Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

Improving the Security of a Low Cost Tag Search Protocol

S. Saderi Oskuiee, F. Moazami*, G. Oudi Ghadim

Department of Content Transfer Technology, Cyberspace Research Institute, Shahid Beheshti University, Tehran, Iran.

Article Info

Article History:
Received 21 May 2020
Reviewed 07 July 2020
Revised 11 September 2020
Accepted 12 November 2020

Keywords:
RFID systems
Reader
Tag
Server
Search protocol
Eavesdropping
Tracing

*Corresponding Author's Email Address:
f_moazemi@sbu.ac.ir

Abstract

Background and Objectives: Radio Frequency Identification (RFID) systems use radio frequency waves to exchange information between a legitimate sender and a receiver. One of the important features of RFID systems is to find and track a specific tag among a large number of tags. Numerous works have been done about authentication and ownership protocols, but the number of researches done in the tag searching area is much less. Although security is a paramount factor in search protocols, but these days designers are looking for a secure search protocol that is also low cost. One way to have a low cost search protocol is that to be compatible with EPC C1G2 standard, which is an electronic product code class 1 generation 2 that works in the 860-960 MHz frequency range.

Methods: Most recently, Sundaresan et al. have proposed an RFID tag search protocol based on quadratic residues and 128-bit pseudo random number generators and XOR operation that can be easily implemented on passive tags and is compatible with EPC C1G2 standard. We show that this protocol is not immune against tag tracing, and try to improve the protocol in a way that traceability attack will not be applicable and the protocol stays low cost and EPC compatible.

Results: Since the problem in Sundaresan et al.'s search protocol is due to the tag not being able to recognize the used queries from the new ones, we improved the protocol using a counter within the queries, so the tag will realize that the query is used or not. Then we analyze the security of the improved protocol and prove its formal and informal security against known attacks.

Conclusion: In this paper, we firstly analyze the security of Sundaresan et al.'s search protocol and show that the search protocol is vulnerable to traceability attack with two different scenarios. Then we propose an improved search protocol that is secure against tracing the tags. Following that, we analyze the security of the improved search protocol.

Introduction

Radio Frequency Identification (RFID) system is a wireless technology that consists of three key parts: tags, readers, and a back-end server [1].

RFID system uses electronic and electromagnetic waves to make a conversation between a tag and a reader. A tag has an antenna and uses it for receiving and transmitting a Radio Frequency (RF) signal. It also has an integrated circuit that modulates and demodulates these signals [2].

RFID system can easily search for a particular tag among other tags by using RF signals. A reader will send its request for finding a specific tag and the wanted tag will answer the reader's request if it was the wanted tag.

An RFID system could use an authentication protocol to find a specific tag. But if the number of tags in the searching area increases, it will be costly to use an authentication protocol [3], so it is better to use a search protocol if we want to find a specific tag among a vast number of tags. Early works in tag searching area were based on cryptographic functions such as hash function [3], AES [4] and symmetric encryption function [5], meaning the reader sends its request in the form of a cryptographic function and the tag also answers to reader's request with the same function.

One of the first works in the tag searching area is the serverless search protocol that has presented by Tan et al. [3]. In their protocol, tags should only answer to the requests of authenticated readers, and readers should only query authenticated tags. In the proposed protocol, the reader queries the tag using a hash function, and the tag responds to the reader using a hash function as well. The authors state that an adversary can identify the tag in this search protocol. To improve the search protocol, authors suggested different solutions. At first, they oblige the reader to use a different random number within each query and also made the tags to store the received random numbers from previous queries. Authors also state that an opportunistic adversary can track at least one tag after a specific number of queries. The second solution is to query the first m bits of tag's id (id_j) alongside with reader's and tag's random numbers n_r, r_r . Then the tag checks m bits of its own id with the one in the reader's query. If it holds, it will answer to the query. This solution does not work well when the id for each tag is structured (it means few first bits of the id is for product code, and the next bits are for the tag's origin). Another solution is when each tag receives the query from the reader, it checks the query and if the query does not belong to that specific tag, it will answer the query by probability of λ. This way, the adversary will not be able to realize if the wanted tag is present in that area or not. This protocol uses hash functions, so it is not accordant with EPC C1G2 standard. In 2011, Kim et al. [6] showed that Tan et al.'s search protocol is vulnerable against reader tracing, and then they proposed a serverless search protocol. In their protocol, the server provides an access list for each reader, and each access list contains the group of tags that are authorized to search. The proposed protocol has three stages: setup, authentication, and search. At the first stage, the server gives an access list of the tags that are authorized to search to the reader. Tags are divided evenly into small groups by server and each group has its own identity that is set by the server. Search protocol uses hash functions for queries and answers. Sundaresan et al. stated in their papers that an adversary easily could impersonate the reader and compromise tag location privacy [7], [8], [9].

In 2012, Safkhani et al. [10] showed that Tan et al.'s search protocol [3] is vulnerable against traceability attack. They stated that in the first and third search protocols a tag could be traced.

In 2009, Lin et al. [11] proposed a serverless authentication and search protocol, which was simply an improved version of Tan et al.'s [3] search protocol. They use hash function within their protocol and aim to reduce the amount of computation in Tan et al.'s protocol. The proposed protocol is vulnerable to replay and impersonation attacks [12]. Won et al. [4] proposed a search protocol that was based on 128-bit AES cryptography function and timestamp, without a central database. In this protocol, the reader encrypts the query with 128-bit AES and sends it with a timestamp to the group of tags, tags that receive the query will check the timestamp. This way, the tags will be sure that it is not the query of the previous sessions. If the timestamp of the received query is smaller than timestamp of previous query it shows that there was a replay attack, and the tag denies the query. Otherwise, the query is correct and the tag will decrypt the query. Then, if the query is sent for that specific tag, it will answer to it. The answer to the query will also be in an encrypted form. Since all the queries are in an encrypted form, an adversary cannot get any information by eavesdropping. This protocol prevents illegal tracking of the tags and also provides secure privacy for them. DoS and de-synchronization attacks are impossible in this protocol. Since this protocol uses one way hash functions, it is not compatible with EPC C1G2 standard.

Ahamed et al. [13] proposed a serverless tag search protocol. They presented three search protocols but claim that the last one is immune against the attacks applicable to the previous two protocols. They used a pseudo random number generator P that takes a seed as an argument and a function M that generates the next random number. The reader generates a random number with P and sends it to the tag. The tag checks the validity of the received number. If the query is valid, the tag updates its own number using M. Otherwise, it will replay with probability of λ. They state that the proposed search protocol is immune against tracking, de-synchronization, and also cloning attacks.

