Cloud Database Service Oriented RFID Security Authentication Protocol Design

Dan Li[1]

1 Shaanxi Xueqian Normal University,
Shaanxi Xi'an, 710100 China

***Abstract* — A safe, effective and expandable RFID authentication protocol with cloud database as server (CRFID) is proposed in order to ensure the security of RFID network. Firstly, the tree structure management tag is used to realize privacy protection and meanwhile the time complexity of the cloud database searched by RFID system is reduced from O(N) to O(logN); then, the size of each sub-item in the secret key path of RFID tag is increased from 4 bits to 60 bits in order to prevent pursuit attack; finally, according to tag feedback information, reader judges whether to update the secret key stored in tag so as to prevent desynchronized attack. The simulation result shows that the scheme can not only reduce search complexity, but also significantly improve the security level of secret key path.**

Keywords* - *RFID; cloud database; time complexity; desynchronized attack; pursuit attack

I. INTRODUCTION

Extensively concerned in recent years, RFID (Radio Frequency Identification) system has the advantages of cloud data management, parallel processing, short access time, long-distance non-contact sensing, rewriting, etc. Specifically, RFID system is composed of three parts: RFID tag (transmitter), back-end server used for storing tag associated information and RFID reader (receiver), and wireless connection mode is used for tag scanning and database query [1-2]. Therein, the back-end database server allows all RFID readers to access and meanwhile has strong parallel computing ability in order to meet the requirements of all readers for concurrent query. The back-end server, also called cloud database, is used to compute and store information as well as store Hash value or ID code and secret key checklist, and one of the important functions of back-end server is secret key management [3]. The information transmission from tag to reader and from reader to server is transparent, so security risk may exist during information transmission process if the proper measure is not provided to protect the information stored in tag. In literature, in order to protect tag privacy, a privacy protection authentication (PPA) is proposed for directly maintaining the secret keys of all tags in cloud database. Since cloud database must store a lot of secret keys, thus PPA method cannot be extended. In literature, a simple solution for preventing pursuit attack is proposed, but the search time complexity of the back-end database thereof is O(N) and the efficiency in large RFID system is very low [4-5], wherein N is the number of tags. In literatures, several kinds of balance tree based structures are proposed to solve the privacy problems in RFID system; although the balance tree mechanism can reduce the search time complexity from O(N) to O(logN), some tags may still have partially same secret keys; therefore, once any attacker captures several tags, information may be leaked.

In literature, a balance tree based ACTION protocol is proposed, the search time complexity thereof is O(logN) and such protocol still can protect tag privacy under the condition that most tags are captured; however, ACTION protocol can easily receive pursuit attack and desynchronized attack, and the attacker may track the tag without the need to capture any tag [6-7].

II. PROPOSING OF CRFID PROTOCOL

An expandable, safe and efficient RFID authentication protocol, namely CRFID, is designed in this paper in order to conquer the security weakness of existing methods. The symbols used in this paper are as shown in Table 1.

TABLE I SYMBOLS USED IN THIS PAPER

Symbol	Meaning
R	RFID reader
Tag	RFID tag
n_i	Characteristic random number of i
l_n	Length of n_i
$H()$	One-way Hash function
l_h	Length of $H()$ output value
k_i	Secret key path of T
$k_i[x]$	x-th sub-item of secret key path
l_k	Length of one sub-item of secret key path
s_i	Secret key of tag
δ	Branch factor of secret key tree
T	Secret key tree in database
d	Depth of secret key tree
c	TagJoin running times
N	Total number of tags
V	Victim list

