A Light-weight Security Protocol for RFID System

Jung-Hyun Oh, Hyun-Seok Kim, Jin-Young Choi,

Computer Theory and Formal Methods Lab,
Dept. Of Computer Science and Engineering , Korea Unversity,
5-1 Anam dong, Sungbuk gu, Seoul, 136-701, Korea,
{jhoh, hskim, choi}@formal.korea.ac.kr

Abstract. RFID is automatic object identifying technology via radio frequency. And its application areas are un-describable for its convenience and pervasiveness. However, because the communication channel between the verifier and the tag is wireless, serious privacy problems such as the data leakage and the data traceability can be occur. Without resolving these privacy problems, RFID system cannot be adapted fully in any area. Many kinds of security protocols have been proposed to resolve these problems. However, previous proposals did not satisfy security requirements and still leaved vulnerabilities. In this paper, we describe the security vulnerabilities of previous works for RFID systems. Finally, we propose a security protocol which based on one-time pad scheme using random nonce and shared secret values. The proposed protocol satisfies security requirements such as the data secrecy, data anonymity and the data authenticity between the verifier and the tag. We have proved security requirements satisfaction formally by using GNY logic.

Keywords: RFID, Security protocol, One-time pad, GNY logic

1 Introduction

RFID system is automatic object identification system using radio frequency signal. The small tags(transponders) attached to the products carries the unique information of the products and whenever the verifier(transceiver) request the product specific information tags transmit the information via radio frequency signal. The verifier relays the received information to the back-end DB whether the information is valid or not. Because of the RFID system uses the radio frequency, objects can be identified easily and quickly, and the management of the object could be efficient. Because of its merits, the attempt to apply the RFID system to many areas is in progress.

However, there are several problems that RFID system should resolve before their pervasive deployment. The problems in RFID system are privacy related problems such as *the data leakage* and *the data traceability*. These problems occur because the communication channel between the verifier and the tag is wireless. Simply eavesdropping the messages that transmitted between the verifier and the tag, the attacker can obtain the unique information of the tag, and the attacker can make private information profiles of the tag carrier. They also can track tag carrier without any authorization.

To resolve these privacy problems, security measures for the RFID system should satisfy several security requirements which are described down below.

- *Data Secrecy* means that any transmitted data between the RFID system components should not be understandable to the attacker. The tag stores the unique ID and the verifier identifies the tag by receiving the unique ID of the tag. However, if the unique ID of the tag is exposed to attackers, it could be used in identifying tag carrier's private items and making the tag carrier's private information profile.
- *Data Anonymity* means that any transmitted data between RFID system components should not be distinguishable to the attacker. Even though the attacker could not understand the meaning of the messages between the RFID system components, the attacker could track the tag or the tag carrier if the transmitted messages are fixed and being used in every session.
- *Data Authenticity* means that any transmitted data between the RFID system components should be authenticable. That is, the messages transmitted between the RFID system components should be check whether or not they are from the honest entity. If there is no measure for data authentication, the attacker would attempt to authenticate himself to the honest entities by using the previously obtained messages. And if attacker succeed in authenticating himself to the honest entities, the information such as the tag ID leakage could be possible.

The contributions of this paper are outlined as follows: First, we describe security vulnerabilities of the previous works based on three security requirements mentioned above and then we propose a light-weight security protocol based on one-time pad scheme. One-time pad scheme is proven to guarantee the perfect secrecy of the message by Shannon[8]. In our proposal, one time pad scheme with the secret values and a fresh pseudonym is applied to satisfy the three security requirements.
Second, using GNY logic[9], we proved formally the satisfaction of security requirements in our proposed protocol. In particular, GNY logic provides several notations and logical postulates that help to express the security protocol and to deduce the security requirements(goals) of the protocol logically. If the security requirements deduction of the protocol is failed, then we cannot assure that this protocol satisfies the security requirements. Because of its precise protocol expression and verification capability, GNY logic is known as one of the successful skills for formal security verification methods[17].

This paper is organized as follows: In section 2, we summarize the previous works and its security vulnerabilities. In section 3, we describe our proposal. In section 4, we present the analysis of the proposed protocol using GNY logic and discuss the module aspect of security for RFID system. Finally, the conclusion is addressed in the last section.

2 Related Works

Many proposals have been proposed to satisfy the security requirements. In this section, we have categorized the previously proposed protocols as down below.

