International Journal of u- and e- Service, Science and Technology
Vol.7, No.2 (2014), pp.263-274
http://dx.doi.org/10.14257/ijunesst.2014.7.2.24

A Security Mechanism for Diffusing RFID-Sensor Networks

He Jialiang[1] and Xu Zhiqiang[2]

*[1]*College of Information and Communication Engineering, Dalian Nationalities University, China*
*[2]*Department of digital media technology, Sichuan College of Media and Communications, China*
urchin2012@sina.com; starsep928@yahoo.com.cn

Abstract

Diffusing RFID-Sensor Networks have been used widely in many domains. This paper investigates and generalizes security and privacy requirements for diffusing RFID-Sensor networks; A secure authentication protocol and its corresponding search protocol based on one-way hash function and dynamic ID mechanism are proposed subsequently, this authentication protocol only requires O(1) work to identify and authenticate a tag in the backend server and is particularly suitable for the low-cost RFID systems, the security and privacy characters of these two protocols are analyzed as well.

Keywords: *RFID-Sensor networks; Privacy; Security*

1. Introduction

RFID technology has been widely used in various academia and industry domains [5, 6]. Usually, a typical RFID system consists of a backend-server, readers and tags. A RFID tag has a unique number, such tag can be attached to an object, including product, animal or people. A RFID reader can execute reading and writing operations on each tag over wireless communication, so a specific tag attached target can be recognized and located. However, readers cannot authenticate and search tags very accurately, the information of a tag that is near or out of rated range cannot be read exactly. This distance limitation severely restricts application and deployment of RFID technology. Despite equipping readers and tags with stronger antenna wire can promote communication capability of a RFID system, readers still have difficulties in recognizing tags at a long distance. For solving this problem, a diffusing wireless RFID-sensor network can act as a bridge between the readers and remote targets [3].

Table 1. Classes of RFID Tags [8]

Type	Attributes		
	Class	Functions	Properties
Passive	0	Read Only	No power needed
	1	Read, and Write Once	Sensitive to interference
	2	Read / Write	Short rage
Semi-Passive	3	Increased Range	Can connect to sensor, data logging or cryptography
Active	4	Active Tags Communications	Same as Class3
	5	Reader	Can communicate with other tags

ISSN: 2005-4246 IJUNESST

A wireless sensor network consists of groups of sensor nodes connected by wireless links that perform sensing tasks. These sensors are employed for specialized tasks like environment monitoring, security management, location tracking, *etc*. Sensor nodes can communicate with RFID tags through the wireless interface. Low-cost sensor nodes can be deployed widely, so allowing readers to find targets at a long distance. In a diffusing RFID-sensor networks environment, a certain amount of sensor nodes and tags can form a dynamic, distributed cluster, a certain amount of clusters are connected by coordinators that can communicate with RFID readers [1].

Using sensor nodes can solve the distance limitation problem, but it leads to additional security and privacy challenges. Due to strictly limited resources, small storage capacity and faint power supply of low-cost tags, it is difficult to apply an ordinary and complicated but safe cryptographic algorithm to a RFID system and these factors are hindering the rapid spread of this technology [2]. So designing an efficient and low-cost security scheme for RFID-sensor systems becomes a challenging research object. A well designed diffusing RFID-sensor system should satisfy the requirements of security as follows:

(1)Tag untraceability

If responding message in authentication process or search process from a tag always contains a constant, namely the response is linkable to each other or distinguishable from those of other tags, an adversary can recognize and locate the tag by intercepting and analyzing. That is to say, the location privacy of the user that attached by the tag can be traced [4].

(2)Tag information protection

A tag is always attached to a specific object, storing data in an encrypted form helps retain its confidentiality. Because sensors are vulnerable, computation-limited, and low-cost devices, allowing sensors to decrypt data to perform a search result in unnecessary risk of disclosure. Thus, sensors must execute a secret search directly on ciphertext, rather than plaintext. Furthermore, data transmitted over a wireless interface is susceptible to exposure. Therefore, sensors must only transmit encrypted data. In summary, the data must remain in an encrypted form and should not be decrypted unless necessary to minimize the possibility of disclosure [3]. So through all transmission process of information, an illegimate user should not acquire the holder's detailed information.

(3)Spoofing attack

An adversary may feign a legitimate tag and communicate with a reader instead of the tag and be authenticated as the tag but the genuine legitimate tag may be out simultaneously.

(4)Replay attack

Such an attack in which an adversary repeatedly launches a message that obtained by eavesdropping or intercepting from a regular communication between a reader and a tag during a normal authentication access.

(5)Denial of Service (DoS) attack

According to whether the backend server and a tag update the identifier or not in an entire authentication access, RFID authentication protocols are divided into dynamic ID mechanism and static ID mechanism. For the protocols based on dynamic ID mechanism, in a execution access of corresponding authentication protocol of this RFID system, if an adversary disturbs the communications between a reader and a tag by means of intercepting or blocking messages transmitted, the secret value that shared between the server and the tag cannot be updated successfully in this authentication access, it would cause losing synchronization between the backend server and the tag, if this RFID system search for this specific tag at

that time, the search process maybe fail, so the design of a search protocol should solve this problem[10].