The scheme includes two stages: initialization and reading. In the initialization stage, unique secret key pair k_i and s_i is allocated to each tag $T_{i,;}$ at the same time, reader establishes pseudo secret key tree. Please notice that the tag itself does not include any output information to indicate authentication process success or failure. Therefore, any reader cannot determine whether the protocol is successfully executed or not [8]. In order to solve critical defect, a communication process is designed in step 9 in the algorithm proposed in this paper. CRFID reading protocol is as follows:

```
1.$R \rightarrow T$ : Request , $n_1 ; n_1 \xleftarrow[R]{} \{1\}^n$
2.$T$       : $U = \{n_2, H(n_1 \| n_2 \| k_i[0]), H(n_1 \| n_2 \| k_i[1]),$
              $..., H(n_1 \| n_2 \| k_i[d-1]), H(n_1 \| n_2 \| s_i)\} ; n_2 \xleftarrow[R]{} \{1\}^n$
3.$T \rightarrow R : U$
4.$R$       : $\{i, k_i, s_i\} \leftarrow Identify(U)$
5.$R$       : Add $\{k_i, s_i\}$ to Junk list $L$.
            : $\{k_i', s_i'\} \leftarrow KeyGen(k_i, s_i, i)$
6.$R \rightarrow T$ : $\sigma = \{H(n_1 \| n_2 \| k_i) \oplus k_i', H(n_1 \| n_2 \| s_i \| k_i)\}$
7.$T$       : $\{k_i', s_i'\} \leftarrow NewVerify(\sigma , n_1, n_2)$
            : If verification fails
            : Update $k_i \leftarrow k_i', s_i \leftarrow s_i'$
8.$T \rightarrow R$ : $\alpha = H(n_1 \| n_2 \| k_i' \| k_i)$
9.$R$       : verification $\alpha$ ?=$H(n_1 \| n_2 \| k_i' \| k_i)$
            : If verification fails:
                Re check the label.
            : $or$ :
                Update $k_i \leftarrow k_i', s_i \leftarrow s_i'$
                $Record(k_i'', s_i'')$ to $L$,
                  $TagLeave(k_i'', s_i'')$
              clear $L$
```

Additionally, if the reader does not receive the confirmation message from the tag, then the reader will read the tag again (step 9). Besides initialization and reading processes, the addition of new tags and the verification of updated secret key (keygen and NewVerify, namely algorithm 1 and algorithm 2) are also considered in this paper [9].

III. PROTOCOL SECURITY AND PRIVACY

A. *Security*

CFRID protocol will update critical path and secret key after each successful reading; the quantity complexity of secret keys in tags is *O(1)*; the reader needs to match Hash value in U in order to identify leaf node, each Hash value needs at most δ times of Hash calculations, each leaf node contains a few of tags, and the overall search complexity is $O(\delta \times d)^2$. Therefore, CRFID has three properties: restorability of captured tag, fixed secret key size and high search efficiency. The processes for CRFID to resist the two major RFID attacks are as follows:

(1) Resistance to pursuit attack: CRFID increases the size of each sub-item of k_i from 4 bits to 60 bits and meanwhile keeps the same δ, namely same search complexity, thus to largely avoid pursuit attack and make attackers unable to extract the value of each sub-item from σ. Fig. 1 shows the relation between l_k and search times. According to the figure, if l_k is increased to 30, the value still can be forcedly reversed when tag tree N is 0.1, 1 or 10000000.

(2) Resistance to desynchronized attack: step 8 in CRFID reading protocol is used to resist desynchronized attack. When receiving the message generated in step 8, R can judge whether the secret key stored in T has been updated or not. If message is not sent to R, then T will not update secret key and terminate the protocol, or T will update the secret key as (k_i' , s_i') (R generates corresponding secret key). In some circumstances, R will read T again till receiving the message generated in step 8. In other words, the previous generated secret keys are invalid, thus to avoid pseudo tags (inexistence in records).

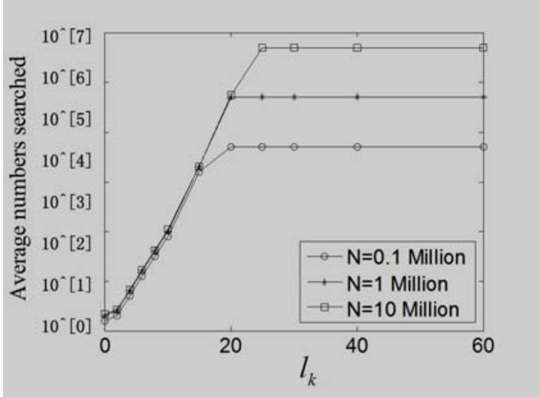

Figure 1. Relation between $k_i[]$ & l_k length and search complexity

B. *Privacy*

After tag is illegally read, attacker can obtain a group of tags ϑ able to generate corresponding messages. Privacy function P(c) defined as $1/|\vartheta|$ is used to denote the privacies of various schemes, wherein c is the number of the captured tags. The ideal privacy protection scheme can generate the function $P(c) = 1/(N-c)$, $\forall c \leq N-1$, wherein N is the total number of all tags in the system [10].

At the time of legal reading (passive monitoring) and illegal reading (active scanning) [11], message $U = (n_2, U_0, U_1, ..., U_{d-1}, U_s)$ is only used to provide information for tag authentication. If attacker does not capture any tag, then it is difficult to confirm whether the data streams for two communications are from the same tag, except that the attacker can guess the secret key path k_i or secret key, or same n_1 and n_2 are used for two times of failed reading (such possibility is very small). Therefore, P (0) can be regarded as $P(0) \approx 1/N$ [12].

