Keeping It Under Control: A New Authentication Scheme for RFID Tags

Tongliang LI [1,2], Zhigang JIN[1], Hongguang YANG [2,3], Xiaokun SI [2,3]
[1.] Tianjin University, Tianjin, China
[2.] Hebei Institute of Applied Mathematics, Shijiazhuang, China
[3.]SJZ JKSS Technology Co.,Ltd, Shijiazhuang, China
litongliang@tom.com; zgjin@tju.edu.cn; hongguangyang@gmail.com; xiaokunsi@tom.com

Abstract—**The increasingly used Radio Frequency Identification (RFID) systems are facing with the security and privacy concerns for the illegitimate reading and potential tracking reason. Many schemes have been proposed by using of cryptographic technology, while they are lacking the people's involvement. In this paper, we propose a new crypto-based scheme that makes users involved in the tag's authentication process by using of physical privacy type assistant tags. Before identifying common tags, the reader needs to get some data from the assistant tags first. The data make the shared secret between the tag and the back-end database different. It not only keeps the tag can be read or not under user's control, but also makes the scheme safer even if the data on back-end database is revealed. Light weight block ciphers or hash functions can be employed in our scheme; a contrast of the performance is also given after the security and privacy analysis.**

Keywords-RFID; ownership transfer; security ; privacy; mutual authentication

I. INTRODUCTION

RFID (Radio Frequency Identification) is a kind of wireless technology, identifies an object or a person over short distances. A RFID system consists of RFID tags, readers and a back-end database. When a tag detects a signal that broadcasted by the reader, it replies to the reader required data to identify itself. Then reader can query the detail information of the tag from the back-end database.

Obviously, RFID systems benefit us many fields requiring retrieve information from objects automatically and efficiently, and they gain more popularity in areas such as supply chain management, automated identification systems and any place requiring identification of products or people. Every coin has two sides; due to the tag responds to the reader without alerting the tag's owner, people carrying the RFID tag may be surreptitiously tracked without their knowledge and consent. If a customer buys something with tags attached to, puts them in his/her bag, and the tag replies with a constant bit string (static identifier or protected identifier), the person bearing the tag will broadcast this value along his/her way, then a clandestine reader can track him/her. Likewise, if the tag replies with a value that can be related to a particular item, clandestine readers will be able to harvest information about the person/ object. So the security and privacy are two important issues in RFID systems.

Several kinds of method can enhance the security and privacy [1], such as ***Kill Function***: The tags are deactivated. ***Physical Privacy***: Reading of the RFID tags is physically restricted by some means, such as *faraday cage*. ***Cryptographic Scheme***: The readers communicate the tags with the protection of using cryptographic techniques. *Physical Privacy* offers a visual confirmation to the people that the tag is privacy protected. But it is costly to equip all the tags with *faraday cage*.

There are two types of tags: *active* and *passive*. The active ones have small power source, but the passive ones have not, they use the power generated by the reader and they cost efficiency of mass production. This paper takes focus on the low cost passive tags and proposes a new RFID authentication scheme for them. The scheme enables user to control a batch of tag's authentication session by controlling a *physical privacy* type assistant tag.

The remainder of this text is organized as follows. Section 2 discusses the related works. Section 3 describes the security and privacy problems. Section 4 presents our new scheme based on cryptographic and the analysis is shown in Section 5.

II. RELATED WORKS

The hash-lock [2] protocol from Sarma et al. uses hash of a key as the tag's metalID to avoid the leakage of the true ID. But the metalID is kept unchanged. Obviously, it is cannot protect location privacy. Later, they use challenge/ response mechanisms to give a randomized hash-lock protocol [3]. When the reader finds the correct tag ID, it will send it to the tag in plaintext and intruder can eavesdrop on it. Ohkubo et al. use two hash functions in their hash-chain protocol [4]. But it is vulnerable to replay attacks. D. Henrici et al. 's protocol use random numbers to update the tag's information [5]. If a malicious reader sends *zero* as the random number, it makes the tag and the back-end database to update respective new tag ID with different data. In order to solve the de-synchronization problem, Ha et al. proposes a protocol called LCSS [6]. In fact, it is still vulnerable to de-synchronization attacks. The lacking of random numbers' protection and verification leads to the weakness in reaching their goal. Due to the property of *XOR*