(6)Privacy of search result

The other privacy requirement to be considered is the search result of a mobile reader. It is undesirable to reveal the search result of a mobile reader. In some circumstances to an adversary, it might be useful information whether a mobile reader holder found a particular tag or not [7]. So a well designed search protocol should protect privacy of search result from an illegal user.

We present a mutual authentication protocol that is feasible for diffusing RFID-Sensor networks. Subsequently, we present its corresponding search protocol that enables readers to perform searches over encrypted data. By only using one-way hash function, pseudorandom number generation function and XOR operation, these two protocols are based on dynamic ID mechanism and accommodate the resource limitations of both tags and sensor nodes.

The rest of this paper is organized as follows. Section 2 introduces the related research works; Section 3 introduces architecture for diffusing RFID-sensor networks, while Section 4 presents a mutual authentication protocol for this architecture. The search protocol is porposed to query encrypted data in Section 5, and more properties analysis of security is discussed in Section 6. Finally, Section 7 concludes our work.

2. Related Works

In 2010, Shih-I Huang and Shiuhpyng Shieh proposed authentication and secret search mechanisms for RFID-aware wireless sensor networks [3], these mechanisms may solve the tag searching problem by specific encrypted ciphertexts in RFID-aware wireless sensor networks, but theirs method have some limitations as follows:

(1)These mechanisms are not suitable for the tags attached mobile targets, these mechanisms save the encrypted information of each tag in the tiny database of the sensor nodes near the tag, however, if the tag attached mobile targets move out the ranges that the sensor node can recognize then the system shows the specific tag has been founded by calculation outcome of sensor node nearby but the genuine tag may be out there in fact. So it is suitable for the tags attached fixed targets, then using domain is limited [9].

(2)The information of each tag user stores in tag part, as we know, in three parts of a RFID system, the attackers get user information most easily by cracking tags than by cracking readers or backend server, so a well-designed RFID system, only the necessary data that needed for authentication process and search process stores in tag part for security, the others information storing in backend-server is more secure [9].

(3)Only specific reader can perform search process for the specific tag, because the calculation process of the related encrypted information stores in the sensor nodes nearby the specific tag use a symmetric encryption key used by the specific reader, so the others readers cannot perform search process, so it is inconvenient for using.

(4)According to whether the backend server and a tag update the identifier or not in an authentication access, RFID authentication protocols are divided into dynamic ID mechanism and static ID mechanism. Dynamic ID mechanism is always used in the circumstance that the ownership of ID is needed to transfer. However, this method is based on static ID mechanism, so the application domain is limited.

(5)Low-cost passive tags have constraint requirements of limited resources, using less hardware cost is an important research object. We can see that using pseudo random number generator in tags leads to extra hardware cost. The method assumes that each reader, tag and

sensor node has a synchronized timer, allowing them to verify that an authentication process has not expired. However, it is unpractical to put a timer into a low-cost passive tag [9].

In 2013, the authors proposed authentication and search mechanisms for diffusing RFID-Sensor networks [9]. Therefore, these two mechanisms has two limits as follows:

(1)According to whether the backend server and a tag update the identifier or not in an authentication access, RFID authentication protocols are divided into dynamic ID mechanism and static ID mechanism. Dynamic ID mechanism is always used in the circumstance that the ownership of ID is needed to transfer. However, the method in [9] is based on static ID mechanism, so the using application domain is limited.

(2)The method assumes that each reader, tag and sensor node has a synchronized timer, allowing them to verify that an authentication process has not expired. However, it is unpractical to put a timer into a low-cost passive tag.

Based on the above analysis, we present our solution mechanism as follows.

3. System Architecture

In this section, we introduce an architecture, participating roles and their set-ups when deploying such a network. The notations we used in this paper as follows:

Table 2. The Notations Used

Symbol	Meaning
ID	The tag's unique serial number (The length is *l*)
IDS	The tags' unique index-pseudonym
IDTrail	The last access trail of each ID
Data	Information of the corresponding tag
RID	The unique identifier of a reader (The length is *l*)
H()	An hash function, H: $\{0,1\}^{l^*} \rightarrow \{0,1\}^{l}$ (The length is *l*)
PRNG()	The pseudo random number generator (The length is *l*)
$E_K()$	A symmetric encryption function used between server and reader
$\oplus$	XOR operator
$\|\|$	Concatenation operator
M_L	The left part of the message M
M_R	The right part of the message M
R	The random number generated by the reader
Pre-x	The previous value of x
Cur-x	The current value of x
x_i	The x value in the (i)th session of the protocol
T	Temporary value
A→B:M	A sends message M to B

For solving the distance limitation problem from RFID readers to tags, the authors proposed this architecture of diffusing RFID-sensor networks [9]. Four roles are involved in the system: Backend server, RFID reader, RFID tag, and RFID-aware Sensor node. The backend server has a central database storing each ID of an authorized tag and each RID of an authorized reader.