Zuo proposed a search protocol in which the reader uses a hash function and a shared secret key to encrypt its own hashed random number along with the wanted tag's id, and the tag evaluates the query by decrypting it with its own secret key [14]. He used noise tags to guarantee that there would be an answer to the received queries. In Zuo's protocol, important secrets and data are stored in the reader. If an attacker stoles the portable reader, he can perform cloning and

impersonation attacks [8].

In 2011, Chun et al. [5] proposed a search protocol that uses symmetric encryption function. Yoon showed that this protocol is vulnerable against DoS attacks [15]. Also, Chun's protocol is not compatible with EPC C1G2 standard.

Mtita et al. [12] proposed a serverless mutual authentication and search protocol in which the reader firstly downloads a list of tags from the server that are authorized to search.

Then it uses an HMAC function and a timestamp to query a tag and the tag answers the query using a random number and an HMAC function as well. However, Sundaresan et al. stated that their protocol is susceptible to DoS and de-synchronization attacks [9]. Since Mtita's protocol uses HMAC, it is not compatible with EPC C1G2 standard.

Some of the researchers tried to propose lightweight search protocols that use lightweight functions such as Physical Unclonable Function (PUF) [16], Linear Feedback Shift Register (LFSR)[16] , and Nonlinear Feedback shift Register (NLFSR)[16].

Kulseng et al. [16] use LFSR to generate random numbers when sending a query and also when tag answers the query, and use PUF to authenticate the tags. Later on, Lv et al. showed that Kulsang et al.'s search protocol is vulnerable against tracing attack [18].

In 2019, Eslamnezhad namin et al. [19] proposed a lightweight search protocol that uses an encryption technique called Authentication Encryption (AE), that guarantees confidentiality and integrity at the same time. In their search protocol to query a tag, the reader firstly increases its counter and generates a random number and uses XOR to hide the wanted tag's id, the shared key between the reader and the tag, the random number and also encrypts the computed value and the counter. The tag evaluates the received query by decrypting the received query and checks if the id is also a valid one. If the query is not valid, the tag will answer with probability of λ. Although using these cryptographic and lightweight functions in tag search protocols made them secure against some of the possible attacks on RFID protocols [20], such as eavesdropping, physical attacks, DoS attacks, and traceability.

They were not compatible with EPC C1G2 standard. To have a low cost search protocol that is compatible with EPC C1G2 standard, the tag that concerns us is the passive one, and since it has no battery inside, it is cheaper and simpler [9].

EPC C1G2 standard is an electronic product code class 1 generation 2 that works in the 860-960MHz frequency range [21]. Besides, this standard uses FHSS (Frequency Hopping Spread Spectrum). FHSS is a method of transmitting radio signals by rapidly switching a carrier among many frequency channels. When a radio signal is transmitted, it can read the tags at slightly different frequencies to get the best possible read from the tags. Protocols that are compatible with EPC C1G2 standard use simple functions, so they are low cost protocols.

Recently, some of the researchers favor protocols that do not utilize hash functions and are compatible with EPC C1G2 standard. In 2012, Sundaresan et al. [7] proposed a search protocol that is based on quadratic residues and 128-bit pseudo random number generators and XOR operation that can be easily implemented on passive tags and is compatible with EPC C1G2 standard. But it is not secure enough. They showed later in 2017 [9] that it is not forward secure. In 2015, Sundaresan et al. [8] proposed another search protocol that was based on a 128-bit pseudo random number generator and XOR operation and was compatible with EPC C1G2 standard, but in 2016 Jannati and Bahrak showed that it was vulnerable against de-synchronization and impersonation attacks, and tag location privacy is not satisfied [22]. Also, in 2018, Eslamnejhadnamin et al. [23] showed that the search protocol proposed by Sundaresan et al. [8] is not safe against traceability attack.

In 2017, Sundaresan et al. [9] proposed a search protocol that is based on quadratic residues and 128-bit pseudo random number generator and XOR operation that can be easily implemented on passive tags and is compatible with EPC C1G2 standard. It was the improvement of their work from 2012 [7]. We will show in this paper that this protocol is vulnerable to traceability attack. The rest of the paper is organized as follows. We will briefly review the Sundaresan et al.'s latest search protocol. Then, we will propose a traceability attack on Sundaresan et al.'s search protocol with two different scenarios. In the next section, we will propose an improvement on their protocol. Then we analyze the security of the improved search protocol. The last section concludes the paper.

Review of Sundaresan search protocol

Sundaresan et al. [9] proposed a search protocol that is based on quadratic residues by using basic MOD, XOR, and 128-bit PRNG operation. For additional security, the protocol hides the random number generated by PRNG function in queries. The proposed protocol is consists of two phases – the setup phase and the secure search phase. In the setup phase, the server gives the reader an access list AL which contains the tags that are authorized to search. The second phase is where a secure search happens with the proposed protocol. Table 1 shows notations that are used in proposed protocol. In this section, we describe two phases of the protocol in details as follow:

A. Setup Phase

Assume the channel between server and reader is safe, and an adversary cannot obtain any information from conversation between the server and the reader. Server S, at first, authenticates the reader R and then gives an access list AL to the reader. This list contains all the tags that are authorized to search by the reader. This list does not include any information about secrets of the tag and its ID. And the reader only has $h(TID, t_s)$. The server also determines rts, which is a shared secret between the tag and the reader.

The reader has to store $m = g * h$ (g and h are two big prime numbers) and $h(RID)$. The tag also has to store hashed form of tag ID, and the random number s, and $n = p * q$ (p and q are two big prime numbers). The server stores TID, $h(TID)$, RID, $h(RID)$, t_s, and prime numbers p, q, g, and h, current and previous shared secret between the server and the tag s, s^{-1} and also ctr and $ctrmax$.

Table 1: Notations

Notations	Descriptions
R, S, T	Represents Server, Reader and Tag respectively
AL	Access List for the Reader
$TID, h(TID)$	Unique Tag ID and hash value of TID
$RID. h(RID)$	Unique Reader ID and hash value of RID
t_s	Secret key unique for each tag in the system, used to generate id =h(TID,ts); known only to the server
s, s^{-1}	Random number generated by server and previous value of s
p, q, g, h	Four large prime numbers generated by the server
m, n	m=g.h stored in reader and n=$p.q$ stored in the tag
$\boldsymbol{k}$	Number of the readers that can access a tag
l	Number of tags a reader is authorized to search
R_{TID}	Computed as R_{TID}=$h(TID)\oplus s$
R_{TID}^{-1}	Computed as R_{TID}^{-1}=$h(TID)\oplus s^{-1}$
rts, rts^{-1}	Shared secret between reader and a tag; previous value of rts
r_r, δ	Random numbers generated by the reader
t_r	Random number generated by the tag
$ctr, crmax$	Current and maximum value for counter
λ	Probability that a tag replies if query is not for that tag
$\oplus, \|\|$	Exclusive-OR Function (XOR) and Concatenation of two values

ctr is 0 at the beginning and will increase by 1 after each successful search and when it reaches the $ctrmax$, the reader has to get another search authorization from the server.

