2016 8th International Symposium on Telecommunications (IST'2016)

RFID Light Weight Server-Less Search Protocol Based on NLFSRs

Abolfazl Falahati and Hamid Azizi
Dept. of Electrical Engineering (DCCS Lab)
University of Science and Technology, Tehran
afalahati@iust.ac.ir

Robert. M. Edwards
Dept. Electronics Engineering (CMCR lab)
Loughborough University, UK
R.M.Edwards@lboro.ac.uk

***Abstract*— Spectacular expansion of RFID systems and applications by several industries such as business, electronics, health, marketing, transportation, etc. has made RFID ever more popular. Within this context, the industry search for lower tag weight and low cost system together with solemn security to satisfy the consumer. Tag search is among the most requested protocols in such systems. Being light weight is regarded as another important specification for the said protocols, so that they can be utilized in low expense tags. In this paper, a server-less search protocol is proposed in low expense tags with the aid of NLFSRs and simple logical operations to conflict with security attacks. This protocol requires only a two steps information exchanges between tag and reader, and although it has simple structure, but is completely safe against most of the known security attacks. In comparison with other protocols, this protocol also requires a small number of gates for implementations. There are no need to exclude the collisions in tags as its other privileges.**

Keywords— RFID System and Security; Light-Weight Search Protocol; Non-Linear Feedback Shift Registers

I. INTRODUCTION

Today, it is a great challenge to employ RFID systems for their spectacular expansions in system tags infrastructure and security risks they may go under. A common RFID system embraces tags, readers and backend servers. It is ideal to dismiss and remove servers as it is recently become popular, making system less vulnerable to reader to servers attacks. Servers just play the preliminary information exchanging role to tags and readers. So dismissing the server will force the reader to acquire greater saving and calculations capabilities with high technology.

In lowest possible instant and security risks, a search protocol aims to find a tag identification in confidence among different tags. The most important security principles in RFID search protocols is the mutual authentication between tag and reader but the attacker should not be aware of the searched tag identity. There exist albeit some other different attacks such as relay and de-synchronization security attacks which are investigated later. In other words, all of these security solutions should be delivered in such a way that not only eliminate the use of power of tags and their establishment cost but also enhance search operation speed as much as possible [1].

Before [2] publication, all RFID protocols were server based and the tag identification authentication being performed through connecting server to reader. In this model, reader movement is restricted and they require a secure connection with server. The server-less protocols introduced in [2] transfer server duties inside the readers. This search protocol is based on a number of hash functions. Indeed paper [2] introduces three types of such search protocols. In the third version, as the most complete one, by the help of random response sent by the tags which have not been searched, the protocol is safe against tag tracking attacks. The weak point of this method is that, tags collision during implementations should be prevented and the reader should have a capability to receive a large amount of messages from tags in parallel. Besides this issue, in this protocol, there exists the possibility of the message reply attack. The two other deficiencies of this protocol, relates to using hash function which causes the increase of the required gates to implement the protocol (more than 8000 gate [14]) and the other one, is the possibility of the reader tracking because the reader's identity will be easily disclosed in the inquiry sent from reader.

Hash functions are the basis of the search protocols introduced in [6], [9] and [12]. Their popularity lies within its low cost tags. In [6], a proposed protocol based on a hash function and a simple random number generator, but by performing cryptanalysis attack to the random number generator and getting the incoming sequences from its output, it is possible to establish de-synchronization and tag tracking attacks on it, without hash function analysis. In [12], it is also needed to resolve tag collision in order to remove tracking. After receiving all responses, the reader should review them one by one so it can distinguish the existence or nonexistence of the desired tag. This process is time consuming to establish the protocol and finally [9] provides a different protocol, based on the hash function but again it is too time consuming.

The introduced and investigated protocols in [5] and [7] use symmetric encryption functions instead of hash functions. It should be noted that, even though, the symmetric encryption functions such as AES128, requires less number of gates for implementations than hash functions, but they requisite more clock generations [11], which results in the increase of the tag power and the decrease of the search protocol speed. Meanwhile, the amount of the requested gates for implementation on inactive tags are high. In [7], the attacker can dump a tag just with having its ID, without needing its secret key. This protocol also uses a time stamp which can

978-1-5090-3435-2/16/$31.00 ©2016 IEEE

prevent it from the reader inquiry reply. In [5], the search protocol too relies on the AES128 and similar to some of the previous mentioned methods, but enquires the tag collision exclusion.

