the generation of RID, which is a group ID, it is susceptible to several attacks, such as, tag impersonation attack, reader impersonation attack, and desynchronization attack. Srivastava et al.[2] provided a hash-based protocol which provided mutual authentication, resistance against eavesdropping and tracing attack but it also suffered from forgery attack.

Tian et al.[3] proposed a security protocol which was based on just three operations, bitwise XOR, left rotation

[1] Varinder Kumar, Computer Science and Engineering, National Institute of Technology, Rourkela, India varinderkumarldc@gmail.com

[2] Sujata Mohanty, Computer Science and Engineering, National Institute of Technology, Rourkela, India sujatam@nitrkl.ac.in

and permutation operation but it is found that this protocol also suffers from security issues of desynchronization attack.

Zhang et al.[4] proposed a security protocol based on ECC, which can satisfy all security issues, but later, Tu et al.[11] proved that this protocol was weak against the impersonation attack. Zhao et al.[6] had also proposed an ECC based protocol for RFID, but it also suffered from the issue of forwarding secrecy. Yoon et al.[7] have also proposed an ECC based protocol, later Xie[8] proved this protocol suffered from stolen verifier and off-line password guessing attack. He proposed an improvement scheme over Yoon's scheme.Farash et al.[9] found that the protocol proposed by Xie[8] was insecure.

III. Proposed Scheme

In this section, we propose a security protocol for RFID system. In this protocol, it is assumed that the communication channel between the server and the reader is secure and the communication channel between reader and tag is considered to be insecure. The proposed scheme consists of three participants, namely, tag, reader, and a back-end server. The notations for proposed protocol given in Table I and the protocol is demonstrated in Figure 1 as follows.

TABLE I: Notations

Notation	Description
ID	Unique identification of tag
$DATA$	Data related to tag
R_r	Random number generated by reader
N_1	Random number generated by tag
q	Random number generated by tag
N_2	Random number generated by Server
$H()$	One way hash function
$S^t{}_{i-1}, S^r{}_{i-1}$	Secret value of previous round
$S^t{}_i, S^r{}_i$	Secret value of shared between tag and server

The operations of the proposed scheme are as per the following steps.

Step 1: The secret values shared between tags and server database.

Step 2: The reader generates a random number R_r and sends this to the tag as a request.

Step 3: The tag generates two random values N1 and q. Then computes the values of following parameters.

$Q = H(s^t{}_i \oplus s^r{}_i) \oplus q$

$L_1 = H((R_r \oplus s^t{}_i)||(s^t{}_i \oplus N_1))$

$M_1 = H(ID||L_1||q)$

$M_2 = H(M_1||R_r||(N_1 \oplus q))$

The tag sends the values of (Q, M_2, N_1) as a response to the reader.

Step 4: After receiving a response from the tag, the reader adds its random number R_r into the response and sends to the server.

Step 5: The server recieves the response from reader and search in the database for the perticular tag based on the received information.

Step 6: First the sever extracts the ID, s^t and s^r from database.

Step 7: Then the server computes following parameters.

$q' = Q \oplus H(s^t{}_i \oplus s^r{}_i)$

$L'_1 = H((R_r \oplus s^t{}_i)||(s^t{}_i \oplus N_1))$

$M'_1 = H(ID||L'_1||q')$

$M'_2 = H(M'_1||R_r||(N_1 \oplus q'))$

Step 8: If for a tag $M'_2 = M_2$, then the server authenticate the tag, else the steps 6-7 are repeated. Then the server updates the secret values by performing following set of operations.

a) Server generates a random number N_2

b) Server computes:

$L_2 = H(ID||(M'_2 \oplus N_2)||R_r)$

$M_3 = H(L_2||N_2||s^r{}_i)$

c) Message generated as

$DATA||M3||N2$

d) Server update the secret values as

$S^t{}_{i-1} = s^t{}_i$, $S^r{}_{i-1} = s^r{}_i$

$S^t{}_i = H(ID||M'_2||R_r||s^r{}_i)$

$S^t{}_i = H(M'_2||q||s^r{}_i)$

If tag is not found, the server repeats the steps with previous session values $S^r{}_{i-1}$ and $S^t{}_{i-1}$

Step 9: When the reader gets the response from server, it extracts the DATA from the message and forwards the message to the tag. Then the tag extracts M_2 and N_2 from message and computes,

$L'_2 = H(ID||(M_2 \oplus N_2)||R_r)$

$M'_3 = H(L'_2||N_2||s^r{}_i)$

The tag will check whether $M'_3 = M_3$. If it matched then the server is authenticated and it update its secret values as follows.