B: Search Phase

The search phase is shown in Fig. 1, and it has six steps. In this phase, the reader sends its query, x and y, to the tag. If the tag is the wanted tag, it answers with α'' and t_r'' and if it is not the wanted tag, it answers with a random number with probability of λ.

Then reader sends $\alpha'', t_r'', \mu'', \delta'', r_r''$ to the server. Server validates the reader and the tag, and sends ACK' to the reader. Reader validates ACK', and sends ACK to the tag.

Traceability Attack on Sundaresan et al. Protocol

In this section, we show that traceability attack is applicable on Sundaresan et al. [9] search protocol with two different scenarios. In the first scenario, an adversary will listen to the reader's query and saves x and y, and will block tag's response to the reader. The reader will not receive the response and will not update its own rts. At the moment, the tag will update its own rts^{-1} and rts as follows:

$$rts \rightarrow rts^{-1}, \quad (1)$$

$$PRNG(rts) \rightarrow rts. \quad (2)$$

The adversary sends the captured queries to the tags. The target tag will check the validity of captured x and y sent by the adversary with both rts^{-1} and rts. Hence it will always consider them as valid queries, therefore the target tag responds to the requests with probability of 1. On the other hand, tags that are not the wanted tag will replay to the requests with probability of λ.

In this scenario, which is inspired by Eslamnejhadnamin et al.'s work [23], suppose that there are N tags in the reader's searching area, the adversary will send a captured valid queries m times to these tags. Assume the random variable X denote the number of received answers by the adversary from the tags. If the wanted tag is not present, then the random variable X follows the binomial distribution with parameters $m.(N-1)$ and λ. Hence the expected value of X is $m.(N-1).\lambda$ and the variance is $m.(N-1).\lambda.(1-\lambda)$. The adversary counts the number of received answers. If for a real number k, $|X - m.(N-1).\lambda| < k\sigma$, then the adversary concludes that the target tag is not present. To find the optimal value of k and m such that traceability attack performs with high probability, the adversary can use Chebyshev's inequality.

By Chebyshev's inequality, if X is a random variable with expected value μ, and non-zero variance σ^2, then for any real number $k > 0$, we have:

$$\Pr(|X - \mu| < k\sigma) \geq 1 - \frac{1}{k^2}.$$

So, by choosing the appropriate value of parameters k and m, the opportunistic adversary can be successful with high probability.

In the second scenario, which is inspired by Jannati and Bahrak's attack [22] on Sundaresan et al. [8], to perform the traceability attack, we assume the adversary

Server	Reader	Tag
	Generates random number r_r Computes: $x = id_j \oplus PRNG(rts_j \oplus r_r)$ $y = PRNG(id_j \oplus rts_j) \oplus r_r$	Extracts r_r : $PRNG(id \oplus rts) \oplus y \rightarrow r_r$ If $id = x \oplus PRNG(rts \oplus r_r)$ Query is valid
Computes μ, δ, r_r with g, h	x , y →	Generates random number t_r
If r_r isn't fresh aborts the protocol If $\mu \oplus \delta \oplus r_r = h(RID)$		Computes: $\alpha = h(TID) \oplus r_r \oplus s \oplus t_r$ $\alpha' = \alpha^2 \bmod n$ $\alpha'' = (\alpha^2)^2 \bmod n$ $t_r'' = (t_r{}^2)^2 \bmod n$
Reader is valid If $ctr \leq ctrmax$ Computes α, t_r with p, q If $\alpha \oplus r_r \oplus t_r = R_{TID} or\ R_{TID}^{-1}$ Tag is valid, if no aborts the protocol.		else restarts this step using rts_j^{-1}
If server identifies tag with R_{TID} then: $s \rightarrow s^{-1}$ $PRNG(s) \rightarrow s$		If both rts and rts^{-1} do not return valid id
	Generates random number δ Computes: $\mu = h(RID) \oplus r_r \oplus \delta$ $\mu' = \mu^2 \bmod m$ $\mu'' = (\mu^2)^2 \bmod m$ $\delta'' = (\delta^2)^2 \bmod m$	Answers with probability of λ If id was matched using rts then: $rts \rightarrow rts^{-1}$ $PRNG(rts) \rightarrow rts$
Computes: $ACK = h(TID) \oplus t_r \oplus PRNG(\alpha \oplus \alpha')$ $ACK' = (ACK \parallel \delta) \oplus h(\mu)$ Increments the ctr variable by 1	$r_r'' = (r_r^2)^2 mod m$ $\alpha'', t_r'', \mu'', \delta'', r_r''$ ←	α'', t_r'' ←
If $ctr \geq ctrmax$ Server should authorized the reader		
ACK' →	If δ is valid in $ACK' \oplus h(\mu)$ $ACK = [ACK' \oplus h(\mu)] \gg l$ Updates rts_j as: $rts_j = PRNG(rts_j)$	
	ACK →	If: $ACK = h(TID) \oplus t_r \oplus PRNG(\alpha \oplus \alpha')$ Updates s as: $PRNG(s) \rightarrow s$ Otherwise aborts the protocol

Fig. 1: Sundaresan et al. search protocol [9].

eavesdrops on the communication between the reader and the tag. Since the channel between reader and tag is insecure, adversary can store reader's queries $x = id_j \oplus PRNG(rts_j \oplus r_r)$ and $y = PRNG(id_j \oplus rts_j) \oplus r_r$ and block tag's respond.

So, the reader will not receive the respond and will not renew its own rts and at the moment, the tag will update rts^{-1} and rts. To trace a specific tag, the adversary will send reader's previous queries to tags and receive responses from them.

The target tag will check the validity of captured x and y sent by the adversary with both rts^{-1} and rts. Hence it always will consider them as valid queries, and if a tag does not answer, then the adversary concludes that it is a noise tag. Adversary repeats this process and tries to find the wanted tag.