Sensor nodes are used as the interface between readers and tags in this architecture, allowing readers to trace a specific tag located far away, which can compensate for the distance limitation of readers. The sensor network allows for multi-hop communication. Furthermore, readers, tags and sensor nodes can maintain secure communications. Figure 1 depicts system architecture.

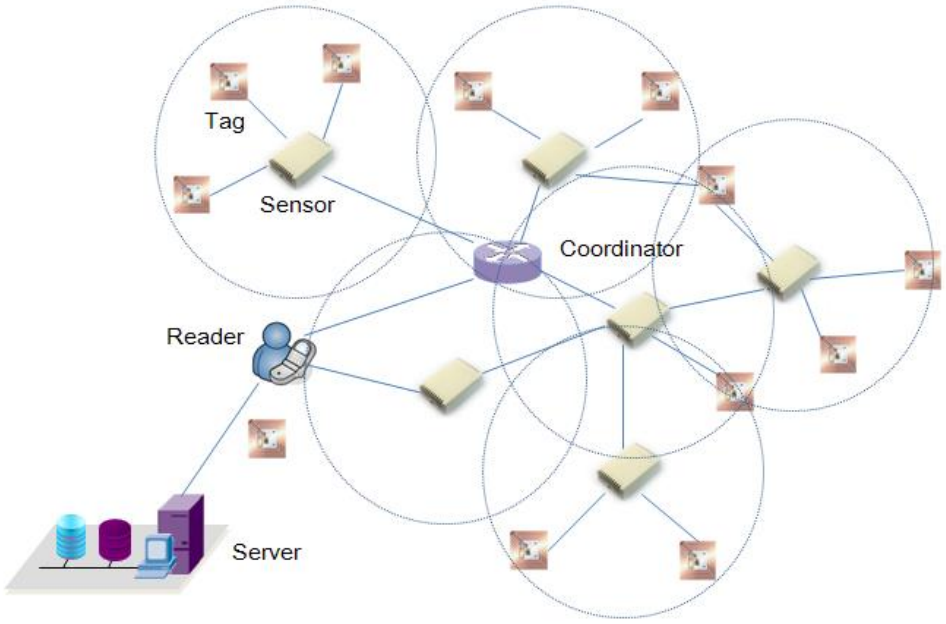

Figure 1. System Architecture [9]

This architecture is workable for passive tags, needs only wireless signals to charge and becomes active. No extra power waste will be needed, and thus each sensor node can reduce unnecessary power consumption in reading or writing tags. The channel between the backend-server and a reader is assumed insecure for wireless connection, and the channel between a reader and a tag is assumed insecure either, we assume that an adversary can observe and manipulate communications between insecure channels.

4. Authentication Mechanism

In this section, we are going to introduce a lightweight authentication mechanism based on one-way hash function between readers and tags. Since all query actions are initiated by readers, the sensor nodes are merely viewed as generic routers and used only to forward these queries to tags. In setting phase, we need to set up two components: the tag and the backend-server. For each tag, it is assigned with a unique ID, and a temporary value S. All fields of (ID, IDTrail, Data) store in the backend server. These settings are performed in factor before deploying them into real work. Since the passive tag has limited computation capabilities, assuming that in our mechanism the tag can afford lightweight operations including XOR and one-way hash function.

The first step in building a relationship between readers and tags is authentication. Since readers and tags rely on wireless communication, attackers may eavesdrop or obstruct transmitted data. Previous research characterizes RFID communication as asymmetrical in signal strength. That is, attackers have an easier time listening in on signals from reader to tag than on data from tag to reader. Additionally, attackers can easily purchase readers and tags to perform malevolent operations. Therefore, we propose this protocol for readers and tags to prevent attackers from impersonating authorized entities. According to the outline in Figure 2, the authentication protocol involves the following steps:

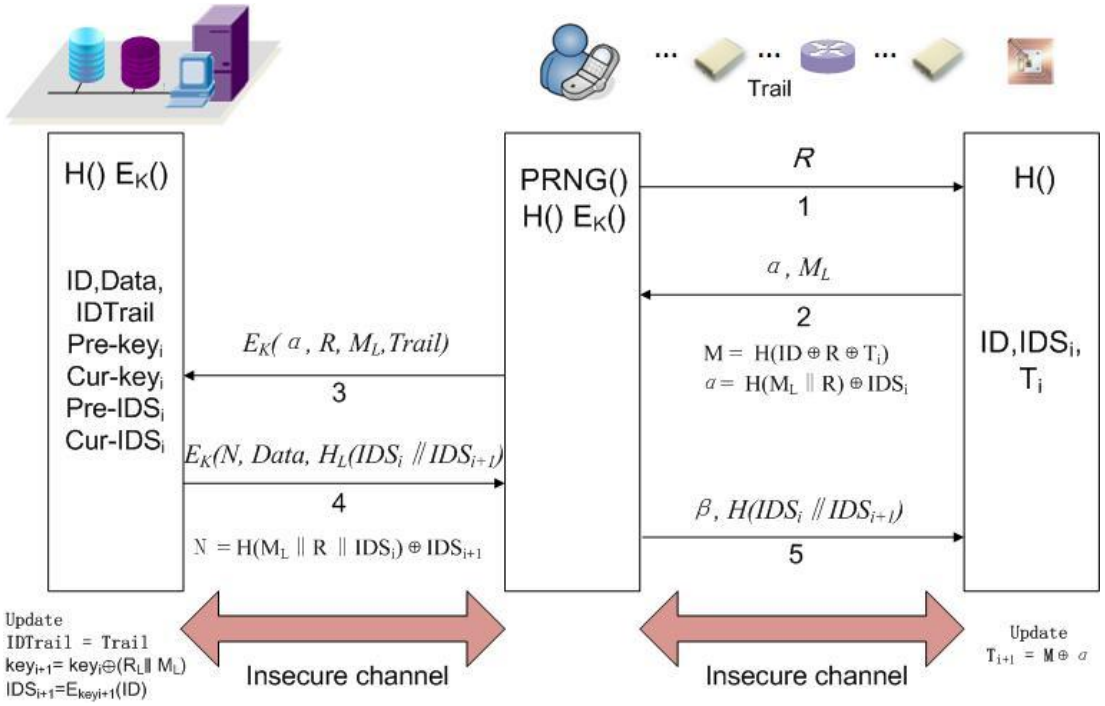

Figure 2. Authentication Mechanism

Step1 To begin the authentication process, the reader generates a random number R, and then sends a request with R to the tag.

Step2 When the tag receives the request with R, firstly, it calculates $M = H(ID \oplus R \oplus T_i)$ and $\alpha = H(M_L \| R) \oplus IDS_i$, then send α and M_L back to the reader. Subsequently the tag should calculate $S_{i+1} = M \oplus \alpha$ and save R, S_{i+1} in its memory, it is quite reasonable that tags have enough memory space to store these two parameters. Especially, we use $H(ID \oplus R \oplus T_i)$ to substitute pseudo random number of the tag.

Step3 After receiving α and M_L, the reader calculates $E_K(\alpha, R, M_L, Trail)$, and sends $E_K(\alpha, R, M_L, Trail)$ to the backend server (Trail indicates the route from the tag to the reader). Since readers have stronger capacity of calculation and communication, so we choose traditional symmetric encryption function between the server and reader.

Step4 When the server receives $E_K(\alpha, R, M_L, Trail)$ from the reader, firstly, it decrypts $E_K(\alpha, R, M_L, Trail)$ and gets α, R, M_L, Trail, then the server would calculate $IDS = H(M_L \| R) \oplus \alpha$ with decrypted α, R, M_L, and search whether there exists certain IDS* in column 'Cur-IDS' of the database, which can make IDS = IDS*. If there exists such record, the tag would be considered as a legitimate tag, then the backend server should calculate key_{i+1} = Cur-key* $\oplus (R_L \| M_L)$ and $IDS_{i+1} = E_{keyi+1}(ID^*)$. In particular, after accomplishing these two calculations, the backend server must search whether there exists IDS' in column 'Pre-IDS' and column 'Cur-IDS', which can make IDS' = $E_{keyi+1}(ID^*)$. If there not exists such record, the backend server would calculate $H_L(IDS_{i+1} \| IDS_i)$, $N = H(M_L \| R \| IDS_i) \oplus IDS_{i+1}$, and $E_K(N, Data, H_L(IDS_{i+1} \| IDS_i))$ then send $E_K(N, Data, H_L(IDS_{i+1} \| IDS_i))$ to the reader and update IDTrail = Trail, Pre-key = Cur-key, Cur-key = key_{i+1}, Pre-IDS = Cur-IDS, Cur-IDS = IDS_{i+1}; if there exists such record, the backend server must generate such a random key' that can make the value which equals to $E_{key'}(ID^*)$ cannot be found in column 'Pre-IDS' and column 'Cur-IDS', then the backend server would calculate $H_L(IDS_{i+1} \| IDS_i)$, $N = H(M_L \|$

$R \| IDS_i) \oplus IDS_{i+1}$, and $E_K(N, Data, H_L(IDS_{i+1} \| IDS_i))$ then send $E_K(N, Data, H_L(IDS_{i+1} \| IDS_i))$ to the reader and update IDTrail = Trail, Pre-key = Cur-key, Cur-key = key', Pre-IDS = Cur-IDS, Cur-IDS = $IDS_{i+1} = E_{key'}(ID^*)$.