operation used in the protocol, Song et al.'s protocol [7] can not reach their goal of making the intruder hard to construct the messages without the tag ID. Molnar et al. 's protocol MSW [8] is the first protocol explicitly supports ownership transfer, and it is a pseudonym type protocol. Those pseudonyms are the tag's secrets and they are organized in a tree structure. The tag stores the secrets corresponding to the path from the root to the tag. It needs more memories in tags. Lim et al. 's protocol LK [9] needs each tag storing a tag secret, a server validator, a counter, and making use of three pseudorandom functions to support tag ownership transfer. It requires a great amount of computations and storage capacities of the tag and the back-end database. Fouladgar et al. [10, 11] present two protocols for tag ownership transfer. In both of the protocols, the tag stores two secret keys for computing the pseudonyms and updating keys, and the back-end database stores the tag's ID. However, the first protocol is vulnerable to replay attacks and the second one may face with being traced past communications of the tag if the intruder compromises it. Lee et al. 's protocol [12] is an ultra lightweight one based on the *Rot* operation. If an intruder eavesdrops on the *n*th and the *n+1*th session's messages between tag and reader, and then he/she will be able to calculate the secret key.

III. PROBLEM STATEMENT

Due to the messages are transferred via electromagnetic waves, RFID systems are more susceptible to be attacked than the wired one because the intruder is able to harvest them without any connections. There are many kinds of attacks: *replay attacks*, *impersonation attacks*, *parallel session attacks*, *man-in-the-middle attacks*, *reflection attacks*, *interleaving attacks*, *de-synchronization attacks* etc. And some privacy threats should be resisted [7, 10], such as:

- **Tag information leakage**: If an unauthorised reader obtains a tag identifier, it may be able to access the related information held in the back-end database.
- **Tag location tracking**: If a tag's responses are distinguishable from other tags, then the tag's location could be tracked by unauthorised readers.
- **Backward traceability**: If a tag is compromised by the intruder, it might be traced the previous communications.
- **Forward traceability**: if a tag compromised by the intruder, it might be traced the future communications.

According to the intruder's methods, those attacks can be simply classified into three groups:

- **Compute the secret.** The intruder computes the secret directly based on the information he/ she had got.
- **Construct the messages.** The intruder constructs some message that qualifies a legal authentication session.
- **Find the relationship.** The intruder gets the relationship between the tag and a person/object or the relationship between the tag and the position.

Bessie, we can define the further requirement:

Definition: The intruder who gets the ID table from the back-end database cannot know which is the correct ID belonging to a given tag.

In order to resist those attacks, some suggested mechanism can be used in the designing in our scheme, such as mutual authentication, pseudonym, message freshness etc.

IV. PROPOSED SCHEME

The main idea of our scheme is adding user's control to the authentication process by using a *Physical Privacy* type assistant tag, and the user can control it whether this assistant tag can be read or not. When reader wants to identify some common tags, it must authenticate the assistant tags first and get some data from the assistant tag. The data is the lynchpin of identifying the common passive tags.

Before proposing our scheme, we define some notations in Table I.

TABLE I. NOTAIONS

Notation	*Meaning*
F( )	A kind of fucntion, lightweith AES, hash etc.
$\oplus$	Exclusive-or (*XOR*) function.
$\|\|$	Concatenation.
MSG $\rightarrow$	Send messages.
Query	Reader send query command to tags.
Gen	Generate random numbers.
Find X:Y	Find *X* that satisfy the condition *Y*.
Verfy:X	Check wether *X* is true.