Assume N tags exist in the reader's searching area. We know that noise tags will answer the query with probability of λ. Also, suppose that T_1 is the wanted tag and $\mathrm{T}_2, \mathrm{T}_3, \ldots, \mathrm{T}_\mathrm{N}$, are noise tags, and all of them are available.

Let an adversary sends the captured queries m times, and let A_i be the event that T_i answers to all the received queries, hence $\Pr(\mathrm{A}_\mathrm{i}) = \lambda^\mathrm{m}$. In this scenario, the traceability attack will fail if at least one of the noise tags answers all received queries. On the other hand, the probability of the attack's failure is equal to $\Pr(\mathrm{A}_2 \cup \ldots \cup \mathrm{A}_\mathrm{N})$.

Also, we have:

$$\Pr(\mathrm{A}_2 \cup \ldots \cup \mathrm{A}_\mathrm{N}) \leq \sum_{\mathrm{i}=2}^{\mathrm{N}} \Pr(\mathrm{A}_\mathrm{i}) = (\mathrm{N}-1) \quad (3)$$

So, if the adversary wants to limit the probability of the attack's failure to γ, the number of queries that has to send is $\mathrm{m} = \left\lceil \frac{\log(\gamma) - \log(\mathrm{N}-1)}{\log(\lambda)} \right\rceil$.

In this section, we show that Sundaresan et al. [9] search protocol is unprotected against traceability attack. In the next section, we will improve their protocol to prevent tracing the tag.

The traceability attack is applicable because the tag cannot recognize if the received query is reused or not. So, to improve Sundaresan et al. [9] search protocol and prevent tractability attack, we use a counter within the queries. This way the tag will recognize if a query is fresh or not.

Proposed Improved Search Protocol

The setup phase in improved protocol is like Sundaresan et al. [9] search protocol's setup phase. Also, in this phase server gives a counter (ctr_j) to the reader. The tag and the reader will increase the amount of counter after each query. After setup phase, reader starts to search for the wanted tag. At first, reader will increase its counter by 1 ($ctr_j = ctr_j + 1$), then calculates A, B and C as follow:

$$\boldsymbol{A} = id_j \oplus PRNG(rts_j \oplus r_r) \oplus PRNG(ctr_j \oplus r_r) \quad (4)$$

$$\boldsymbol{B} = PRNG(id_j \oplus rts_j) \oplus r_r \quad (5)$$

$$\boldsymbol{C} = ctr_j \oplus PRNG(r_r \oplus id_j) \quad (6)$$

Reader then sends $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ to the tags in the area. Each tag will extract pseudo random number from received $\boldsymbol{B}$ using its own id and rts:

$$PRNG(id_j \oplus rts_j) \oplus \boldsymbol{B} \rightarrow r_r. \quad (7)$$

Then the tag will extract the counter from received $\boldsymbol{C}$ using extracted pseudo random number and its own id:

$$\boldsymbol{C} \oplus PRNG(r_r \oplus id_j) \rightarrow ctr_j. \quad (8)$$

If the counter (ctr_j) extracted from query is smaller than the one in tag's memory, the tag will notice that either replay or impersonation attack took place. Otherwise, tag will check if the id in $\boldsymbol{A}$ is equal to its own id as follows:

$$id_j = \boldsymbol{A} \oplus PRNG(rts_j \oplus r_r) \oplus PRNG(ctr_j \oplus r_r) \quad (9)$$

If above check fails, tag will repeat these steps using rts_j^{-1}. If tag's id didn't match using both rts and rts_j^{-1}, then the tag will respond with probability of λ. The improved search protocol is shown in Fig. 2.

Informal Security Analysis

In this section, we informally prove that our improved search protocol is resistant to replay, traceability, de-synchronization, and Dos attacks. We also indicate that our improved protocol provides tag and reader anonymity and location privacy.

A. *Replay Attack*

To perform a replay attack, the adversary stores legitimate queries sent from the reader during a session then uses this information to query the tag. Since both the reader and the tag use random numbers and encipher them properly, and the reader uses a counter (ctr) to create a query. Also, both the reader and the tag update their counter after a query. So, the tag will understand that if the received message is fresh or not and will answer the old query with probability of λ.

So, the replay attack is not applicable, since every legitimate query will be fresh.

B. *Traceability Attack*

The weakness of Sundaresan et al. [9] search protocol is that the freshness of the queries is not guaranteed. In our improved search protocol, the reader uses a counter (ctr) in each message and increases its counter by 1 after each query.

The wanted tag also increases its counter after receiving a legitimate query. Thus, each query will be fresh, and the attacker will not notice that if the wanted tag is present or not.

So, performing a traceability attack is impossible, and tag location privacy is satisfied.

C. *Tag Anonymity*

In improved search protocol, the attacker cannot detect the value of tag's unique id, since it is well hidden in $h(TID, t_s)$. So, the adversary cannot obtain any information about the tag, and its anonymity is guaranteed.

D. De-synchronization Attack

If the adversary blocks answer from the tag $\{\alpha'', t_r''\}$ or the answer gets lost during the communication, this way the tag updates its own rts but the reader does not, and this causes de-synchronization of the keys. Since the tag saves its own rts_j^{-1} from the last session and checks the validity of the received query with it, a de-synchronization attack is not applicable.

E. DoS Attack

If an attacker blocks the message $\{ACK\}$ or forges it, it can cause de-synchronization between the tag and the server leading to Dos attack. If the message $\{ACK\}$ is blocked, it cannot lead to DoS attack.

Since the server keeps both s and s^{-1}, and it will be able to validate the next answers by R_{TID}^{-1}.