If there not exists certain IDS* in column 'Cur-IDS' of the database, which can make IDS = IDS*. The backend server would search whether there exists certain IDS* in column 'Pre-IDS' of the database, which can make IDS = IDS*. If there exists such record, the tag would be considered as a legitimate tag, but in the last authentication access, the tag has not updated key and IDS successfully for some reason, so the backend server calculates key_{i+1} = Pre-key* $\oplus (R_L \| M_L)$ and $IDS_{i+1} = E_{keyi+1}(ID^*)$. In particular, after accomplishing these two calculations, the backend server must search whether there exists IDS' in column 'Pre-IDS' and column 'Cur-IDS', which can make IDS' = $E_{keyi+1}(ID^*)$. If there not exists such record, the backend server would calculate $H_L(IDS_{i+1} \| IDS_i)$, $N = H(M_L \| R \| IDS_i) \oplus IDS_{i+1}$, and $E_K(N, Data, H_L(IDS_{i+1} \| IDS_i))$ then send $E_K(N, Data, H_L(IDS_{i+1} \| IDS_i))$ to the reader and update IDTrail = Trail, Cur-key = key_{i+1}, Cur-IDS = $E_{keyi+1}(ID^*)$ but Pre-key and Pre-IDS would keep unaltered; if there exists such record, the backend server must generate such a random key' that would make the value which equals to $E_{key'}(ID^*)$ cannot be found in column 'Pre-IDS' and column 'Cur-IDS', then the backend server would calculate $H_L(IDS_{i+1} \| IDS_i)$, $N = H(M_L \| R \| IDS_i) \oplus IDS_{i+1}$, and $E_K(N, Data, H_L(IDS_{i+1} \| IDS_i))$ then send $E_K(N, Data, H_L(IDS_{i+1} \| IDS_i))$ to the reader, then update IDTrail = Trail, Cur-key = key_{i+1} = key', Cur-IDS = $E_{key'}(ID^*)$ but Pre-key and Pre-IDS would keep unaltered.

If there not exists certain IDS* in column 'Cur-IDS' and column 'Pre-IDS' of the database, which can make IDS = IDS*, the authentication is failed, F(failure information) would be sent to the reader.

Praiseworthily, in this phase, only one hash operation would be needed in verifying and authenticating a tag, so time complexity of hash function calculation achieves O(1).

Step5 After receiving $E_K(N, Data, H_L(IDS_{i+1} \| IDS_i))$ from the backend server, the reader would get N, Data, $H_L(IDS_{i+1} \| IDS_i)$ by decrypting $E_K(N, Data, H_L(IDS_{i+1} \| IDS_i))$, so acquire the detailed information about the tag attached the target. Subsequently, the reader sends N, $H_L(IDS_{i+1} \| IDS_i)$ to the tag.

Step6 When the tag receives N, $H_L(IDS_{i+1} \| IDS_i)$ from the reader, the tag would calculate IDS" = $(M_L \| R \| IDS_i) \oplus N$, then calculate $H_L(IDS" \| IDS)$. If $H_L(IDS" \| IDS)$ equals to received $H_L(IDS_{i+1} \| IDS_i)$, then the object of mutual authentication achieves, the tag should update IDS_{i+1} = IDS", and inform the sensor nodes nearby it that it has been authenticated successfully already; otherwise, the authentication is failed.

This protocol can reduce the re-authentication cost when a reader wishes to send request to a tag that has been authenticated successfully. The reader need not re-authenticate the tag again because the nearby sensor nodes storing the tag's information. If a new reader queries the record (Time, ID) that stores in the sensor nodes nearby the tag, then it would judges that another reader has authenticated the tag already.

5. Search Mechanism

To preserve data privacy, simply encrypting data prevents attackers from discerning the contents. However, traditional cryptography is not feasible in tags and sensor nodes because of their limited computation and calculation capability. Moreover, it is difficult to search encrypted data. To solve this problem, we propose a corresponding search mechanism, which maintains data in an encrypted form but allows authorized readers to perform search process without disclosing data during transmissions process.

This search mechanism involves two conditions: tag search by Trail (The outline in Figure 3) or tag search in all networks (The outline in Figure 4). Tag search by Trail is that the reader use the Trail data storing in the database in the backend server to trace the location of the specific tag, however, if the tag attached the target can moveable and has move out of the range of sensor nodes nearby it, the Trail data is no useful, so the system must perform search process in all network on ciphertexts.