II. System Model and Assumptions

The considered model includes tags and readers. In the produced protocol, instead of the conventional encryption functions such as hash functions and symmetric encryption, proposed model deploys a k bit NLFSRs in readers and tags. In [1], by the help of the NLFSRs, Misera et al. introduced a light weight authentication protocol. This idea was employed for the design of the search protocol and the other services helped to improve its security. Fig.1 illustrates one 32 bit NLFSR including OR and AND gates. It should be noted that, by changing the situation, type and number of gates, it is possible to produce a plenty of separated NLFSRs [1]. In this protocol, each tag needs to have an exclusive NLFSR. The NLFSR related to T_j illustrates as $NLFSR_j$. None of the two tags in RFID, should have similar NLFSR structure, because the NLFSRs are used to tag authentication. In order to have relation with the desired tags, every reader should have the related NLFSR of the tags. In fact, these NLFSRs hold a secret key role. In contrast to LFSRs, NLFSRs are more resistant against cryptanalysis attacks, but they lack a regular method to produce a NLFSR with the maximum period [10]. In [1] too, for the increase of the NLFSRs security, it uses one f function which encompasses several 8 bit S-boxes and k/2 bit permutation, in which k denotes the amount of bits for NLFSRs. In the proposed search protocol, for the security improvement, it is possible to use this function or any other similar function, but the used function has to be simple as much as possible.

Each reader with R_i symbol, includes one access list (L_i) which specifies the information for those tags which the reader can establish a connection (search for). This access list will be located inside the reader by the reliable backup server. The access list of each reader is then:

$$L_i = \begin{bmatrix} K_1 & ID_1 & NLFSR_1 \\ \dots & & \\ K_j & ID_j & NLFSR_j \\ \dots & & \\ K_n & ID_n & NLFSR_n \end{bmatrix} \quad (1)$$

In which ID_j and $NLFSR_j$ are identity and the NLFSR related in tag T_j respectively. The definition for the K_j is given as follows:- At first, the structure of $Z=NF_j(x,y)$ is shown in Fig. 2.

The f function structure is totally discretional, it is possible, for example, to use the utilized structure given in [1]. Fig. 3 illustrates this structure, in which in1and in2 are the inputs and m is the f output.

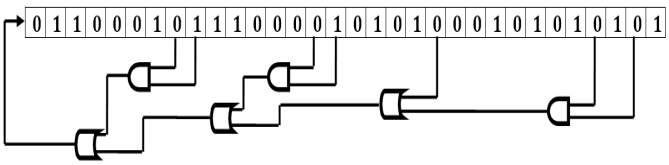

Fig1. Structure of a simple 32 bit NLFSR including 3 OR and 3 AND gates

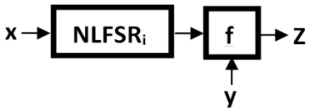

Fig2. Structure of NF_j function
Fig3. Structure of f function in [1]

1. x=in1+in2
2. if k−1-th bit of x ==0 or 0-th bit of x==0 then
3. L=Right-Half(x)
4. R=Left-Half(x)
5. Else
6. L=Left-Half(x)
7. R=Right-Half(x)
8. End if
9. L′=L is divided into blocks of 8-bit data units each and is sent to 8 bit
10. R′=R is passed through a k/2-bit permutation choice
11. L′ and R′ are concatenated to obtain m

Now definition for the K_j is $K_j=NF_j(SK_j,ID_j)$ in which SK_j is the secret key of tag T_j. The counting stamp is another factor which is used in this protocol. The counter stamp is defined as one k bit number with (called CTC_i) which locates inside the reader R_i and it will increase one unit for each second. It is functionally the same as time stamp but it is easier to implement. A k bit memory is located in each tag called Cj which saves the counter amount in the last search of tag. In the protocol structure, the variable RN^x is the x-th random number, M^x is the x-th output of NF_j function, $\oplus$ means XOR operation and || means concatenation.