Server	Reader	Tag
	Generates random number r_r Increments the ctr variable by 1 Computes: $\boldsymbol{A} = id_j \oplus PRNG(rts_j \oplus r_r) \oplus PRNG(ctr_j \oplus r_r)$ $\boldsymbol{B} = PRNG(id_j \oplus rts_j) \oplus r_r$ $\boldsymbol{C} = ctr_j \oplus PRNG(r_r \oplus id_j)$	Extracts r_r : $PRNG(id \oplus rts) \oplus \boldsymbol{B} \rightarrow r_r$ $\boldsymbol{C} \oplus PRNG(r_r \oplus id_j) \rightarrow ctr_j$
Computes μ, δ, r_r with g, h		If it's lower than tag's own counter then impersonation attack took place
If r_r isn't fresh aborts the protocol If $\mu \oplus \delta \oplus r_r = h(RID)$		If $id = \boldsymbol{A} \oplus PRNG(rts_j \oplus r_r) \oplus PRNG(ctr_j \oplus r_r)$
	$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ →	Query is valid Generates random number t_r Computes: $\alpha = h(TID) \oplus r_r \oplus s \oplus t_r$ $\alpha' = \alpha^2 \quad mod\ n$ $\alpha'' = (\alpha^2)^2 \quad mod\ n$ $t_r'' = (t_r{}^2)^2 \quad mod\ n$
If $ctr \leq ctrmax$ Computes α, t_r with p, q If $\alpha \oplus r_r \oplus t_r = R_{TID}$ or R_{TID}^{-1} Tag is valid, if no aborts the protocol. If server identifies tag with R_{TID} then: $s \rightarrow s^{-1}$ $PRNG(s) \rightarrow s$		Else restarts this step using rts_j^{-1} If both rts and rts^{-1} do not return valid id
	Generates random number δ Computes: $\mu = h(RID) \oplus r_r \oplus \delta$ $\mu' = \mu^2 mod\ m$ $\mu'' = \left(\mu^2\right)^2 mod\ m$ $\delta'' = \left(\delta^2\right)^2 mod\ m$ $r_r'' = (r_r^2)^2 mod m$	Answers with probability of λ If id was matched using rts then: $rts \rightarrow rts^{-1}$ and $PRNG(rts) \rightarrow rts$
Computes: $\mathrm{ACK} = \mathrm{h(TID)} \oplus \mathrm{t_r} \oplus \mathrm{PRNG}(\alpha \oplus \alpha')$ $\mathrm{ACK}' = (\mathrm{ACK} \parallel \delta) \oplus \mathrm{h}(\mu)$ Increments the ctr variable by 1	α'', t_r'' ←	
If $ctr \geq ctrmax$ Server should authorized the reader	$\alpha'', t_r'', \mu'', \delta'', r_r''$ ←	
ACK' →		
	If δ is valid in $\mathrm{ACK}' \oplus \mathrm{h}(\mu)$ $\mathrm{ACK} = [\mathrm{ACK}' \oplus \mathrm{h}(\mu)] \gg l$ Updates rts_j as: $rts_j = PRNG(rts_j)$	
	ACK →	If: $ACK = h(TID) \oplus t_r \oplus PRNG(\alpha \oplus \alpha')$ Updates s as: $PRNG(s) \rightarrow s$ Otherwise aborts the protocol

Fig. 2: Improved Search Protocol.

To forge the message $\{ACK\}$, the attacker has to know both the reader and the tag's secrets. Since it is not possible to obtain any information about secrets as they are well hidden during the sessions, only a

legitimate server can calculate the message $\{ACK\}$. Therefore it is impossible to perform a DoS attack.

F. Reader Anonymity and Location Privacy

The reader's id is sent in a hash form $h(RID)$, and it is properly enciphered in the μ. So, if the attacker does not know the value of prime numbers g and h, he will not obtain any information. Therefore, the search protocol provides reader anonymity and location privacy.

Formal Security Analysis

In this section, we use GNY logic [24], which is a formal method to evaluate the security of improved search protocol. Table 2 shows the rules of GNY logic.

In the first step, we translate the messages of the protocol to the GNY logic parser:

Message1:R → T: ***A***:

$T\triangleleft * (id_j \oplus PRNG(rts_j \oplus r_r) \oplus PRNG(ctr_j \oplus r_r))$ (The tag T receives message ***A***).

Message2: R → T: ***B***: $T\triangleleft * (PRNG(id_j \oplus rts_j) \oplus r_r)$ (The tag T receives message ***B***).

Table 2: GNY logic rules [24]

Rules	Description
T1	A principal is being told of a "not-originated-here" formula.
P1	A principal is capable of possessing a formula he is told.
P2	If a principal possesses two formulas, then he is capable of possessing a function F of them.
F1	If P believes a formula X is fresh, then he is entitled to believe that any formula of which X is a component is fresh.
I1	Suppose that for principal P, all of the following conditions hold: (1) P receives a formula consisting of a X encrypted with key K and marked with a not-originated-here mark; (2) P possesses K; (3) P believes K is a suitable secret for himself and Q; (4) P believes formula X is recognizable; (5) P believes that K is fresh or that X is fresh. Then P is entitled to believe that (1) Q once conveyed X; (2) Q once conveyed the formula X encrypted with K; (3) Q possesses k.
J1	J1 states that if P believes that Q has jurisdiction over some statement C and that Q believes in C, then P ought to believe in C as well.

Message3: R → T: ***C***: $T\triangleleft * (ctr_j \oplus PRNG(r_r \oplus id_j))$ (The tag T receives message ***C***).

Message4:T → R: $\acute{\alpha}$:

$R\triangleleft * ((h(TID) \oplus r_r \oplus s \oplus t_r)^4 \bmod n))$ (The reader R receives message $\acute{\alpha}$).

Message5: T → R: $t_r^{"}$: = $R\triangleleft * ((t_r)^4 \bmod n)$ (The reader R receives message $t_r^{"}$).

Message6: R → S: $\acute{\alpha}$

$S\triangleleft * ((h(TID) \oplus r_r \oplus s \oplus t_r)^4 \bmod n))$ (The server S receives message $\acute{\alpha}$)

Message7: R → S: $t_r^{"}$: $S\triangleleft * ((t_r)^4 \bmod n)$ (The server S receives message $t_r^{"}$)

Message8:R → S: $\mu^{"}$: $S\triangleleft * ((h(RID) \oplus r_r \oplus \delta)^4 \bmod m)$ (The server S receives message $\mu^{"}$)

Message9:R → S: $\delta^{"}$: $S\triangleleft * ((\delta)^4 \bmod m)$ (The server S receives message $\delta^{"}$)

Message10:R → S: $S\triangleleft * ((r_r)^4 \bmod m)$ (The server S receives message $r_r^{"}$)

Message11: S → R: $ACK^{'}$

$S\triangleleft * (ACK^{'} = (h(TID) \oplus t_r \oplus PRNG(\alpha \oplus \alpha^{'}) \| \delta) \oplus h(\mu))$ (The reader R receives message$ACK^{'}$)

Message12: $R \rightarrow T$: ACK: $T\triangleleft * (ACK^{'} \oplus h(\mu)) \gg l$

(The tag T receives messageACK)

Then we write the assumptions used in the improved protocol that are going to use to evaluate the correctness of the protocol:

A1:R ∋ r_r : The reader R possesses r_r.

A2: $R| \equiv \#r_r$: The reader R believes that r_r is fresh.

A3: $R \ni rts$: The reader R possesses rts.