$s^t{}_i = H(ID||M_2||R_r||s^r{}_i)$

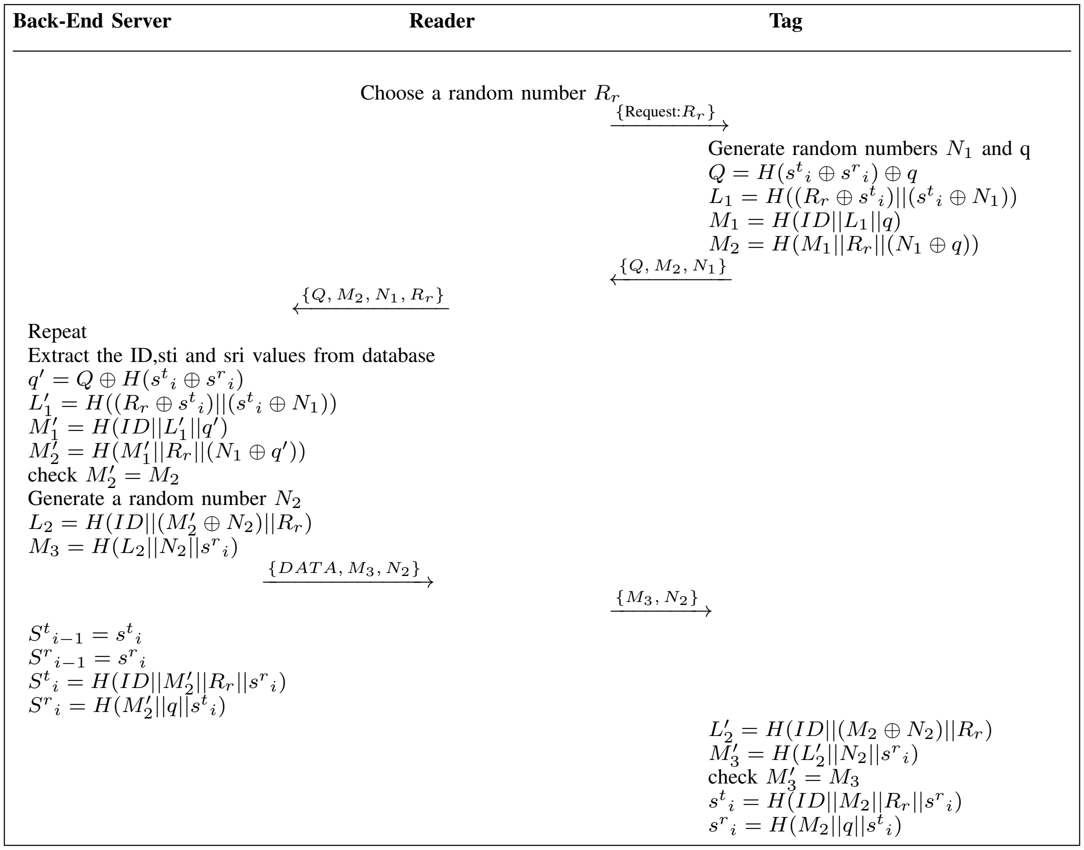
$s^t{}_i = H(M_2||q||s^r{}_i)$

If match not found then the authentication is considered to be failed and no updation will occur.

IV. Security Analysis

A. Desynchronization Attack

The Desynchronization attack related to RFID in which the secret values stored in back-end server database and tags memory become different. To perform this attack the attacker blocks the communication between the parties. The proposed protocol is secure against the desynchronization attack. In this protocol, the adversary can block the last message $\{M_3, N_2\}$ sent by the reader to the tag for updating the values to perform desynchronization between the tag and the server. The server stores both new as well as previous session's secret values $\{S^t{}_{i-1}, S^r{}_{i-1}, S^t i, S^r i\}$ and when a new request comes for authentication of tag, the server first compares request using current secret values $\{S^t{}_i, S^r{}_i\}$. If it does not match, then the server compares the request