Hash Function Based Security Protocol In [2], Weis *et al*. proposed Hash-lock protocol and the randomized hash-lock protocol. In these protocols, the tag stores the ID and the unlocking key and stay in locked state before the reader request the ID. When the reader attempts to identify the tag, the tag sends the hashed ID to the verifier. Then the reader seeks the unlocking key ID in the back-end DB with the received hashed because the tag unlock itself only when the reader sends the unlocking key. But the unlocking key is transmitted without any encryption, so the attacker can obtain the unlocking key. Therefore, *the data secrecy, the data anonymity* and *the data authenticity* are not satisfied.

In[3], Henrici *et al*. proposed the hash-based ID variation protocol which ID of the tag varies in each session with help of fresh nonce. However, the hashed ID is also sent to the reader to make unlocking key searching easier. Therefore, even though the ID of the tag is not exposed, the attacker could track the tag because the fixed hashed ID is used in every session. Therefore, *the data anonymity* is not satisfied.

In [4], Okubo *et al*. proposed the hash chain protocol which used two different one way hash functions to send the hashed ID to the reader and to update the ID after the authentication procedure is completed. However, the hashed ID should be synchronized to the back-end data-base. Therefore, the hashed ID could be used in authentication for attacker before the tag authentication session is completed. And the attacker could de-synchronize the ID between the tag and the back-end DB, the tag can be un-identifiable. Therefore, hash chain scheme does not satisfy *the data authenticity*. Moreover, this scheme gives the great burden to the DB for a single tag authentication.

Arithmetic Calculation Based Security Protocol In [5], Juel proposed minimalist cryptography using one-time pad scheme for low-cost RFID system. However, storing the triple shared keys and number of padding vectors in a single tag will cost large amount of gates to implement. Moreover, in the authentication procedure, the triple shared keys are exposed to the attacker. Therefore, the attacker could use these triple shared keys to authenticate himself to the reader just before the honest tag authentication is completed. So, this scheme does not satisfy *the data secrecy* and *the data authenticity*.

In [7], Juel *et al*. proposed a light-weight security protocol based on HB algorithm which is known to secure to both passive attack and active attack. The HB algorithm is based on hardness of the learning parity bit with noise[12]. Guessing the plaintext of the encrypted message is computationally infeasible, because guessing the plaintext is identical to solving the LPN problem. However, several papers have shown that several security vulnerabilities are still exist in the HB based protocols[13][14]. Moreover, it is open question whether the LPN problem based protocols are provably resistant to man-in-the-middle-attack or not.

In [18], Vajda *et al*. proposed light-weight security protocol using modular product such as XOR. In this scheme, the reader and tag use the secret key which is padded with random bit to authenticate each other. In [19], Defend proved that this scheme does not satisfy *the data(secret key) secrecy* and *the data authenticity* Because the fixed key is continuously used in a session before the authentication step is completed, the initially shared secret key and newly computed secret key for current session can be leaked by padding the two messages transmitted between the reader and tag.

3 Our Proposal

We have described that the previous proposed schemes did not satisfied the security requirements of the RFID system. In section, we present our protocol that satisfies the three security requirements.

Initial assumptions for the security protocol design Firstly, our scheme is focused on to design a security protocol which can be adapted into the class 1 generation 2 standard(proposed by EPC-Global) based RFID system. That is, the bit-length of a tag ID is 128. Secondly, the communication channel between the tag and verifier is wireless and between the verifier and back-end DB is, generally, wired. And, also, the hardware capability of the verifier and back-end DB is considered to be limited. Therefore, it is convenient to assume that the communication channel between the verifier and back-end DB is secure. Hence, we assumed that the verifier and the back-end DB work as one component, the verifier.

One-Time Pad Scheme Shannon has proved that the perfect secrecy of the message is guaranteed if and only if the message is padded with a fresh pseudonym which bit length is the same or longer than the message[8]. Encryption and decryption in one-time pad scheme the plaintext is combined with a random secret key, *K*, that is as long as the plaintext, *x*, and used only once. A modular addition, such as XOR, is used to combine the plaintext with the random secret key as described in (1).

$$E_K(x) = x \oplus K = D_K(x). \quad \textbf{(1)}$$

We applied this advantage to our protocol in encryption and decryption. In our protocol, the unique tag ID is padded with a fresh pseudonym. Because of the ID is padded with the fresh pseudonym as long as the ID, the perfect secrecy of the ID can be satisfied. Therefore, with using this scheme we can satisfy the security requirements such as *the data secrecy*.

$$E_{Pseudonym}(ID) = ID \oplus Pseudonym. \quad \textbf{(2)}$$

The generation of the fresh pseudonyms will be explained in detail at the next paragraph. This fresh pseudonym can be generated by both verifier and tag, and any information related to pseudonym generation will not be leaked. Therefore, the attacker could not decrypt the tag ID out of the message, $E_{Pseudonym}(ID)$, unless the attacker happened to obtain the this padded pseudonym.