A4: $R| \equiv \#rts$: The reader R believes that rts is fresh.

A5: $R \ni ctr$: The reader R possesses ctr.

A6: $R| \equiv \#ctr$: The reader R believes that ctr is fresh.

A7: $T \ni t_r$: The tag T possesses t_r.

A8: $T| \equiv \#t_r$: The tag T believes that t_r is fresh.

A9: $T \ni rts$: The tag T possesses rts.

A10: $T| \equiv \#rts$: The tag T believes that rts is fresh.

A11: $T \ni ctr$: The tag T possesses ctr.

A12: $T| \equiv \#ctr$: The tag T believes that ctr is fresh.

A13:$R| \equiv R \overset{rts}{\leftrightarrow} T$: The reader R believes that rts is a suitable secret between the reader R and the tag T.

A14: $T| \equiv T \overset{rts}{\leftrightarrow} R$: The tag T believes that rts is a suitable secret between the tag T and the reader R.

A15: $T \ni s$: The tag T possesses s.

A16: $T| \equiv \#s$: The tag T believes that s is fresh.

A17: $S \ni s$: The sever S possesses s.

A18: $S| \equiv \#s$: The server S believes that s is fresh.

A19: $R \ni \delta$: The reader R possesses δ.

A20: $R| \equiv \#\delta$: The reader R believes that δ is fresh.

A21: $T| \equiv T \overset{t_s}{\leftrightarrow} S$: The tag T believes that t_s is a suitable secret between the tag T and the server S.

A22: $S| \equiv S \overset{t_s}{\leftrightarrow} T$: The server S believes that t_s is a suitable secret between the server S and the tag T.

Then we described the security correctness goals:

B1:$T| \equiv R| \sim \#id_j \oplus PRNG(rts_j \oplus r_r) \oplus PRNG(ctr_j \oplus r_r)$: The tag T believes that the reader R conveys the formula $(id_j \oplus PRNG(rts_j \oplus r_r) \oplus PRNG(ctr_j \oplus r_r))$.

B2: $T| \equiv R| \sim \#PRNG(id_j \oplus rts_j) \oplus r_r$: The tag T believes

that the reader R conveys the formula $(PRNG(id_j \oplus rts_j) \oplus r_r)$.

b3: $T|\equiv R|\sim\#(ctr_j \oplus PRNG(r_r \oplus id_j))$: The tag T believes that the reader R conveys the formula $(ctr_j \oplus PRNG(r_r \oplus id_j))$.

B4: $R|\equiv T|\sim\#(h(TID) \oplus r_r \oplus s \oplus t_r)^4 \ mod\ n)$: The reader R believes that the tag T conveys the formula $((h(TID) \oplus r_r \oplus s \oplus t_r)^4 \ mod\ n)$.

B5: $R|\equiv T|\sim\#((t_r)^4 \ mod\ n)$: The reader R believes that the tag T conveys the formula $((t_r)^4 \ mod\ n)$.

B6: $S|\equiv R|\sim\#(h(TID) \oplus r_r \oplus s \oplus t_r)^4 \ mod\ n)$: The server S believes that the reader R conveys the formula $((h(TID) \oplus r_r \oplus s \oplus t_r)^4 \ mod\ n)$.

B7: $S|\equiv R|\sim\#((t_r)^4 \ mod\ n)$: The server S believes that the reader R conveys the formula $((t_r)^4 \ mod\ n)$.

B8: $S|\equiv R|\sim\#\left(h(RID) \oplus r_r \oplus \delta\right)^4 \ mod\ m)$: The server S believes that the reader R conveys the formula $\left(\left(h(RID) \oplus r_r \oplus \delta\right)^4 \ mod\ m\right)$.

B9: $S|\equiv R|\sim\#((\delta)^4 \ mod\ m)$: The server S believes that the reader R conveys the formula $((\delta)^4 \ mod\ m)$.

B10: $S|\equiv R|\sim\#((r_r)^4 \ mod\ m)$: The server S believes that the reader R conveys the formula $((r_r)^4 \ mod\ m)$.

B11:$R|\equiv S|\sim\#(h(TID) \oplus t_r \oplus PRNG(\alpha \oplus \alpha') \parallel \delta) \oplus h(\mu))$: The reader R believes that the server S conveys the formula $(h(TID) \oplus t_r \oplus PRNG(\alpha \oplus \alpha') \parallel \delta) \oplus h(\mu))$.

B12: $T|\equiv R|\sim\#(\ ACK' \oplus h(\mu)) \gg l$: The tag T believes that The reader R conveys the formula $\left(\ ACK' \oplus h(\mu)\right) \gg l$.

-Proofing B1, B2 and B3:

D1: Assume that the tag is being told of messages A, B and C (T1).

D2: by D1, assume that the tag possesses A, B and C (P1).

D3: by D2, assume that the tag believes that A, B and C are fresh (F1).

D4: From D3 (after series of steps D1, D2, D3) and assumptions A2, A4, A6, and A13 and applying postulates I1 and P2, B1 is achieved.

D5: From D3 and assumptions A2, A4, A13 and applying postulates I1 and P2, B2 is achieved.

D6: From D3 and assumptions A2, A6 and applying postulates I1 and P2, B3 is achieved.

-Proofing B4 and B5:

D7: Assume the reader is being told of messages α'' and t_r''(T1).

D8: By D7, assume that the reader possesses α'' and t_r'' (P1).

D9: By D8, assume that the reader believes α'' and t_r'' are fresh (F1).

D10: B4 is achieved From D9 (after series of steps D7, D8, D9) and assumptions A2, A8, A16 and applying postulates I1 and P2.

D11: B5 is achieved From D9 (after series of steps D7, D8, D9) and A8, and applying postulates I1 and P2.

-Proofing B6, B7, B8, B9 and B10:

D12: Assume the server is being told of messages α'', t_r'', μ'', δ'' and r_r'' (T1).

D13: By D12, Assume the server is possesses messages α'', t_r'', μ'', δ'' and r_r'' (P1).

D14: By D13, assume server believes that messages α'', t_r'', μ'', δ'' and r_r'' are fresh (F1).

D15: B6 is achieved From D14 (after series of steps D12, D13, D14) and A2, A8, A16 and applying postulates I1, J1 and P2.

D16: B7 is achieved From D14 (after series of steps D12, D13, D14), A8 and applying postulates I1, J1 and P2.

D17: B8 is achieved From D14 (after series of steps D12, D13, D14), A2, A20 and applying postulates I1, J1 and P2.