Back-End Server **Reader** **Tag**

Choose a random number R_r

$\xrightarrow{\{\text{Request}:R_r\}}$

Generate random numbers N_1 and q
$Q = H(s^t{}_i \oplus s^r{}_i) \oplus q$
$L_1 = H((R_r \oplus s^t{}_i)||(s^t{}_i \oplus N_1))$
$M_1 = H(ID||L_1||q)$
$M_2 = H(M_1||R_r||(N_1 \oplus q))$

$\xleftarrow{\{Q, M_2, N_1\}}$

$\xleftarrow{\{Q, M_2, N_1, R_r\}}$

Repeat
Extract the ID,sti and sri values from database
$q' = Q \oplus H(s^t{}_i \oplus s^r{}_i)$
$L_1' = H((R_r \oplus s^t{}_i)||(s^t{}_i \oplus N_1))$
$M_1' = H(ID||L_1'||q')$
$M_2' = H(M_1'||R_r||(N_1 \oplus q'))$
check $M_2' = M_2$
Generate a random number N_2
$L_2 = H(ID||(M_2' \oplus N_2)||R_r)$
$M_3 = H(L_2||N_2||s^r{}_i)$

$\xrightarrow{\{DATA, M_3, N_2\}}$

$\xrightarrow{\{M_3, N_2\}}$

$S^t{}_{i-1} = s^t{}_i$
$S^r{}_{i-1} = s^r{}_i$
$S^t{}_i = H(ID||M_2'||R_r||s^r{}_i)$
$S^r{}_i = H(M_2'||q||s^t{}_i)$

$L_2' = H(ID||(M_2 \oplus N_2)||R_r)$
$M_3' = H(L_2'||N_2||s^r{}_i)$
check $M_3' = M_3$
$s^t{}_i = H(ID||M_2||R_r||s^r{}_i)$
$s^r{}_i = H(M_2||q||s^t{}_i)$

Fig. 1: RFID protocol based on Nonce

using the previous sessions secret values $\{S^t{}_{i-1}, S^r{}_{i-1}\}$. Hence the desynchronization problem can never occur in the proposed scheme.

B. Mutual Authentication

In this protocol, only the tag and the back-end server has the knowledge of the secret $\{S^t{}_{i-1}, S^r{}_{i-1}, S^t i, S^r i\}$. The nonce $\{N_1, N_2\}$ are also transferred secretly in the open channel which cannot be revealed without knowing the secret values that are known only to the tag and server. Thus only tag and back-end server can authenticate each other. Thereby it provides the mutual authentication between tag and server.

C. Impersonation attack

This protocol is also secure against the impersonation attack. As the nonce values $\{N_1, N_2\}$ are sent secretly using the secret values $\{S^t{}_i, S^r{}_i\}$ in the communication channel and at the other end the secret values are used to check the authentication of sending party. Hence to perform impersonation attack, the adversary needs to know the secret values in advance, which is not possible because the secret values are updated after each session.

D. Replay Attack

The replay attack is the attack in which the adversary receives the messages sent by one party to the other party using the insecure channel and later the adversary sends the received message to the other party impersonating as a legitimate person. In the proposed scheme, the nonce $\{N_1, N_2\}$ and the secret values $\{S^t{}_i, S^r{}_i\}$ change after each session, hence the replay attack is not possible.

V. Validation of the proposed scheme

We first show that the proposed scheme is tested in the widely accepted AVISPA tool. Then we demonstrate verification using ProVerif tool

A. Validation using AVISPA

AVISPA(Automated Validation of Internet Security Protocol) tool is used for the verification of large-scale internet protocol. This tool uses an HLPSL language to specify the cryptographic protocols for the AVISPA tool. The AVISPA tool uses the four backends for validation of the protocols namely: OFMC, CL-ATSE, SATMC, TA4SP. The result of the AVISPA tool is given Figure 2.

B. Verification using ProVerif

In the proposed protocol the adversary can get some information using eavesdropping on the channel. The adversary gets the values of Q,M_2,N_1 but he cannot get the ID and the secret values from the received value.In the last phase, the adversary get the information of M_3 and N_2 but these values are not enough to get the secret values.