The more the captured tags, the more the leaked path keyword information; *C* is the set of the captured tags, and *K* is the set for obtaining path keyword sub-times from *C*. Additionally, the secret key pair of tag T_i is k_i and s_i , the output message after illegal reading is received is $U_0, U_1, ..., U_{d-1}$, and $E_{x,y}$ is defined as follows: when $k_i[0], k_i[1], ..., k_i[x-1] \in K$ but $k_i[x], k_i[x+2], ..., k_i[d-1] \notin K$ [17], y nodes below node $k_i[x]$ are captured.

As for event $E_{x,y}$, the remaining *N-C-1*in set ϑ is averaged according to the following calculation formula:

$$\bar{N}_{x,y}=1+(\delta-y)(N-c-1)/\delta^x \tag{1}$$

In order to calculate the possibility of $E_{x,y}$, it is necessary to carry out the following analysis. *c* balls are used to fully fill δ^x boxes and *C* is assumed as one group including δ boxes. The possibility of y empty boxes in *C* is calculated, for example, one box in *C* is empty. β is defined as the number of full boxes. Pr ($E_{x,y}$) can be regarded as a function $f(c, y, 0)$ as shown in the following calculation formula:

$$f(c,y,\beta)=\frac{\delta-\beta-1}{\delta^x}f(c-1,y-1,\beta+1) + \frac{\delta^x-\delta+\beta}{\delta^x}f(c-1,y,\beta) \tag{2}$$

$$f(1,1,\beta)=\frac{\delta-\beta-1}{\delta^x} \tag{3}$$

$$f(c,0,\beta)=\left(\frac{\delta^x-\delta+\beta}{\delta^x}\right)^c \tag{4}$$

$$f(u,v,\beta)=0, \forall u < v \tag{5}$$

Formula (5) shows that there are c balls, β boxes in C are fully filled and y boxes in C needs to be filled, and two methods can be used to meet the above requirements: the first circumstance: if one ball is put in one box in C, thenβ+1 boxes are fully filled by c-1 balls and y-1 boxes need to be filled; the second circumstance: if one ball is not put in an empty box, then there are c-1 balls and y empty boxes needing to be filled. Formula (4) shows that only one ball can be put into any box rather than the prohibited box, so there are δ-β-1 boxes. Formula (10) shows that there are c balls but there is no empty box needing to be filled, and all balls can be only put in the boxes not belonging to C or in filled boxes, so there are $\delta^x - \delta + \beta$ boxes. Formula (5) shows that there are insufficient balls to be put in remaining boxes, and the possibility is 0.

The privacy function can be calculated according formulae (3) ~ (5):

$$P(c)=\frac{1}{\sum_{x=1}^{d-1}\sum_{y=1}^{\delta-1}\Pr(E_{x,y})\bar{N}_{x,y}} \tag{6}$$

The privacy function curve is as shown in Fig. 2. Please notice that active scanning is only available for pursuit attack before tag is read again, and once the tag is read again, the secret key stored in the tag will be updated, the pursuit attack information obtained by the attacker will be useless.

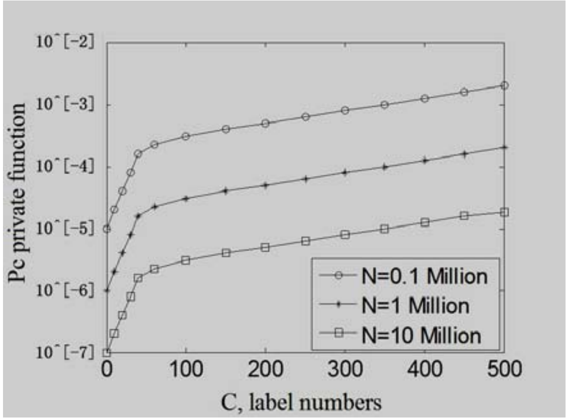

Figure 2. Privacy level curve

IV. Discussion

A. *Parameter Setting*

This section mainly aims at discussing how to choose system parameters in order to make CRFID system have such properties as high expansibility, rapid search, short authentication time and high security level. Specifically, four parameters will be discussed in detail: δ, d, N and l_k . The total number N of tags depends on application program, δ and d shall be set as the function of N, and l_k shall be an independent parameter. Minimum calculation result and the waiting time of R and T shall be considered in the design thought, namely: when T is calculated, R needs to wait; similarly, when R is calculated, T needs to wait. In order to avoid too many tags sharing the verification code of the same path (namely reducing the legal reading search time and avoid path damage when tag is destroyed), δ and d shall be directly proportional to N, wherein δ is the branch factor of the secret tree. When tag is read, smaller δ value can accelerate search but still can cause larger d value. Therefore, several times of Hash calculations must be executed for T and R, but the total time of authentication process will be increased. Oppositely, larger δ value can reduce the tag calculation quantity but meanwhile can increase the search time of T and R verifications.