III. The Proposed Search Protocol

The proposed RFID search protocol is completely shown in Fig. 4 in which all above mentioned operations are performed on k bit numbers.

Step 1: It is supposed that the reader intends to search tag T_j with ID_j. The reader by the help of its L_i, finds Kj and NLFS R_j. Then the reader produce a random number called Rn^1. Now the reader calculates $m^1= NF_j(RN^1,K_j\oplus CTC_i)$ through RN^1, K_j, CTC_i and NF_j.

Step 2: The reader sends the amount of the m^1, CTC_i and $RN^1\oplus K_j$ to all tags.

Reader R_i		Tag T_i
1. $m^1 = NF_j(RN^1, K_j \oplus CTC_i)$		
2. $RN^1 \oplus K_j \| CTC_i \| m^1$	send →	3. If $CTC_i > C$ Then
		4. $K' = NF(SK, ID)$
		5. $m^2 = NF(RN^1 \oplus K_j \oplus K', K' \oplus CTC_i)$
		6. If $m^1 = m^2$ Then
		7. $C_j = CTC_i$
		8. $m^3 = NF_j(RN^2, m^2)$
11. $m^4 = NF_j(RN^2 \oplus K_j' \oplus K_j, m^1)$	← send	9. $RN^2 \oplus K_j' \| m^3$
12. If $m^4 == m^3$ Then: The tag has been found		10. Else if : end
13. Else if: The tag hasn't been found		10. Else if : end

Fig. 4. The proposed RFID light-weight search protocol

Step 3: The amount of the CTC_i is compared with C in tags and if $CTC_i \leq C$, it means that the received inquiry is replied so the protocol goes to the step 10 and the tag doesn't reply to it. Otherwise it means to receive a new inquiry and the protocol goes to the step 4.

Step 4: In this step, by the help of the ID, SK and NLFSR, tags find $K' = NF(SK,ID)$

Step5: Then all tags by received CTC and $RN^1 \oplus Kj$ and the amount of k' which is calculated in the previous step, will calculate the amount of $m^2 = NF(RN^1 \oplus K_j \oplus K', K' \oplus CTC_i)$.

Step 6: About the searched tag(Tj) surely $K' = K_j$ and $M^1 = M^2$ and the protocol will go to the next step. But it is not true about other tags and here protocol goes to step 10 and there is not any reply from tags.

Step 7: Tag T_j will replace its current C_j with the received CTC_i.

Step 8: Then the tag T_j will produce the random number $RN^2$and it will calculate $m^3 = NF_j(RN^2, m^2)$

Step 9: Now the searched tag will send the calculated m^3 and $RN^2 \oplus K_j'$to the reader.

Step 10: This step means the disconnection between the tag and the reader.

Step11: The reader by the help of K_j, NF_j, $RN^2 \oplus K_j'$ and m^1 which was previously calculated, calculates the amount of $m^4 = NF_j(RN^2 \oplus K_j' \oplus K_j , m^1)$.

Step 12: In case of $m^3 = m^4$, the searching tag is correctly disclosed and it is located in reader location. Otherwise the message is iterated or faced with change so the reader will go to the step 13 and it recognizes that the tag has not been found.

IV. Security Analysis of the Proposed Search Protocol

Mutual Authentication: The inquiry of the reader will be confirmed by the tag when $CTC_i > C$ (tag does not respond to the replied inquiries) and also when $m^1 = m^2$. Just a valid reader holding L_i and having access to the K_j and $NLFSR_j$ function can calculate m^1 correctly and send it to tags. Just one valid tag having correct C_j, $NLFSR_j$, SK_j and ID_j can authenticate the reader and calculate m^3 correctly and send it to reader and finally the same valid reader can calculate m^4 correctly and authenticate the tag, therefore the authentication in the proposed protocol is completely mutual.

Reader Tracking: In spite of the said protocol in [2], in the proposed protocol there is not any information disclosure about reader identity and there does not exist any debate about reader tracking.

Desired Tag Tracking: Based on the protocol type, it is possible to track a desired tag just when it is also possible to send a correct inquiry to the said tag and receive the reply from that tag. Therefore the attacker should be informed about tag K_j and Nf_j so it can play the role of a fake reader. Because of using K_j in m^1, the attacker cannot send correct inquiry for its desired tag through changing the sent inquiries for other tags.