Generating Padding Pseudonym To decrypt the padded ID, the verifier should know this pseudonym. The pseudonym will be generated by padding or concatenating the two fresh nonce, N_T and N_V, which are generated by the tag and the verifier ($Pseudonym = (N_T \oplus N_V)$ or $(N_T \| N_V)$, pseudonym generation methods depends on the bit length of the secret value and nonce). The delivery of each nonce to each other(the

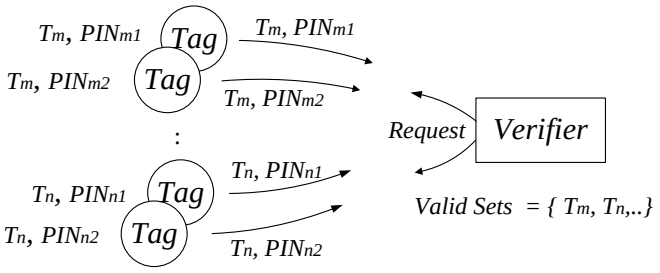

Figure. 1. The secret value set management between the different subset of RFID tags

verifier and tag) will be done by using the secret values in the secret value set which is initially shared between the verifier and the tag (Secret value set $T_x = \{S_1, S_2, \ldots, S_n\}$). The nonce N_V generated by the verifier will be delivered to the tag by padding with selected secret value S_x ($S_x \in T_x$). The nonce N_T generated by the tag will be delivered to the verifier by padding with the selected secret value S_y ($S_y \in T_x$).

$$E_{Sx}(N_V) = N_V \oplus S_x. \quad \textbf{(3)}$$

$$E_{Sy}(N_T) = N_T \oplus S_y. \quad \textbf{(4)}$$

However, a tag cannot store secret value set which contains many secret values because of its memory size or implementation cost. To make the same efficiency such as storing a number of secret values with reasonable number of secret values by considering the memory size and implementation cost, whenever the secret values are needed to deliver the pseudonym, the tag permutes the sequence of the stored secret values order and chooses necessary amount of secret values from the permuted order. For example, if stored secret value set is $T_x \rightarrow \{ S_1, S_2, .. , S_n\}$ and if two secret values are to use in padding then available number of permuted secret value pairs are ${}_nP_2$. By this method, we can gain additional number of secret value orders without storing large number of secret values. The tag selects the permutation index number, *PIN*, which carries the information about the order of the permuted secret value set. And then the tag delivers the *PIN* to the verifier for reporting the verifier which secret value order should be used for nonce delivery

Secret Value Set Management The management method of the secret values was inspired by the internet banking authentication scheme. To use internet banking, we should register our personal information in the bank and receive the secret card where a random secret value set is printed on. To use the internet banking service, the internet banking server requires you to put the specific secret values from the secret value set on the secret card for the user authentication. The secret card has several types with different secret value sets. Therefore the exposure of one type of the secret card cannot lead to the entire internet banking service failure.

Likewise, the secret value set stored in each tags should not be the same values to all tags or unique. If the shared secret value sets are the same in the tags, exposure of