In addition, $\boldsymbol{D}$, $\boldsymbol{D_a}$ denote the back-end database which keeps the tag's information and keeps the assistant tag's information. $\boldsymbol{T}$, $\boldsymbol{T_o}$, $\boldsymbol{T_c}$ denote the tag to be identified, the object's original owner's assistant tag, the customer's (new owner's) assistant tag; $\boldsymbol{ID}$, $\boldsymbol{ID_o}$, $\boldsymbol{ID_c}$ denote $\boldsymbol{T}$, $\boldsymbol{T_o}$, $\boldsymbol{T_c}$'s ID and $\boldsymbol{Key}$, $\boldsymbol{Key_o}$, $\boldsymbol{Key_c}$ denote the *shared secret* between $\boldsymbol{T}$, $\boldsymbol{T_o}$, $\boldsymbol{T_c}$ and the back-end database

As a premise, $\boldsymbol{R}$ is trusted, it must be authenticated when connect $\boldsymbol{D}$ and $\boldsymbol{D_o}$, and the communication channel between them are trusted.

A. Identify assistant tag phase

We start it from a bookstore. When a customer wants to buy the some books, the reader needs to authenticate T_o and T_c first, and the two processes are similar. The following step is based on the authentication of T_o and the process is shown in Fig.1.

At the beginning, the tags store the *ID* and the *Key*, and it is the same with the back-end database.

Step 1: $\boldsymbol{R}$ broadcasts to $\boldsymbol{T_o}$ a query with a new generated random number R_o.

Step 2: $\boldsymbol{T_o}$ generates a random number R_o^* and computes:

$M_1 = F(ID_o, R_o) \oplus Key_o$

$M_2 = R_o^* \oplus ID_o$

$M_3 = F(R_o, Key_o) \oplus F(ID_o \oplus R_o^*)$

and sends them to $\boldsymbol{R}$.

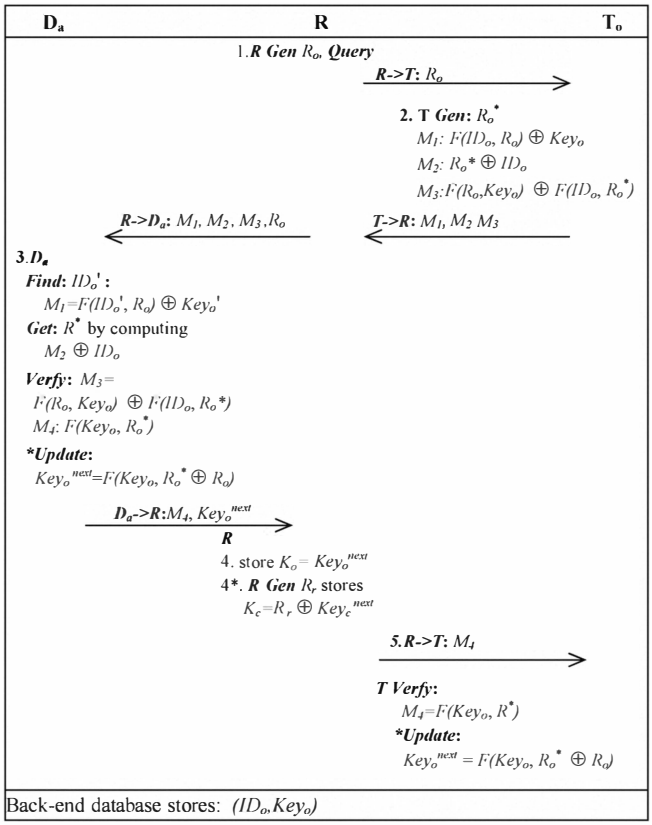

Figure 1. Identify assistant tag phase

Step 3: When ***R*** gets M_1, M_2, M_3, it sends them and R_o to D_a, then D_a searches whether there is a record (ID_o', Key_o') that makes $M_2=F(ID_o',R_o) \oplus Key_o'$. The right answer indicates ID_o' equals to ID_o, Key_o' equals to Key_o. Then D_a can get R_o^* by computing $R_o^* \oplus ID_o \oplus ID_o'$. After that, D_a checks whether $M_3=F(R_o, Key_o') \oplus F(ID_o, R_o^*)$. If it is true, D_a updates $Key_o^{next} = F(Key_o, R_o^* \oplus R_o)$, and sents $M_4 = F(Key_o,R_o^*)$, Key_o^{next} to ***R***.