Searched Tag Tracking: In order to prevent the tag tracking, some of the previous searching protocols such as [2] [12], [9], [6] and [4] have used the reply of those not searched tags to the reader. In this way, because of the large amount of the returned reply to the reader, the attacker cannot recognize whether the desired tag has been found or not. But it has to be said that this solution will bring some problems thereafter.

In the proposed protocol, because of using counting stamp, there does not exist the possibility for replying the previous inquiry of the reader for tags. By the help of man in the middle attack and enhancement of the sent CTC_i amount, attacker also cannot keep its job going, the reason goes to this matter that the amount of the CTC_i have been used before in reader to make the m^1 and by changing in CTC_i the tag will refrain the inquiry. Therefore, the only way for tag tracking is to eavesdrop to the exchanged information between reader and tags. But as the reader inquiry and tag respond has random nature so the attacker never realize which tag is required by the reader and it just understand that one tag is searching and it might be there or not which is not helpful for the tracking. It should be noted that in this method, each reader will receive one message in searching and tag collision does not occur.

Tag or Reader Forgery: The only way to make a fake reader, is having NF_j and K_j and tag forgery is possible just when the attacker holds NF_j, ID_j and SK_j. replying of the tags responses is not possible too because through sending $RN^2 \oplus K_j' || m^3$ in a new search, the reader will use m^1 to make m^4 which has been generated before and as the amount of the m^1 is changed in the new search so the reader cannot authenticate the tag.

De-synchronization Attacks: In protocols like [4] and [12], the attacker has the capability to perform de-synchronization attacks. It has to just allow the reader to send its inquiry to the tags so the common key between tag and the reader will be updated in the desired tag, now the attacker does not let the reader to receive the valid reply from that tag (by changing, decrease or interpolate the reply or completely prevents reader from receiving the reply). Therefore the desired variable is updated in tag but there is not any change in reader. In our search protocol, no key is updated inside the protocol and there is not any problem in synchronizing either.

Physical Attacks: The only attacks that cannot be prevented over any existing protocol is physical attacks. Our protocol too suffers from such attacks. Nevertheless in our protocol the enemy cannot access to all tags information just through attacking to one tag. Also by using L_i in each reader and restricting the amount of those tag's data in reader, the enemy through attacking to one reader can access to a limited tags information.

V. Performance Analysis of the Proposed Protocol

Number of inquiry sending: The proposed search protocol is server-less and it does not need a connection between reader and server. Also, the total amount of the relations between the searched tag and the reader is just 2 steps (in comparison with protocol [7] which has 3 steps for sending information). Therefore, the number of communication steps for the proposed protocol is well optimized.

Number of gates and clocks: Today inactive tags hold a maximum of 4000 gates bearing security protocols. So it is necessary to design protocols in such a way that the amount of

needed gates for implementation be at its least number. Another important specification of protocols is that tags should execute protocol with least number of clocks since reduction in the number of needed operational clocks can result in tags saving power.

Table 1 illustrates complete comparisons about security and the performance of the various search protocols.

Table1. Security and performance comparisons with various search protocols (H: hash function, R: random number generator, E: symmetric encryption, P: PUF function, LF: LFSR function and NLF: NLFSR function

Attacks/Refs	Ref. [2]	Ref. [12]	Ref. [5]	Ref. [6]	Ref. [9]	Ref. [4]	Proposed Protocol
Mutual authentication	No	Yes	Yes	Yes	No	Yes	Yes
Resistance to reader reply enquiry	No	Yes	No	No	No	Yes	Yes
Resistance to reader tracking	No	Yes	Yes	No	Yes	Yes	Yes
Resistance to tag tracking	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Resistance to tag forgery	No	Yes	Yes	Yes	Yes	Yes	Yes
Resistance to de-synchronization attack	Yes	No	Yes	Yes	Yes	No	Yes
Resistance to physical attacks	No	No	No	No	No	No	No
Not needing to remove tag collision	No	No	No	No	No	No	No
Desired tag calculation	3H+1 R	2H+2 R	3E+1 R	3H+1 R	2H+1 R	4LF+ 2P	3NLF+1R
Number of enquiries	2	2	2	3	2	2	2

V. Performance Analysis of the Proposed Protocol

Number of inquiry sending: The proposed search protocol is server-less and it does not need a connection between reader and server. Also, the total amount of the relations between the searched tag and the reader is just 2 steps (in comparison with protocol [7] which has 3 steps for sending information). Therefore, the number of communication steps for the proposed protocol is well optimized.