5.1. Tag Search by Trail

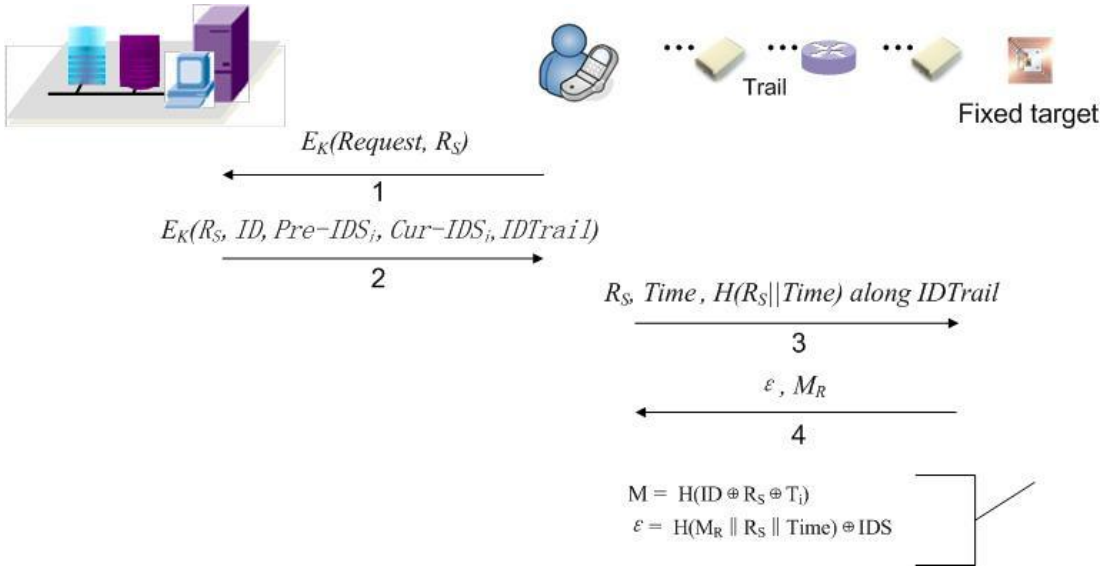

Figure 3. Tag Search by Trail

Step1 To begin the search process, the reader generates a random number R_S, and then calculates $E_K(Request, R_S)$. Subsequently the reader sends $E_K(Request, R_S)$ to the backend server.

Step2 After receiving $E_K(Request, R_S)$, the server gets R_S by decrypting $E_K(Request, R_S)$, then calculates $E_K(R_S, ID, Pre\text{-}IDS_i, Cur\text{-}IDS_i, IDTrail)$ and sends it back to the reader.

Step3 When the reader receives $E_K(R_S, ID, Pre\text{-}IDS_i, Cur\text{-}IDS_i, IDTrail)$ from the server, it gets ID, $Pre\text{-}IDS_i$, $Cur\text{-}IDS_i$, IDTrail by decrypting $E_K(R_S, ID, Pre\text{-}IDS_i, Cur\text{-}IDS_i, IDTrail)$, then checks the current time Time and calculates $H(R_S \| Time)$, subsequently send R_S, Time, $H(R_S \| Time)$ along with IDTrail towards the tag. When this information packet gets to the nearest sensor node nearby the tag, the sensor broadcasts it to the tags around it.

Step4 Each tag near the sensor node calculates $H(R_S \| Time)$ using received R_S, Time to verify this request, it calculates $M = H(ID \oplus R_S \oplus T_i)$ and $\varepsilon = H(M_R \| R_S \| Time) \oplus IDS$, then sends ε and M_R back to the reader along Trail. Subsequently the tag should calculate $T_{i+1} = M \oplus \varepsilon$ and save T_{i+1} in its memory. Especially, we use $H(ID \oplus R_S \oplus T_i)$ to substitute pseudo random number of the tag.

Step5 After receiving some response information from the tags from terminal sensor node, firstly, the reader calculates $H(M_R \| R_S \| Time) \oplus \varepsilon$ with received M_R, if one of the calculation outcomes equals to the $Cur\text{-}IDS_i$ received in step3, or one of the calculation outcomes equals to the $Pre\text{-}IDS_i$, then the specific tag is found, however, if there is no such outcome, meaning that the tag attached the target maybe move out of the range of the

terminal sensor node in the Trail nearby it, the Trail data is no useful, so the system must perform search process in all network on ciphertexts.

5.2. Tag Search in all Network

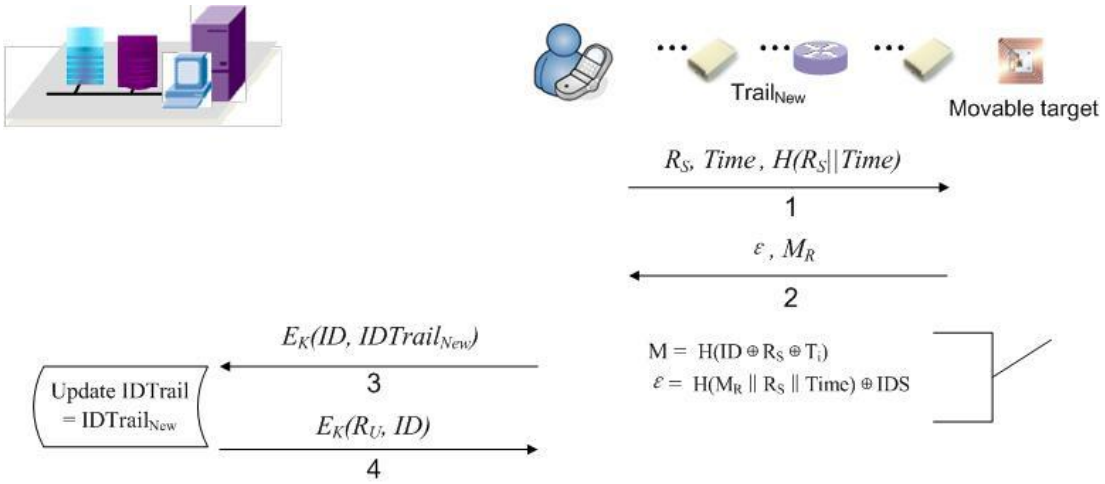

Figure 4. Tag Search in all Network

Step1 The reader generates a random number R_S and sends R_S, Time, $H(R_S \| Time)$ towards sensor nodes. When this information packet gets to any sensor node, the sensor broadcasts it to the tags around it.