Additionally, the length of the secret key stored in T is d× l_k which still depends on d.

According to the scheme, R will execute XOR operation and maximally d×δ+ 5 times of Hash operations; T will execute XOR operation and d + 5 times of Hash operations; compared with Hash operation, XOR operation has relatively small calculation quantity, so only the total number of Hash operations is considered hereby. Additionally, an important issue needing to be considered is that the calculation quantities of R and T are different due to their different hardware characteristics, and compared with T, R has stronger calculation ability. In an extremely short period, R only verifies one tag at each time, so the time for each verification event shall be minimized. In order to maximally balance the operation time of reader and transition, it is necessary to calculate the minimum costs of R and T. The calculation ratio of R and T is assumed as r: r=1 indicates that R and T have same Hash operation execution time; r=100 indicates that Hash operation execution of R is 100 times faster than that of T. The verification cost can be estimated according to different r values in order to obtain optimal δ and d values, as shown in Fig. 3. Therein, the verification cost depends on δ function of different r values, as shown in the following formula [18]:

$$Auth\_cost = (\delta \times d + 5) + (d + 5) \times r \qquad (7)$$

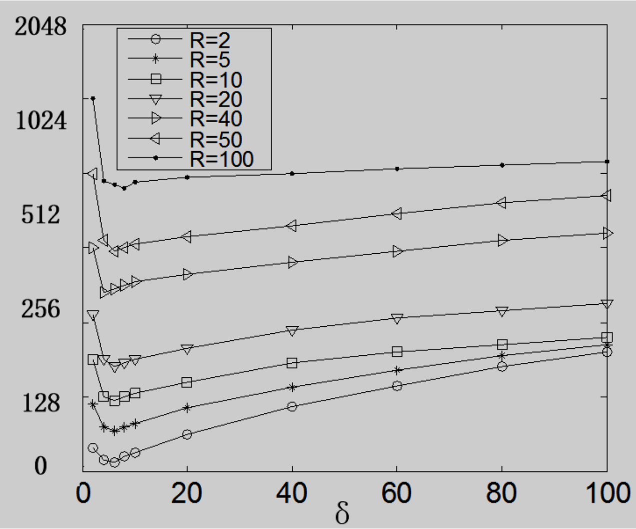

Figure 3. Authentication time cost function curve

B. *Case Study*

The total number N of tags may be different in different application programs. For example, when N $=2^{20}$ (one million), δ= 16 and d = 5, the secret key tree includes $16^5(=2^{20})$ nodes. This means that there are 2^{20} paths with different combination trees. For δ= 16, each internal node of the tree is the exponent of 16 and each value is the random number of l_k -bit. The path secret key of T is the combination of the manipulative indexing from root to leaf node. Due to CRFID protocol, R can only know 16 values randomly generated by each internal node. However, according to ACTION protocol, the values of secret key sub-items are the values of 1, 2, … 2^{l_k}. Additionally, the search complexity is also related to δ.

As mentioned above, ideal privacy protection scheme can generate function $P(c) = 1/(N - c)$, $\forall c \le N - 1$, so the length of l_k shall be more than 60 bits according to the privacy curve shown in Fig. 2 and the privacy function calculation formula in order to meet security requirement; in other words, each secret key sub-item shall be the random number with the length more than 60 bits. Although, each node (δ= 16) still only has 16 different sub-keys, the attacker cannot know 16 possible values, because each sub-key has 2^{60} mutually irrelevant possibilities. Therefore, if 1s needs to be taken to destroy 2^{20} Hash values, then the total time will be 2^{40}s rather than 2^{-4}s, thus significantly increasing the time needed for destruction. In a word, this scheme not only maintains the search complexity O(logN), but also greatly reduces the destruction possibility.