Number of gates and clocks: Today inactive tags hold a maximum of 4000 gates bearing security protocols. So it is necessary to design protocols in such a way that the amount of needed gates for implementation be at its least number. Another important specification of protocols is that tags should execute protocol with least number of clocks since reduction in the number of needed operational clocks can result in tags saving power.

In [15] a complete comparison for gate numbers and the required clocks to perform SHA-256, SHA-1, MD5, MD4 and AES algorithms are executed. Here, in making comparison with the search protocols, this issue is helpful that the least amount of the gates and clocks of those protocols which use these algorithms are identical to the gates and clocks for the establishment of the above mentioned algorithms.

In our proposed protocol, we just deployed NLFSR and function f. In the authentication protocol [5], the required amount of gates for 64 bit case is about 2600 gates and because each tag has used just these two functions so these two protocols are identical in gate numbers. Therefore, for 64bit case, the amount of the needed gates for the implementation of our protocol will be about 2600 gates.

In our protocol, each searched tag needs to use the f function and NLFSR for 3 times and a random number generator for 1 time, and other tags also need to 2 times using NLFSR and f. Regarding [1], in which the clock amount for deploying one NLFSR, f and random number generation is about 140 clocks, so it is possible to calculate the amount of the needed clocks in our protocol for the searching tag is 300 clocks and it would be 150 clocks for other tags. In protocol of [4], the searched tag needs 600 clocks and other tags need about 200 clocks. Table 2 shows completely these comparisons.

Table2. Number of needed gates and clocks in various search protocols

Protocol	Number of Gates	Number of Clock Cycles
SHA-256	10868	1128
SHA-1	8120	1274
MD5	8400	612
MD4	7350	456
AES	3400	1032
[4] with MPUF (desired tags)	620	600
[4] with MPUF (other tags)	620	200
Proposed Protocol (desired tags)	2600	300
Proposed Protocol (other tags)	2600	150

VI. Hardware Implementation of the Proposed RFID System

Fig. 5 illustrates structure of the practically implemented RFID system for the proposed protocol. The prototype of the tested tag and reader is also shown in Fig. 6. In order to process information, this system took advantages from microcontroller processors (AVR) from Atmel Company and for sending and receiving information, it deploys the transmitter and receiver modules of the Hope Microelectronic company. The implemented system shows that the protocol works correctly and it can search a tag in high speed and high security as it has been proved earlier. The results for various tests showed that the proposed system is secure against many known attacks such as eavesdropping, replying of reader inquiry and tag message and de-synchronization attack [13-16].

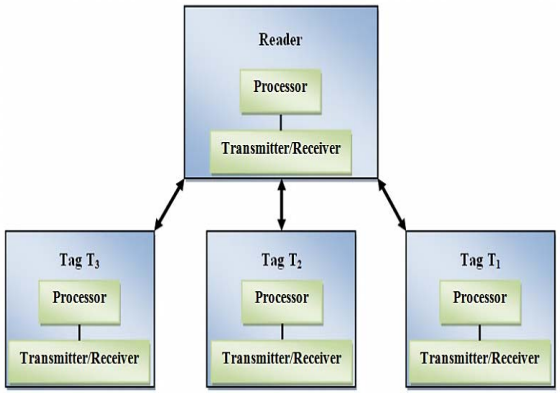

Fig. 5. Structure of implemented RFID system

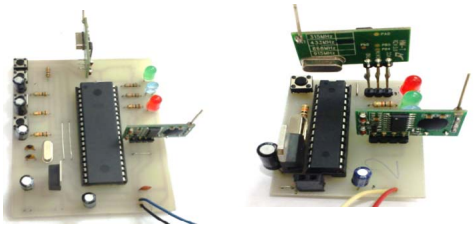

Fig. 6. The implemented tag and reader

VII. Conclusion

In this paper, a RFID light-weight search protocol is proposed that does not require a server and just by deploying NLFSRs it can resist against known security attacks. Also, because of using counting stamp, it removes the need to exclude the collisions between tags, and the search process is performed with higher speed than other protocols. By hardware implementation, it was observed that, this protocol is more optimized than most of the protocols previously established. It can be easily implemented on an AVR micro controllers or any cheap chipsets. How to establish NLFSR with maximum period and also elimination of the tag calculation numbers are now the subjects for the foregoing works. Also, the resisting to the physical attacks or decreasing the effect of these attacks can be regarded as a useful topic for the future investigations.