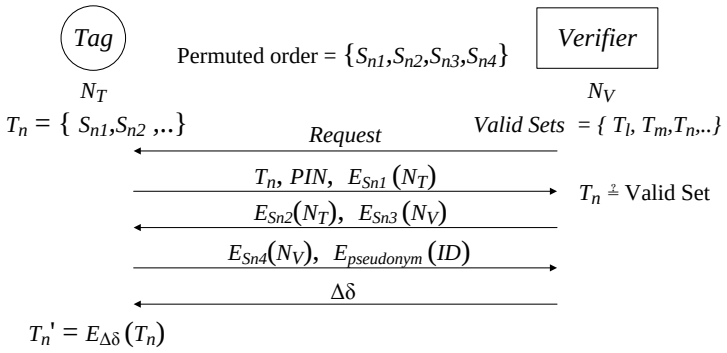

Figure. 2. The message sequence of the protocol

the one secret value set can lead us to entire information leakage of the RFID system. On the other hand, if the shared secret value set in each tag is unique, then the management of the secret value sets will be extremely difficult and this uniqueness may be used for identifying the tag. To resolve these problems, designing some subset of tags to use the same shared secret value set and the other subset of tags to use different secret value set will be the best way. Every secret value set stored in tags has label like T_x or T_y and these labels should be sent for the verifier to recognize what secret value set is being used in current session. The attacker cannot derive any information about the secret values from these labels, because the label will be assigned randomly such as, T_{123} or T_{201}. The basic idea of secret value set management between the different subset of RFID tags is depicted in Figure.1.

For *the forward secrecy*, shared secret value set will be updated at the end of the successful communication. The verifier sends the update message $\Delta\delta$ to the tag for secret value set update, and the tag update secret value set T_n to T_n' by padding the T_n with $\Delta\delta$.

$$E_{\Delta\delta}(T_n) = T_n \oplus \Delta\delta \quad \textbf{(5)}$$

And the updated secret value set T_n' should be valid set which is already stored in the DB.

Protocol Steps

1.Verifier	Request the tag its unique ID
2.Tag	Permute order of T_n. Selects four secret values out of permuted order of secret values, ${}_nP_4$, and puts permutation order information in the *PIN*. Generates a random nonce N_T and encrypts it by padding S_{n1}. Sends messages, T_n, *PIN* and $E_{Sn1}(N_T)$ to V.
3.Verifier	Finds T_n in the *DB.* Permutes T_n with *PIN*. R decrypts the $E_{Sn1}(N_T)$ and encrypts the N_T by padding S_{n2}, to authenticate himself to T. Generates N_V and encrypts it by padding S_{n3}. Sends $E_{Sn2}(N_T)$ and $E_{Sn3}(N_V)$ to T.
4.Tag	Decrypts the $E_{Sn2}(N_T)$ to authenticate the V by verifying the N_T. If

	$D_{Sn2}(E_{Sn2}(N_T)) = N_T$ then authenticate V. T decrypts the $E_{Sn3}(N_V)$ and encrypt the N_V by padding S_{n4}, to authenticate himself to V. Generates *pseudonym* by padding two nonce N_V, N_T. Pad the ID with *pseudonym*. Sends E_{Sn3} (N_V) and $E_{Pseudonym}(ID)$ to V
5.Verifier	Decrypts the $E_{Sn4}(N_V)$ to authenticate T by verifying the N_V. If D_{Sn4} ($E_{Sn4}(N_V)$) = N_V, then V authenticate T. Generates *pseudonym* for ID decryption by padding two nonce N_V, N_T. Finds ID in the *DB* and updates T information. *DB* selects the available secret value set T'_n and calculates the secret value set update message $\Delta\delta$ which can be used in updating the set T_n to T'_n Sends the $\Delta\delta$ to T
6.Tag	Updates the secret values set T_n by padding with $\Delta\delta$

4 Analysis

4.1 Security Correctness Proof

Frequently, informal and intuitive way of security verification has been used in previous proposals. However, informal and intuitive way of verification can leave design flaws and security errors undetected. For example, Needham-Schroeder protocol which was believed to be secure, and the BCY protocol which was considered to be secure in mobile communication, formal verification methods proved its security vulnerabilities[15][16].

With formal verification, we can assure that there is no undetected design flaw in system which we want to verify the security. In formal methods, there are two major methods in verification, the model checking and the theorem proving. The verification using theorem proving is logical deduction step that help to deduce the security goals of the target system. If the security goals are successfully deduced, we can assure that the target system has no design flaws.

In this section, we analyze the security correctness of our proposed protocol using theorem proving method which name is GNY logic. GNY logic is considered to be one of the successful methods for the security protocol verification[17]. With GNY logic, we have proved security goals of our proposed protocol; the tag and verifier can assure themselves that received messages are delivered from trustable agent and these messages are fresh so that they can also assure themselves that this messages are never been used before this session. Because *the data secrecy* is proved by Shannon, we are focus on proof of *the data authenticity* and *the data anonymity* satisfaction. The security verification using GNY logic involves four steps, the formalization of the protocol messages, the specification of the initial assumptions, the specification of the protocol goals and the application of the logical postulates. The precise meaning of notations and logical postulates is described in appendix A.