C. *Expansibility*

The initial design scheme aims at enabling each node to be only associated with one tag. Therefore, for the secret tree (δ,d) = (32,4), there are 2^{20} leaf nodes and the corresponding tag capacity is 2^{20}. Actually, the system can contain 2^{20} tags. In other words, one node may be associated with multiple tags. When the total number of tags is more than 2^{20}, the tolerance can still be ensured.

Although two tags have same secret key paths, they have different secret keys. The secret key updating mechanism can refresh new secret key path and login secret key, so the secret key path will be changed when one tag is destroyed and accordingly the attacker cannot track other tags belonging to the same leaf node. Additionally, tag confidentiality is determined by the unique secret key of the tag. Therefore, the scheme designed in the paper includes over 2^{20} tags. For most existing applications, 2^{20} tags are enough to meet relevant requirement.

V. Conclusions Conflict Of Interest Acknowledgment

A safe and efficient RFID authentication scheme is proposed in this paper, and meanwhile security analysis and discussion are also provided. In the scheme, the reader is designed to read the tag again after secret key updating, and if the previous secret key has established information and meanwhile the tag has responded thereto, then it is indicated that the secret key has not yet been completely updated in order to prevent desynchronized attack. Relevant simulation experiment result shows that the scheme proposed in this paper not only can main the same search complexity, but also can maximally improve the security level of various parts in secret key paths.

ACKNOWLEDGMENT

This work is supported by the Key Project of Guangxi Social Sciences, China (No.gxsk201424), the Education Science fund of the Education Department of Guangxi, China (No.2014JGA268), and Guangxi Office for Education Sciences Planning, China (No.2013C108).

REFERENCES

[1] J. He, Y. Geng and K. Pahlavan, “Modeling Indoor TOA Ranging Error for Body Mounted Sensors”, 2012 IEEE 23nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), Sydney, Australia Sep. pp. 682-686, 2012.

[2] Dingde Jiang, Zhengzheng Xu, Peng Zhang, et al., “A transform domain-based anomaly detection approach to network-wide traffic”, Journal of Network and Computer Applications, vol. 40, pp.292-306, 2014.

[3] Jie He, Yishuang Geng and Kaveh Pahlavan, “Toward accurate human tracking: modelling time-of-arrival for wireless wearable sensors in multipath environment”, IEEE Sensor Journal, vol. 14, No. 11, pp. 3996-4006, 2014.

[4] Jinping, Wang, Lv Zhihan, Zhang Xiaolei, et al., “3D Graphic Engine Research Based on Flash”, Henan Science, vol. 4, pp. 015, 2010.

[5] Li, Xiaoming, Zhihan Lv, Baoyun Zhang, Ling Yin, Weixi Wang, Shengzhong Feng, Jinxing Hu. “Traffic Management and Forecasting System Based on 3D GIS Cluster”, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on. IEEE, 2015.

[6] Li, Xiaoming, Zhihan Lv, Baoyun Zhang, Weixi Wang, Shengzhong Feng, Jinxing Hu. “WebVRGIS Based City Bigdata 3D Visualization and Analysis”. In Pacific Visualization Symposium (PacificVis), 2015 IEEE. IEEE, 2015.

[7] Li, Xiaoming, Zhihan Lv, Jinxing Hu, et al., “XEarth: A 3D GIS Platform for managing massive city information”. IEEE Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA). IEEE, 2015.

[8] Lv, Zhihan, Alex Tek, Franck Da Silva, Charly Empereur-Mot, Matthieu Chavent, and Marc Baaden. “Game on, science-how video game technology may help biologists tackle visualization challenges”. PloS one 8, No. 3, e57990, 2013.

[9] Lv, Zhihan, Tengfei Yin, Yong Han, Yong Chen, and Ge Chen. “WebVR——web virtual reality engine based on P2P network”. Journal of Networks, Vol. 6, No. 7, pp. 990-998, 2011.

[10] MA, Ruina, Zhihan LV, Yong HAN, et al., “Research and implementation of geocoding searching and lambert projection transformation based on WebGIS”. Geospatial Information, vol. 5, pp. 013, 2009.

[11] S. Li, Y. Geng, J. He, K. Pahlavan, “Analysis of Three-dimensional Maximum Likelihood Algorithm for Capsule Endoscopy Localization”, 2012 5th International Conference on Biomedical Engineering and Informatics (BMEI), Chongqing, China Oct. pp. 721-725, 2012.

[12] Su, Tianyun, Zhihan Lv, Shan Gao, et al., “3D seabed: 3D modeling and visualization platform for the seabed”. In Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on, pp. 1-6. IEEE, 2014.