D18: B9 is achieved From D14 (after series of steps D12, D13, D14) and A20 and applying postulates I1, J1 and P2.

D19: B10 is achieved From D14 (after series of steps D12, D13, D14) and A2 and applying postulates I1, J1 and P2.

-Proofing B11:

D20: Assume the reader is being told of message $\{ACK'\}$ (T1)

D21: By D20, assume that the reader possesses $\{ACK'\}$ (P1).

D22: By D21, assume that the reader believes message $\{ACK'\}$ is fresh (F1).

D23: From D22 and assumptions A8, A20 and applying postulates I1 and P2, B11 is achieved.

-Proofing B12:

D24: Assume the tag is being told of message $\{ACK\}$ (T1).

D25: By D24, assume that the tag possesses $\{ACK\}$ (P1).

D26: By D25, assume that the tag believes message $\{ACK\}$ is fresh (F1).

D27: From D26 and assumptions A8, and applying postulates I1 and P2, B12 is achieved.

Security and Performance Comparison

In this section, we compare the improved search protocol with search protocols proposed earlier.

In Table 3, we compare the improved protocol with some other search protocol in terms of hash, HMAC, PUF, PRNG, LFSR, Encryption and Decryption functions, XOR, and concatenate used in them.

We observe that Tan et al.'s search protocol [3] uses hash functions, so it cannot be a lightweight protocol. Chun et al. use symmetric encryption functions when querying a specific tag [5], so their protocol cannot be EPC compatible. Sundaresan et al. use lightweight function within their protocols [8], [9], so their protocols

can be low cost and compatible with EPC standard.

Table 3: Complexity comparison

Protocol	[3]	[5]	[8]	[9]	[16]	Proposed Protocol
H1	2	0	0	0	0	0
H2	3	2	0	1	0	1
H3	2	2	6	8	4	12
H4	0	0	0	0	3	0
H5	0	0	2	3	0	5
H6	0	0	0	0	4	0
H7	0	2	0	0	0	0

H1: Number of Hash/ H2: Number of ‖/H3: Number of ⊕/H4: Number of PUF /H5: Number of PRNG/ H6: Number of LFSR/ H7: Number of E(.) and D(.)

Kulseng et al. use PUF functions in their protocol [16], but it is hardly unlikely to implement PUF on a large scale [9]. In our improved protocol, the number of PRNG functions and XOR operators increased slightly, comparing to the Sundaresan et al. search protocol [9]. But since our protocol still uses lightweight functions, the cost of implementing it on RFID system will be low as well.

In Table 4, the improved search protocol is compared with some well-known search protocols based on various parameters such as tag and reader anonymity, tag and reader location privacy, EPC compliance and some attacks such as replay, tracing, DoS and de-synchronization.

Table 4: Security comparison

Protocol	[3]	[4]	[6]	[9]	[12]	[14]	IMPROVED PROTOCOL
Tag anonymity	S	S	S	S	S	S	S
Reader anonymity	NS	S	NS	S	S	S	S
Tag location privacy	NS	NS	NS	NS	S	S	S
Reader location privacy	NS	S	NS	S	S	S	S
Replay attack	NA	NA	NA	A	NA	NA	NA
Traceability attack	A	A	A	A	NA	NA	NA
DoS/ Desynchronization attack	NA	NA	NA	NA	A	NA	NA
EPC compatible	N	N	N	Y	N	N	Y

Applicable (A)/ Not Applicable(NA)/ Satisfied(S)/ Not Satisfied (NS)/ Yes (Y)/ No (N)

Conclusion

In this paper, we probed the security of Sundaresan et al. search protocol [9] and discovered that an adversary could trace a tag. This weakness causes from the tag not being able to recognize an old query.

We showed that traceability attack applicable and explained it with two separate scenarios.

In the first scenario, we indicated that a traceability attack is executable, and the adversary could recognize whether the wanted tag is present with high probability by applying Chebyshev's inequality.

In the second scenario, we showed that a traceability attack is applicable with high probability if the adversary sends out a sufficient amount of queries to the tags in the area.

Following that, we proposed an improvement on Sundaresan et al. search protocol [9], that uses a counter when sending a query to ensure the freshness of the legitimate requests. Then, in the last section, security analysis of the improved protocol showed that it is immune against replay, traceability, Dos and de-synchronization attacks, and tag and reader anonymity and location privacy are maintained as well.

Author Contribution

S. Saderi, F. Moazami, G. Oudi proposed traceability attack on Sundaresan et al. search protocol. Sh. Saderi, F. Moazami proposed an improvement on the Sundaresan search protocol and analyzed security and performance of improved search protocol.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments, which helped to increase the quality of our paper.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviation

R, S, T	Represents Server, Reader and Tag respectively
AL	Access List for the Reader
$TID, h(TID)$	Unique Tag ID and hash value of TID
$RID. h(RID)$	Unique Reader ID and hash value of RID
t_s	Secret key unique for each tag in the system, used to generate id= h(TID,ts); known only to the server
s, s^{-1}	Random number generated by server and previous value of s

p, q, g, h	Four large prime numbers generated by the server
m, n	m=g.h stored in reader and n=$p.q$ stored in the tag
$\boldsymbol{k}$	Number of the readers that can access a tag
l	Number of tags a reader is authorized to search
R_{TID}	Computed as R_{TID}=$h(TID)\oplus s$
R_{TID}^{-1}	Computed as R_{TID}^{-1}=$h(TID)\oplus s^{-1}$
rts, rts^{-1}	Shared secret between reader and a tag; previous value of rts
r_r, δ	Random numbers generated by the reader
t_r	Random number generated by the tag
$ctr, crmax$	Current and maximum value for counter
x, y	Queries sent from reader to tag in Sundaresan et al. search protocol
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$	Queries sent from reader to tag in our improved search protocol
$\alpha^{"}, t_r^{"}$	Wanted tag's respond to a legitimate query
$\acute{\alpha}, t_r^{"}, \mu^{"}, \delta^{"}, r_r$	Legitimate reader's answers to server
ACK′	Server's acknowledgment sent to the reader
ACK	Reader's acknowledgment sent to the valid tag
λ	Probability that a tag replies if query is not for that tag
PRNG	Pseudorandom number generator
$\oplus, \|$	Exclusive-OR Function (XOR) and Concatenation of two values
X	Random variable
μ	Expected value of X
σ^2	Finite non-zero variance of X
k	A real number
N	Assumed number of tags in reader's searching area
m	Number of captured queries sent by an adversary
T1	Being told formula
P1	First possession rule
P2	Second possession rule
F1	Freshness rule
I1	Interpretation rule
J1	Jurisdiction rule
A	Assumptions used in security analysis
B	Security correctness goals
D	Security correctness proofs

References

[1] S. Lahiri, RFID sourcebook, IBM press, 2005.