The formalization of the protocol messages The message with the asterisk, *, denotes that the entity who received this message did not make and send this message

in the previous stage of the protocol. In the message notation, the symbol $\lhd$ means the entity receives the message.

Message 1 :	$T \lhd *Request$
Message 2 :	$V \lhd *T_n$, $*PIN$, $*\{N_T\}_{Sn1}$
Message 3 :	$T \lhd *\{N_T\}_{Sn2}$, $*\{N_V\}_{Sn3}$
Message 4 :	$V \lhd *\{N_V\}_{Sn4}$, $*\{ID\}_{(NV, NT)}$
Message 5 :	$T \lhd *\Delta\delta$

The initial assumptions of the proposed protocol This section specifies the initial possessions and abilities of the each participant of the protocol. The message with symbol # denotes freshness of the message and the message with symbol $\varnothing$ denotes it can be recognized by the entity who receives it. The message with symbol $\ni$ denotes that the entity, left hand side of the symbol, possess the formula, right hand side of the symbol. The message with the arrow symbol denotes secret values described above the arrow are believed to suitable values between two entities.

$$T \ni N_T \qquad T \ni S_{n1}, S_{n2}, S_{n3}, S_{n4} \qquad T \models \#\, N_T \qquad T \models \varnothing\, N_V$$
$$V \ni N_V \qquad V \ni S_{n1}, S_{n2}, S_{n3}, S_{n4} \qquad V \models \#\, N_V \qquad V \models \varnothing\, N_T$$
$$T \models T \xleftrightarrow{S_{n1}, S_{n2}, S_{n3}, S_{n4}} V \qquad V \models V \xleftrightarrow{S_{n1}, S_{n2}, S_{n3}, S_{n4}} T$$

The goals of the proposed protocol The goals of the proposed protocol are belief , $\models$, and the freshness, #, of the messages between the verifier and the tag. The belief denotes that the message is delivered from right trustable party. And the freshness denotes that the message value is not used in previous protocol session. Satisfying the freshness is important because it can make the communication party assure that the received message was not used in reply attack.

$T \models V \mid\sim N_V$	Tag believes verifier sent N_V
$T \models V \mid\sim N_T$	Tag believes verifier returned N_T
$V \models T \mid\sim N_T$	Verifier believes tag sent N_T
$V \models T \mid\sim N_V$	Verifier believes tag returned N_V
$V \models \# E_{(NV, NT)}$ (ID)	Verifier believes this message is fresh
$V \models T \mid\sim E_{(NV, NT)}$ (ID)	Verifier believes tag sent this message

The first through the fourth goal and the sixth goal related to *the data authenticity*. These goals indicate that the received messages are sent from honest agent, so these messages are trustable. The sixth goal is related to *the data anonymity*. This goal indicates that the received message is fresh, so this message is guaranteed that it was not used in previous session. If these goals are proved logically by applying the logical postulates in GNY logic, then we could assure that this protocol satisfies *the data authenticity* and *the data anonymity*.

The application of the logical postulates The security verification of our protocol will be done by goals deduction with help of the logical postulates provided by the

GNY logic. There are several postulates in GNY logic and we wrote the name of the postulates in the right-side of the messages which are used in the deduction. The first and the fifth message are removed from the following steps because message 1 and 5 are just delivering the data which are not related with the goal deduction.

Message 2 : $V \lhd *T_n , *PIN, *\{N_T\}_{Sn1}$
$V \lhd T_n , PIN, \{N_T\}_{Sn1}$ /* by T1
$V \ni T_n , PIN, \{N_T\}_{Sn1}$ /* by P1
$V \models T \sim N_T$ /*by initial assumption & I1'
The third goal is successfully deduced

Message 3 : $T \lhd *\{N_T\}_{Sn2}, *\{N_V\}_{Sn3}$
$T \lhd \{N_T\}_{Sn2}, \{N_V\}_{Sn3}$ /* by T1
$T \ni \{N_T\}_{Sn2}, \{N_V\}_{Sn3}$ /* by P1
$T \models V \sim N_V$ /*by initial assumption & I1'
The first goal is successfully deduced
$T \models V \sim N_T$ /*by initial assumption & I1'
The second goal is successfully deduced