Step 4: When ***R*** gets ID_o, Key_o^{next}, it stores Key_o^{next} as K_o, transmits M_4 to T_o. In authentication process of T_o, ***R*** stores $K_o = Key_o^{next}$

While in authentication process of T_c, ***R*** generates a random number R_r, stores $K_c=Key_c^{next} \oplus R_r$ for identifying common tags phase. (4* shows it in Fig.1.)

Step 5: T_o checks whether $M_4=F(Key_o, R_o^*)$, if it is right, it will update its new $ID_o^{next} = ID_o \oplus R_o \oplus R_o^*$.

After the authentication of T_o and T_c, ***R*** stores K_o and K_c.

B. Identify common tag phase and ownership transfer

After authenticating the assistant tags, the identifying common tags can be processed. Fig.2 shows it.

Step 1: ***R*** broadcasts to ***T*** a query with a random number R.

Step 2: ***T*** generates a random number R_o^* and computes:

$M_1=F(ID, R) \oplus Key$;

$M_2=R^* \oplus ID$

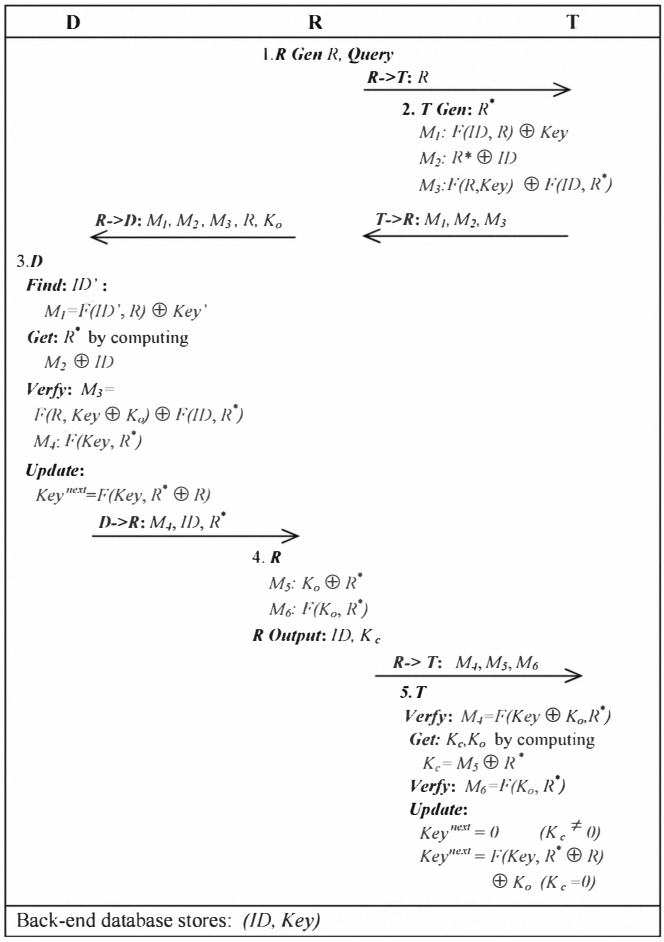

Figure 2. Identify common tag phase

$M_3=F(R, Key) \oplus F(ID, R^*)$

and sends them and K_o to ***R.***

Step 3: When the ***R*** gets M_1, M_2, M_3, it sends them and R^* to ***D***, then ***D*** searches whether there is a record (ID', Key') that makes $M_2=F(ID', R) \oplus Key' \oplus K_o$. The right answer indicates ID' equals to ID, Key' equals to Key. Then ***D*** can get R^* by computing $R^* \oplus ID \oplus ID'$. After that, ***D*** checks whether $M_3 = F(R,Key \oplus K_o') \oplus F(ID,R^*)$. If it is true, ***D*** updates, $Key^{next} = F(Key, R^* \oplus R)$, and computes:

$M_4=F(Key,R^*)$

and sends it and ID and R^* to ***R***.