Step2 Each tag near the sensor node calculates $H(R_S \| Time)$ using received R_S, Time to verify this request, it calculates $M = H(ID \oplus R_S \oplus T_i)$ and $\varepsilon = H(M_R \| R_S \| Time) \oplus IDS$, then sends ε and M_R back to the reader along Trail. Subsequently the tag should calculate $T_{i+1} = M \oplus \varepsilon$ and save T_{i+1} in its memory.

Step3 After receiving some response information from the tags from terminal sensor node, firstly, the reader calculates $H(M_R \| R_S \| Time) \oplus \varepsilon$ with received M_R, if one of the calculation outcomes equals to the Cur-IDS_i received in step3, or one of the calculation outcomes equals to the Pre-IDS_i, then the specific tag is found, then the reader calculates $E_K(ID, IDTrail_{New})$ and sends it to the backend server. However, if there is no such outcome, meaning that the tag attached the target maybe move out of the range of all the terminal sensor nodes.

Step4 After receiving $E_K(ID, IDTrail_{New})$, the server gets ID, $IDTrail_{New}$ by decrypting (ID, $IDTrail_{New}$) and updates IDTrail with $IDTrail_{New}$, then generates a random number R_U, calculates $E_K(R_U, ID)$ and sends it back to the reader.

Step5 When the reader receives $E_K(R_U, ID)$ from the server, gets R_U, ID by decrypting $E_K(R_U, ID)$, then the search process in all network is finished.

6. Analysis

6.1. Security of Authentication Mechanism

We would analyze this protocol to evaluate whether it meets the security requirements as follows:

(1)Tag untraceability

An adversary can eavesdrop the response message ($H(M_L \| R) \oplus IDS_i$, M_L) from a tag, and analyze the information carefully and try to detect the user's location privacy by tracking the tag. Because the tag generates a new substituted random number $M = H(ID \oplus R \oplus T_i)$ during each authentication access, and updates $S_{i+1} = M \oplus (H(M_L \| R) \oplus IDS_i)$ in the step2, so the adversary cannot differentiate which tag does the response from the message ($H(M_L \| R) \oplus IDS_i$, M_L). So this protocol can meet tag untraceability.

(2)Tag information protection

Because the detailed information of an ID (Data) is only stored in the backend server, for the security reason, the tag part only stores the prerequisite data for authentication process and search process, Data only is transmitted from the channel from the backend server to a reader that has used traditional symmetric encryption algorithm for a reader has relative stronger capacity of calculation and communication. So an adversary cannot acquire the detailed information of the ID. So this protocol can meet tag information protection.

(3)Spoofing attack

An adversary feigns a legitimate reader which sends a query with R to tags through the forward channel, and obtains the response of a tag ($H(M_L \| R) \oplus IDS_i$, M_L). In the next authentication access, when a legitimate reader sends a query with R', the adversary feigns the tag and responds the legitimate reader with the obtained message ($H(M_L \| R) \oplus IDS_i$, M_L) through the backward channel. However, the reader generates a new random number during each authentication access, that is to say, $R \neq R'$, so the adversary cannot perform tag impersonation attack.

(4)Replay attack

Replay attack can be prevented in this protocol due to the message transmitted for each access is different. Different value of $H(M_L \| R)$ is utilized in individual access and T that stored in a tag plays a key role in providing different value of $H(M_L \| R)$ to conceal ID of the tag. An adversary cannot acquire H() so as to calculate $H(M_L \| R)$, and we use timestamp to control the sequence character of this system. So it is impossible for the adversary to perform replay attack.

(5)Denial of Service (DoS) attack

As pseudonym IDS of a tag is mutative, even if loss of message, power failure or loss of connection with the backend server happens during an authentication access, it will lead to dy-synchronization between the backend server and the tag, this protocol can solve this problem in the next authentication access by searching pseudonym IDS in column 'Pre-IDS' and continuing the verification process. So this protocol can shield DoS attack well.

6.2. Security of Search Mechanism

We would analyze this protocol to evaluate whether it meets the security requirements as follows:

(1)Tag untraceability

An adversary can eavesdrop the response message ($H(M_R \| R_S \| Time) \oplus IDS$, M_R) from a tag, and analyze the information carefully and try to detect the user's location privacy by tracking the tag. Because the tag generates a new substituted random number $M = H(ID \oplus R_S \oplus T_i)$ during each search access, and updates $S_{i+1} = M \oplus (H(M_R \| R_S \| Time) \oplus IDS)$ in the step4, so the adversary cannot differentiate which tag does the response from the message ($H(M_R \| R_S \| Time) \oplus IDS$, M_R). So this protocol can meet tag untraceability.