The public channel is used between the tag and the reader and the secure private channel is used between reader and

```
% OFMC % Version of 2006/02/13 SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/span/span/testsuite/results/finalize.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.04s
visitedNodes: 24 nodes
depth: 5 plies
```

(a) Verification using OFMC

```
SUMMARY
SAFE
DETAILS BOUNDED_NUMBER_OF_SESSIONS
TYPED_MODEL
PROTOCOL
/home/span/span/testsuite/results/finalize.if
GOAL
As Specified
BACKEND
CL-AtSe

STATISTICS
Analysed : 1 states
Reachable : 1 states
Translation: 0.01 seconds
Computation: 0.00 seconds
```

(b) Verification using CL-ATSE

Fig. 2: Verification of Protocol using SPAN AVISPA

the back-end server.The result of the ProVerif verification tool is shown in Figure 3. The code of the ProVerif tool is given in appendices section.

```
Starting query inj-event(ServerAuthed(id)) ==>inj-event(ServerStarted(id))
RESULT inj-event(ServerAuthed(id)) ==>inj-event(ServerStarted(id)) is true.

Starting query inj-event(TagAuthed(id_4424)) ==>inj-event(TagStarted(id_4424))
RESULT inj-event(TagAuthed(id_4424)) ==>inj-event(TagStarted(id_4424)) is true.

Starting query not attacker(q[])
RESULT not attacker(q[]) is true.

Starting query not attacker(Sr[])
RESULT not attacker(Sr[]) is true.

Starting query not attacker(St[])
RESULT not attacker(St[]) is true.
```

Fig. 3: Verification result using ProVerif

Hence it is proved that the proposed satisfy all the queries and the secret values are secured and the adversary can never get the secret values.

VI. Conclusion

This paper presents a mutually authenticated protocol for RFID. This scheme uses the simple HMAC or SHA-1 hash operation. The analysis of the protocol shows that the protocol is suitable for lightweight applications. The security analysis of the proposed scheme done in AVISPA and ProVerif tool proved that the scheme is secure against possible known attacks.

References

[1] Cho, Jung-Sik, Young-Sik Jeong, and Sang Oh Park. ”Consideration on the brute-force attack cost and retrieval cost: A hash-based radio-frequency identification (RFID) tag mutual authentication protocol.” Computers & Mathematics with Applications 69.1 (2015): 58-65.

[2] Srivastava, Keerti, et al. ”A hash based mutual RFID tag authentication protocol in telecare medicine information system.” Journal of medical systems 39.1 (2015): 153.

[3] Tian, Yun, Gongliang Chen, and Jianhua Li. ”A new ultralightweight RFID authentication protocol with permutation.” IEEE Communications Letters 16.5 (2012): 702-705.

[4] Zheng, Lijuan, et al. ”Mutual Authentication Protocol for RFID Based on ECC.” Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), 2017 IEEE International Conference on. Vol. 2. IEEE, 2017.

[5] Zhang, Zezhong, and Qingqing Qi. ”An efficient RFID authentication protocol to enhance patient medication safety using elliptic curve cryptography.” Journal of medical systems 38.5 (2014): 47.

[6] Zhao, Zhenguo. ”A secure RFID authentication protocol for healthcare environments using elliptic curve cryptosystem.” Journal of medical systems 38.5 (2014): 46.

[7] Yoon, Eun-Jun, et al. ”Robust mutual authentication with a key agreement scheme for the session initiation protocol.” IETE Technical Review 27.3 (2010): 203-213.

[8] Xie, Qi. ”A new authenticated key agreement for session initiation protocol.” International Journal of Communication Systems 25.1 (2012): 47-54.

[9] Farash, Mohammad Sabzinejad, and Mahmoud Ahmadian Attari. ”An enhanced authenticated key agreement for session initiation protocol.” Information Technology and Control 42.4 (2013): 333-342.

[10] Zhang, Liping, Shanyu Tang, and Zhihua Cai. ”Efficient and flexible password authenticated key agreement for voice over internet protocol session initiation protocol using smart card.” International Journal of communication systems 27.11 (2014): 2691-2702..

[11] Tu, Hang, et al. ”An improved authentication protocol for session initiation protocol using smart card.” Peer-to-Peer Networking and Applications 8.5 (2015): 903-910.

[12] Shi, Zhicai, et al. ”A Lightweight RFID Authentication Protocol with Forward Security and Randomized Identifier.” Security, Privacy, and Anonymity in Computation, Communication, and Storage: 9th International Conference, SpaCCS 2016, Zhangjiajie, China, November 16-18, 2016, Proceedings 9. Springer International Publishing, 2016.