Step 4: When ***R*** gets M_4, ***R*** computes:

$M_5= K_o \oplus R^*$

$M_6=F(K_o, R_o^*)$

and sends them to ***T,*** outputs K_c and IDs for the customers by some means.

Step 5: ***T*** checks whether $M_4 =F(Key \oplus K_o,R^*)$. If it is right, it can gets K_c by computing $K_c= M_5 \oplus R^*$, and checks whether $M_6=F(K_o, R^*)$. If K_c is not equal to *zero*, ***T*** updates $Key^{next}=0$, otherwise, ***T*** updates $Key^{next}= F(Key, R^* \oplus R) \oplus K_o$ to finish the whole session.

C. Remarks

We should take notice that the assistant tag $\boldsymbol{T_o}$ is indispensable even if the ownership transfer is not necessary. *Key* stored in $\boldsymbol{D}$ is **NOT** equal to that in $\boldsymbol{T}$, it equals to $Key \oplus K_o$. That makes the reading of tags need $\boldsymbol{T_o}$, which is controlled by a person. If the clerk just makes an inventory, he/she can press a button to ignore identifying the $\boldsymbol{T_c}$ and K_c will be *zero*. If ownership transfer is demand, the assistant tag $\boldsymbol{T_c}$ for customer is must be read. When the book is sold to the customer, *ID* and *Key* is updated and the ownership is transferred. $\boldsymbol{T_c}$ becomes $\boldsymbol{T_o}$ for the customer, and he/she can import the $\boldsymbol{R}$'s outputs to his/her own database.

Here is a problem we should pay attention to. If an assistant has the jurisdiction to identify ten common tags, once the reader authenticates the assistant tag, all of the ten tags must be identified, or some of them or all of them cannot be identified any more. So in identifying assistant tag phase, the *update key operation* is optional. In fact, the common tag's key is relate to the assistant tag's key, if update step is employed, after the identifying assistant tag, the key_o or key_c would be changed, if not all of the common tag be identified, some of their key will not be relate to the assistant tag's key.

V. SCHEME ANALYSIS

We perform the security analyses under the Dolev-Yao intruder model [13]. In this model, the intruder may eavesdrop on any message exchanged between tag and reader, modify or block any message sent from tag to reader, and may inject his/her own messages.

A. Security Analysis

Random numbers play a very important role in many protocols; they are as important as the *ID* and *key* and they are also a kind of secret. With the help of them, the scheme has enough unknowns to the intruder and has enough message freshness to resist the intruder's sleight.

Every message, except the first query message, transferred between tag and reader has random numbers, and each of them has at least two unknowns to the intruder. The intruder may eavesdrop on all the message of several consecutive authentication session of a certain tag, and put those messages into his/her knowledge base. Even though he/she can utilize the *XOR* function, and get some intermediate result, but there is still at least 2 unknowns in the result. The only way to calculate the secret is to do a bruteforc, and it is inefficiency.

Each message will be used once, so *replay attacks* are prevented and the simple copy of information of the tag by *eavesdropping* is not possible, that avoids the *tag information leakage*.

The messages consist of fresh *IDs, Keys* and fresh random numbers and they are all unknowns to the intruder, all of the secret data, including credible random numbers, are protected each other by *F()*, and they are not plaintext, moreover the received messages are checked by another. For example, when back-end database gets M_3, it can check whether M_1 and M_2 are correct or unchanged by computing M_3. Mutual authentication can frustrate *spoofing attacks* and *man-in-the-middle attacks*. Challenge response mechanisms are employed to achieve it.

Forward security means data transmitted today will still be secure even if secret tag information is revealed by tampering in the future. Fortunately, this scheme supports it. In fact, *ID* is the secret information, and *ID* and *Key* changed after each session. The intruder may know Key^{next} in the future, but he/she cannot know what R_a and R_a^* are, so data transmitted today will still be secure.