(2)Tag information protection

Because the detailed information of an ID (Data) is stored in the backend server and is transmitted through the secure encrypted channel from the backend server to the reader, an adversary cannot acquire the detailed information of the ID. So this protocol can meet tag information protection.

(3)Privacy of search result

This protocol can protect the search result of a reader. Because all tags nearby the sensor node respond to the request, an adversary cannot learn whether the reader found a specific tag or not. Even if the specific tag itself cannot know whether the reader wants to find it or not, since each tag would calculate $H(M_R \| R_S \| \text{Time}) \oplus \text{IDS}$ of its own and send the message $(H(M_R \| R_S \| \text{Time}) \oplus \text{IDS}, M_R)$ to the reader.

(4)This protocol is based on dynamic ID mechanism, for solving the problem of Denial of Service attack in a execution access of corresponding authentication protocol of this RFID system, the shared value Cur-IDS_i and Pre-IDS_i between the backend server and the tag should be considered, so $E_K(R_S, \text{ID}, \text{Pre-IDS}_i, \text{Cur-IDS}_i, \text{IDTrail})$ has been calculated, the reader can match Pre-IDS_i, Cur-IDS_i in a search operaton process. Even if the backend server and the tag have lost synchronization for some reason in previous authentication access, the tag can be searched successfully.

7. Discussion

In this paper, we propose an authentication protocol which mutually authenticates readers and tags in diffusing RFID-Sensor networks. This protocol can resist common attacks, and also provide a search query trail among the back-end server. This property brings a higher level of efficiency to tag search process. To perform queries on encrypted data, a search protocol is proposed, which searches secrets in an encrypted form without the need to decrypt it, this protocol is feasible for sensor nodes and RFID tags, as it uses low-computation operations, *i.e*., one-way hash function is needed to encrypt and search data, and can prevent the disclosure of information.

Acknowledgement

This work was partially supported by Heilongjiang Province Science and Technology Research Grant of the Education Department No.12533002.

References

[1] J. Cho, Y. Shim, T. Kwon, Y. Choi, S. Pack and S. Kim, "SARIF: A Novel Framework for Integrating Wireless Sensor and RFID Networks", IEEE Transactions on Wireless Communications, vol. 14, **(2007)**, pp. 50-56.
[2] A. Juels, "RFID Security and Privacy: A Research Survey", IEEE Journal on Selected Areas in Communications, vol. 24, **(2006)**, pp. 381-394.
[3] S.-I. Huang and S. Shieh, "Authentication and Secret Search Mechanisms for RFID-aware Wireless Sensor Networks", International Journal of Security and Networks, vol. 5, vol. 1, **(2010)**, pp. 15-25.
[4] H. Jialiang, O. Dantong and Y. Yuxin, "An Efficient Lightweight RFID Authentication Protocol for Low-cost Tags", Advances in Information Sciences and Service Sciences, vol. 3, no. 9, **(2011)**, pp. 331-338.
[5] G. Avoine, "Cryptography in Radio Frequency Identification and Fair Exchange Protocols", Ph.D. Thesis, Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland, **(2005)**.
[6] X. Zhu, S. K. Mukhopadhyay and H. Kurata, "A Review of RFID Technology and Its Managerial Applications in Different Industries", Journal of Engineering and Technology Management, vol. 29, **(2012)**, pp. 152-167.

[7] J. Young Chun, J. Yeon Hwang and D. Hoon Lee, "RFID tag search protocol preserving privacy of mobile reader holders", IEICE Electronics Express, vol. 8, no. 2, **(2011)**, pp. 50-56.
[8] L. Yan, Y. Zhang, L. T. Yang and H. Ning, "The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems", Wireless Networks and Mobile Communications, Auerbach Publicatios, **(2008)**.
[9] H. Jialiang and X. Zhiqiang, "Authentication and Search Mechanism for Diffusing RFID-Sensor Networks", International Journal of Sensor Networks, vol. 14, no. 4, **(2013)**, pp. 211-217.
[10] H. Jialiang, X. youjun and X. Zhiqiang, "A Hash-based RFID Search Protocol for Mobile Reader", International Journal of Hybrid Information Technology, vol. 7, no. 2, **(2014)**.

Authors

He Jialiang, was born in 1977, received the PhD degree in computer software and theory from Jilin University of China in 2012 and the Master degree in computer application from Jilin University of China in 2004. Now he is an associate professor at College of Information and Communication Engineering, Dalian Nationalities University, China. His papers have been published in some well-known international Journals and IEEE conferences. His main interests include Mobile Internet, Internet of Things, and Intelligent Business Information Processing.

Xu Zhiqiang, borned in 1981, received the Bachelor degree in communication Engineering from Communication University of China in 2004 and the Master degree in Electronics & Communication Engineering from Communication University of China in 2012. At present, he is an assistant professor of Communication & Media Institute of Sichuan, China. He is experienced the fields of Mobile Internet, Internet of Things, Intelligent Information Processing, etc., he also is a candidate of MSc of Technopreneurship & Innovation Program in Nanyang Technological University in Singapore.