[2] V. Chawla, D.S. Ha, "An overview of passive RFID," IEEE Commun. Mag., 45(9): 11–17, 2007.

[3] C.C. Tan, B. Sheng, Q. Li, "Secure and serverless RFID authentication and search protocols," IEEE Trans. Wireless Commun., 7(4): 1400–1407, 2008.

[4] T.Y. Won, J.Y. Chun, D.H. Lee, "Strong authentication protocol for secure RFID tag search without help of central database," in Proc. International Conference on Embedded and Ubiquitous Computing, 2: 153-158, 2008.

[5] L. Chun, J. Hwang, D. Lee, "RFID tag search protocol preserving privacy of mobile reader holders," IEICE Electron. Express, 8(2): 50–56, 2011.

[6] Z. Kim, J. Kim, K. Kim, I. Choi, T. Shon, "Untraceable and serverless RFID authentication and search protocols," in Proc. Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops,: 278–283, 2011.

[7] S. Sundaresan, R. Doss, W. Zhou, "A secure search protocol based on Quadratic Residues for EPC Class-1 Gen-2 UHF RFID tags," in Proc. 23rd International Symposium on Personal Indoor and Mobile Radio Communications,:30–35, 2012.

[8] S. Sundaresan, R. Doss, S. Piramuthu, W. Zhou, "Secure tag search in RFID systems using mobile readers," IEEE Trans. Dependable Secure Comput., 12(2): 230–242, 2015.

[9] S. Sundaresan, R. Doss, S. Piramuthu, W. Zhou, "A secure search protocol for low cost passive RFID tags," Computer Networks, 122: 70–82, 2017.

[10] M. Safkhani, P. Peris-Lopez, N. Bagheri, M. Naderi, J. C. Hernandez-Castro, "On the security of Tan et al. serverless RFID authentication and search protocols," in Proc. International Workshop on Radio Frequency Identification: Security and Privacy Issues, 7739: 1–19, 2012.

[11] L.C. Lin, S.C. Tsaur, S.-C, K.P. Chang, "Lightweight and serverless RFID authentication and search protocol," in Proc. Second Int. Conf. on Computer and Electrical Engineerin, 2: 95–99, 2009.

[12] C. Mtita, M. Laurent, J. Delort, "Efficient serverless radiofrequency identification mutual authentication and secure tag search protocols with untrusted readers," IET Inf. Secur., 10(5): 262–271, 2016.

[13] S.I. Ahamed, F. Rahman, E. Hoque, F. Kawsar, T. Nakajima, "S3PR: secure serverless search protocols for RFID," in Proc. 2008 International Conference on Information Security and Assurance (isa 2008): 187–192, 2008.

[14] Y. Zuo, "Secure and private search protocols for RFID systems," Inform. Syst. Front., 12(5): 507–519, 2009.

[15] E.-J. Yoon, "Cryptanalysis of an RFID tag search protocol preserving privacy of mobile reader," in Proc. International Federation for Information Processing,: 575–580, 2012.

[16] L. Kulseng, Z. Yu, Y. Wei, Y. Guan, "Lightweight secure search protocols for lowcost RFID systems," in Proc. 2009 29th IEEE International Conference on Distributed Computing Systems,: 40–48, 2009.

[17] A. Falahati, H. Azizi, R.M. Edwards. "RFID light weight server-less search protocol based on nlfsrs," in Proc. 8th International Symposium on Telecommunications (IST),: 741-745, 2016.

[18] C. Lv, H. Li, M. Jianfeng, B. Niu, "Vulnerability analysis of lightweight secure search protocols for low-cost RFID systems," Int. J. Radio Freq. Identif. Technol. Appl., 4(1): 3–12, 2012.

[19] M. Eslamnezhad Namin, M. Hosseinzadeh, N. Bagheri, A. Khademzadeh, "RSPAE: RFID search protocol based on authenticated encryption,'' J. Electr. Comput. Eng. Innovations, 6(2): 179-192, 2018.

[20] A. Khattab, Z. Jeddi, E. Amini, E., M. Bayoumi, RFID security: a lightweight paradigm, Springer, 2016.

[21] B. Gesuale, P. Agarwal, RFID: READ MY CHIPS!. Piper Jaffray Equity Research Report, 2004.

[22] H. Jannati, B. Bahrak, "Security analysis of an RFID tag search protocol," Inf. Process. Lett., 116(10): 618–622, 2016.

[23] M. Eslamnezhad Namin, M. Hosseinzadeh, N. Bagheri, A. Khademzadeh, "A secure search protocol for lightweight and low-cost RFID systems," Telecommunication Systems, 67(4): 539–552, 2018.

[24] L. Gong, R. Needham, R. Yahalom, "Reasoning about Belief in Cryptographic Protocols," in Proc. 1990 IEEE Computer Society Symposium on Research in Security and Privacy: 234–248, 1990.

Biographies

Shabnam Saderi Oskuiee received her M.Sc. degree in secure communication and cryptography from Shahid Beheshti University, Tehran, Iran, in 2018 and B.S. degree in communication engineering from Tabriz University, Tabriz, Iran, in 2015. Her research interests include RFID security and privacy.

Farokhlagha Moazami is an assistant professor at the Cyber Space Research center at Shahid Beheshti University, Tehran, Iran, since 2013. She received B.S. and Ph.D. degrees in mathematics fromIzahra University, Tehran, Iran, in 2004 and 2012, respectively and M.S. degree in mathematics from Sharif University of Technology, Tehran, Iran, in 2006. She was a postdoctoral at Sharif University of Technology, Tehran, Iran, from 2012 to 2013. Her main research interest is theoretical and practical aspects of cryptography.

Gelare Oudi Ghadim received her M. Sc. degree in secure communication and cryptography from Shahid Beheshti University, Tehran, Iran, in 2018 and B.S. degree in communication engineering from Sistan and Baluchestan University, Zahedan, Iran, in 2015. Her research interests include RFID security and privacy.

How to cite this paper:
S. Saderi Oskuiee, F. Moazami, G. Oudi Ghadim, "Improving the security of a low cost tag search protocol," J. Electr. Comput. Eng. Innovations, 9(1): 25-36, 2021.

DOI: 10.22061/JECEI.2020.7342.383

URL: http://jecei.sru.ac.ir/article_1481.html