B. Privacy Analysis

In this scheme, without the owner's permission, the target tag cannot be read, and the combination of *Key* and *ID* is regarded as the pseudonym in the authentication process. There is no plaintext transferred and every message changes for each session even for the same tags. So the tag anonymity is guaranteed and it can avoid partial information leakage about a tag's location.

At the end of process, *Key* will be changed (in other words, the pseudonym will be changed) and the intruder cannot do a forecast that which pseudonym will be the next when he/she knows the current. Dynamicity of pseudonym can make the intruder who gets the *(ID, Key)* table from the back-end database not knowing which is the correct *ID* belonging to the certain tag and he/she can not distinguish the tag's output from random, so the tracking of a tag owner is impossible. Furthermore the *Key* is also changed when the tag belonging to a new owner. Due to the reader generates a random number R_r which is used to compute K_c in the identifying assistant tag phase, and only the honest reader know it, so both backward and forward traceability are supported.

In addition, the intruder needs D, D_a to ascertain a tag, the intruder cannot know which is the correct *ID* belonging to a given tag even he/she gets the (*ID*, *Key)* table, because he/she do not know what is the assistant tag.

C. Performance Analysis

After identifying an assistant tag, lots of tags can be identified, so it is not a burden for the reader to read the assistant tag's information.

Low-cost RFID tags are a kind of limited resource devices, with only a small amount of memory. We assume that all components are L-bit length. Our scheme is based on L-bit index-pseudonyms and a tag has to store it's ID. Each tag should have an associated key of L-bit length, which is used for the pseudonym and mutual authentication. The back-end database stores the same information, so the total required memory is $2L$ bits. Low-cost RFID tags are also lack of computation ability; they don't have plentiful logic gates and power supplies. To suit this situation, we must use some lightweight functions for the tag. The selection of *F()* not only influences the security, but also influences the performance. Traditional hash function needs more than 16K gates [14], which is higher than the capabilities of those low cost passive tags. An efficient AES only needs around 3400 logic gates [15]. We can see from the Table II that AES is more efficient and more practicable than traditional hash function.

TABLE II. THE PERFORMANCE ANALYSIS

F()	**Identify assistant tag phase**		
	Tag	***Reader***	***Back-end Database***
AES	*1RNG, 6XOR, 5F*	*1RNG, 1XOR*	*5XOR, 5F*
Hash	*1RNG, 11 XOR, 5F*	*1RNG, 1XOR*	*10 XOR, 5F*

x*RNG*,x*XOR* ,x*F* mean do *x* times generate a random number, *XOR* operation, *F()* function.

D. Contrast with Previous Works

Firstly, our scheme makes users involved in the tag's authentication; it is keeping the process under users' control.

Secondly, the shared secret *key* between common tags and the back-end database is not the same, which makes the scheme safe even if the data on back-end database is revealed to the intruder.

Finally, we pay enough attention to the tag ID, and we design it as a dynamic type, this can avoid the drawbacks on [2]. The messages that contain *ID* are protected by fresh random numbers and *Keys*, and they are not plaintext, so our scheme can resist the threat in [3]. With the enough message freshness and protection, this scheme is immune to the replay attacks which [4, 10] are vulnerable to. Due to the protection of the message, the weakness in [6] can be avoided. In [12], the message leakage makes the intruder get the secret key, while in this scheme no such weakness can be utilized. Ref. [8-11] are the ownership transfer supported protocols. Our scheme needs less storage resources than [8, 9]. Due to the reader play an important role in the ownership transfer session, it generates a random number which is to be a part of the new key, so it needn't worry about being traced past communications of the tag when the intruder compromises the tag. As we have talked above, this scheme makes the intruder has no chance to send *zero* as *R*, so the de-synchronization attacks is inefficiency.