[13] Benssalah, Mustapha, Mustapha Djeddou, and Karim Drouiche. ”Design and implementation of a new active RFID authentication protocol based on elliptic curve encryption.” SAI Computing Conference (SAI), 2016. IEEE, 2016.

[14] Mujahid, Umar, Muhammad Najam-ul-Islam, and Shahzad Sarwar. ”A new ultralightweight RFID authentication protocol for passive low cost tags: KMAP.” Wireless Personal Communications 94.3 (2017): 725-744.

[15] Priyanka, D. Daya, et al. ”A survey on applications of RFID Technology.” Indian journal of Science and Technology 9.2 (2016).

[16] Ahmadian, Zahra, Mahmoud Salmasizadeh, and Mohammad Reza Aref. ”Desynchronization attack on RAPP ultralightweight authentication protocol.” Information processing letters 113.7 (2013): 205-209.

APPENDICES

```
(*–channels–*)
free Ch_Pub:channel.
free Ch_Sec:channel[private].
(*–Constants & Variables–*)
free ID, St, Sr, St_1, Sr_1 :bitstring.[private]
free Data:bitstring.
free q:bitstring[private].
```

In this step the events are declared for the protocol.

```
(*–events–*)
event TagAuthed(bitstring).
event TagStarted(bitstring).
event ServerAuthed(bitstring).
event ServerStarted(bitstring).
(*–Queries–*)
query attacker(St).
query attacker(Sr).
query attacker(q).
query id : bitstring; inj − event(TagAuthed(id)) == >inj − event(TagStarted(id)).

id : bitstring; inj − event(ServerAuthed(id)) == >inj − event(ServerStarted(id)).
(*–Constructors–*)
fun H(bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring.
fun add(bitstring,bitstring):bitstring.
fun sub(bitstring,bitstring):bitstring.
fun mult(bitstring,bitstring):bitstring.
fun mod(bitstring,bitstring):bitstring.
fun concat(bitstring,bitstring):bitstring.
(*Destructors and equations*)
equation forall a:bitstring,b:bitstring; xor(xor(a,b),b)=a.
(*–Tag–*)
let Tag=
event TagStarted(ID);
in(Ch_Pub, (xRr : bitstring));
new N1:bitstring;
let Q=xor(H(xor(St,Sr)),q) in
let L1=H(concat(xor(xRr,St),xor(St,N1))) in
let M1=H(concat(ID,concat(L1,q))) in
let M2=H(concat(M1,concat(xRr,xor(N1,q)))) in
out(Ch_Pub,(Q,M2,N1));
in(Ch_Pub,(xxM3:bitstring,xxN2:bitstring));
let L2'=H(concat(ID,concat(xor(M2,xxN2),xRr))) in
let M3'=H(concat(L2',concat(xxN2,Sr))) in
if(M3'=xxM3) then
let St=concat(ID,concat(M2,concat(xRr,St))) in
let Sr=concat(M2,concat(q,Sr)) in
event ServerAuthed(ID)
(*–Reader–*)
let Reader=
new Rr:bitstring;
out(Ch_Pub,(Rr));
in(Ch_Pub,(xQ:bitstring,xM2:bitstring,xN1:bitstring));
out(Ch_Sec,(xQ,xM2,xN1,Rr));
in(Ch_Sec,(xData:bitstring,xM3:bitstring,xN2:bitstring));
out(Ch_Pub,(xM3,xN2))
(*–Server–*)
let Server=
event ServerStarted(ID);
in(Ch_Sec,(xxQ:bitstring,xxM2:bitstring,xxN1:bitstring,xxRr:bitstring));
new N2:bitstring;
let q'=xor(xxQ,H(xor(St,Sr))) in
let L1'=H(concat(xor(xxRr,St),xor(St,xxN1))) in
let M1'=H(concat(ID,concat(L1',q'))) in
let M2'=H(concat(M1',concat(xxRr,xor(xxN1,q')))) in
if(M2'=xxM2) then
event TagAuthed(ID);
let L2=H(concat(ID,concat(xor(M2',N2),xxRr))) in
let M3=H(concat(L2,concat(N2,Sr))) in
let St_1=St in
let Sr_1=Sr in
let St=concat(ID,concat(M2',concat(xxRr,St_1))) in
let Sr=concat(M2',concat(q',Sr_1)) in
out(Ch_Sec,(Data,M3,N2))
```