VIII. References

[1]. S. Myneni, S. Misra and G. Xue, "SAMA: Server-less Anonymous Mutual Authentication for Low-Cost RFID Tags", IEEE International Conference on Communication, pp. 1-5, 2011.

[2]. C. C. Tan, B. Sheng and Q. Li, "Secure and Server-less RFID Authentication and Search Protocols", IEEE Transactions on Wireless Communications, vol. 7, pp. 1400-1407, 2008.

[3]. I. C. Lin, S. C. Tsaur and K. P. Chang, "Lightweight and Server-less RFID Authentication and Search Protocol", Second International Conference on Computer and Electrical Engineering, vol. 2, pp. 95-99, 2009.

[4]. L. Kulseng, Z. Yu, Y. Wei and Y. Guan, "Lightweight Secure Search Protocols for Low-Cost RFID Systems", 29th IEEE International Conference on Distributed Computing Systems, pp. 40-48, 2009.

[5]. J. Y. Chun, J. Y. Hwang and D. H. Lee, "RFID tag search protocol privacy of mobile reader holders", IEICE Electronic Express, vol. 8, no. 2, pp. 50-56, 2011.

[6]. M. E. Hoque, F. Rahman, Sh. I. Ahamed and J. H. Park, "Enhancing Privacy and Security of RFID System with Server-less Authentication and Search Protocols in Pervasive Environments", Wireless Pers Communications, vol. 55, pp. 65-79, 2009.

[7]. T. Y. Won, J. Y. Chun and D. H. Lee, "Strong Authentication Protocol for Secure RFID Tag Search Without Help of Central Database", IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, vol. 2, pp. 153-158, 2008.

[8]. H.-Y. Chien, "SASI: A New Ultra-light weight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity", IEEE Transactions on Dependable and Secure Computing, vol. 4, pp. 337–340, 2007.

[9]. Z. Kim, J. Kim, K. Kim, I. Choi and T. Shon, "Untraceable and Server-less RFID Authentication and Search Protocols", Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops, pp. 278-283, 2011.

[10]. E. Dubrova, M. Teslenko and H. Tenhunen, "on Analysis and Synthesis of (n,k)-Non-linear Feedback Shift Registers", In Proceedings of the Conference on Design, Automation and test in Europe(DATE), pp. 1286-1291, 2008.

[11]. S. Misra, M. Verma, D. Huang and G. Xue, "SEAS: A Secure and Efficient Anonymity Scheme for Low-Cost RFID Tags", IEEE International Conference on Communication, pp. 1-6, 2009.

[12]. C. F. Lee, H. Y. Chien and C. S. Laih, "Server-less RFID authentication and searching protocol with enhanced security", International Journal of Communication Systems, vol. 25, pp. 376-385, 2012.

[13]. H. Jannati and A. Falahati, "Cryptanalysis and enhancement of two low cost RFID authentication protocols", International Journal of UbiComp (IJU), vol. 3, no.1, pp. 1-9, January 2012.

[14]. A. Juels, "RFID security and privacy: A research survey", IEEE Journal on Selected Areas in Communications, vol. 24, pp. 381-394, 2006.

[15]. M. Feldhofer and C. Rechberger, " A Case Against Currently used Hash Functions in RFID Protocols", In On the Move to Meaningful Internet Systems 2006:OTM'06 Workshops, including the First International Workshop on Information Security, LNCS4277, pp. 372-381, 2006.

[16]. H. Jannati and A. Falahati, "An RFID Search Protocol Secured Against Relay Attack Based on Distance Bounding Approach", Wireless Personal Communications, Volume 85, Issue 3, pp 711-726, 2015.