If *F()* is a hash function, then $F(ID, R)=Hash(ID \oplus R)$, and if the intruder controls a reader and selects *zero* as the *R* (a credible random number) to begin his/her sleight. M_1, M_2, M_3 in identifying assistant tag phase will be:$M_1=Hash(ID_o) \oplus Key_o$, $M_2=R_o^* \oplus ID_o$, $M_3=Hash(Key_o) \oplus Hash(ID_o \oplus R_o^*)$. It is still helpless to the intruder; there are still at least two unknowns in each message. he/she cannot construct them, and it is the same with other messages. So the intruder can not reach the goal, and Ref. [5, 7] 's demerit can be prevent.

VI. CONCLUSION

In this paper, we present a new scheme for RFID tags. To resist the attacks, we use pseudonym mechanisms and challenge/ response mechanisms to make the scheme stronger. Taking the people involved in the indentifying of a tag makes the whole process a little complex, but it takes us many benefits. In order to achieve our goal, we employ assistant tags. Both common identifying the tag process or owner transfer process is under the users' control and the security and privacy can be strengthened.

ACKNOWLEDGMENT

This work was supported by the Science and Technology Key Project of Hebei under Grant No. 09206917D and Nature Science Foundation of Hebei under Grant No. 08M009.

REFERENCES

[1] D. Singelée and S. Seys "User Privacy in RFID Networks," https://www.cosic.esat.kuleuven.be/publications/article-1279.pdf, 2009.

[2] S. E. Sarma ,S.A. Weis and D.W. Engels, "RFID Systems and Security and Privacy Implication," The 4th International Workshop on Cryptographic Hardware and Embedded Systems,USA, pp. 454–469, 2002.

[3] S. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and Privacy Aspects of Low-cost Radio Frequency Identification Systems". Security in Pervasive Computing, LNCS, vol.2802, pp.201-212,2003.

[4] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Hash-chain Based Forward-secure Privacy Protection Scheme for Low-cost RFID," The 2004 Symposium on Crytography and Information Security, Sendai, pp. 719-724, 2004

[5] D. Henrici and P. Muller, "Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices Using Varying Identifiers," The 2nd IEEE Annual Conference On Pervasive Computing and Communications Workshops, USA, pp.149-153, 2004.

[6] J.C. Ha, S.J. Moon, J. M. G. Nieto and C. Boyd, "Low-Cost and Strong-Security RFID Authentication Protocol," EUC Workshops 2007, LNCS 4809, Taipei, pp.795-807, 2007.

[7] B. Song and C. Mitchell, "RFID authentication protocol for low-cost tags" . First ACM Conference on Wireless Security, pp.140-147, 2008.

[8] D. Molnar, A. Soppera, and D. Wagner. "A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags," Selected Areas in Cryptography, Canada, 2005, Vol.3897, pp.276–290.

[9] C. Lim and T. Kwon. "Strong and robust RFID authentication enabling perfect ownership transfer," Information and Communica-tions Security — ICICS '06, USA, vol.4307, pp.1–20, 2006.

[10] S. Fouladgar and H. Afifi. "An efficient delegation and transfer of ownership protocol for RFID tags," The First International EURASIP Workshop on RFID Technology, Austria, 2007.

[11] S. Fouladgar and H. Afifi. "A simple privacy protecting scheme enabling delegation and ownership transfer for RFID tags," Journal of Communications, vol.2, no.6, pp. 6–13, 2007.

[12] Y.C. Lee, Y.C. Hsieh, P.S. You, and T.C. Chen "A New Ultralightweight RFID Protocol with Mutual Authentication," . In Proc. of WASE'09,vol 2 of ICIE, pp.58-61, 2009.

[13] D. Dolev, and A.C. Yao, "On the Security of Public Key Protocols" . IEEE Transactions on Information Theory. vol.2, no.29, pp.198–208, 1983.

[14] Datasheet Helion Technology. "MD5, SHA-1, SHA-256 hash core for Asic," http://www.heliontech.com, 2005.

[15] M. Feldhofer, J. Wolkerstorfer and V. Rijmen, "AES Implementation on a grain of sand," Information Security,IEEE proceedings, vol.152, no.1, pp.13-20, 2005.