Message 4 : $V \lhd *\{N_V\}_{Sn4}, *\{ID\}_{(NV, NT)}$
$V \lhd \{N_V\}_{Sn4}, \{ID\}_{(NV, NT)}$
$V \ni \{N_V\}_{Sn4}, \{ID\}_{(NV, NT)}$
$V \models T \sim N_V$ /*by initial assumption & I1'
The fourth goal is successfully deduced
$V \models \# \{ID\}_{(NV, NT)}$ /*by initial assumption, F1 & F2
The fifth goal is successfully deduced
$V \models T \sim \{ID\}_{(NV, NT)}$ /*by initial assumption & I1'
The sixth goal is successfully deduced

The Result of the Proof *the data authenticity* : As you can see above, the tag can assures the received nonce N_V is delivered from the verifier. And the verifier also can assure that the received nonce N_T is delivered from the tag. The delivery assurance means that with this protocol *the data authenticity* is achievable. Because the nonce calculation can only be done by authenticable party who has the secret value set, the attacker cannot be authenticated to honest agent.
The data anonymity : the verifier can assure the received message is fresh which means this message is never been used before the current session. By proving the freshness of the message, the protocol can guarantee the replay attack or tag cloning is impossible by any attacker and *the data anonymity* is achievable.

4.2 Evaluation

In this section, we focus on the security module implementation cost for the passive RFID tag. The passive RFID tag is hardware constrained device so that the

implementation of the complex encryption schemes such as public key encryption or the symmetric key encryption is currently very rough task. Although the complex encryption scheme equipped tag could be implemented, the tag would cost more than 5 cent. Therefore, the implementation cost should be considered very carefully before implementing the security module into the tag. According to [2], possible fabrication amount of gates for a tag within 5 cent is about 10,000 ~ 35,000 gates. Excluding the basic need for RFID tag fabrication such as antenna, IC and memory area, only 1,000 ~ 3,500 gates can be assigned for security module implementation.

To verify whether our scheme can be implemented practically in the tag or not, we made experiment on the total number of gates for our scheme with ASIC implementation.

Table 1. Total number of gates for our scheme implementation.

Bit length of data in padding	*32*	*64*	*128*
# of gates for XOR module	567	850	1,700
# of gates for register	928	1,875	3,713
Total gates	1,495	2,707	5,413

We have designed that the data and pseudonym is padded in parallel. Therefore, 128 XOR modules is needed, and the register which stores the 128 bit-length temporal data for padding such as the secret values, nonce or ID of a tag is also needed for 128 bit-length data padding. However, we can reduce these basic needs by reducing the bit-length of data which the padding module takes for input. For example, if we design the padding module which takes 64 bit-length data as a input then the number of XOR module for the data padding and register size for the temporal input/output data storage can be reduced almost by half. Table 1 shows the estimated total gates for our scheme by using ASIC implementation. The first row in table 1 denotes the bit-length of the input data of the one-time pad module. The fourth row denotes the estimated total number of gates for padding module based on bit-length of the input data.

Table 2. The estimated implementation gates for security modules[10].

Categories	*Types*	*Gates*
Hash Function	SHA-256	10,800
	SHA-1	8,120
	MD5	8,400
	MD4	7,350
Symmetric Enc.	AES	25,000
	Modified AES	3,595
One-time pad(our work)	XOR	1,495

In [10], Feldhofer described that the hash functions and AES based symmetric key encryption algorithms exceed at least 7,000 gates for implementation. Specifically, the implementation of the Hash function cost around 7,000 ~ 11,000 gates and the AES scheme would require 25,000 gates. Therefore, complex encryption scheme and

even hash functions are not suitable in security protocol design for the RFID system because they are not satisfying the least gate limitation of the security module implementation. On the other hand, the one-time pad scheme can be implemented within 1,495 gates if we assumed that the padding module is designed to take 32 bit-length data input for padding. The total gates for our scheme are even smaller than that of the modified AES module which is proposed by Feldhofer. We have compared the total gates of the different encryption module for implementation by referring the Feldhofer's work[10] in table 2.

5 Conclusion

In designing the security protocol for RFID system to resolve privacy problems, the security requirements should be satisfied and also the implementation cost requirements as well. In this paper, we have described the previous works such as hash function based scheme and simple arithmetic calculation based scheme that failed to satisfy all the security requirements. And then, we proposed a light-weight security protocol for RFID system based on one-time pad scheme which satisfies all the security requirements and low-cost implementation requirement. The security requirements satisfaction of our protocol was presented in this paper: the data secrecy was proved by Shannon[8], and the data authenticity and the data anonymity was proved by using GNY logic. Moreover, we have showed that our protocol can be implemented with lower cost than the previous works by comparing the gates for the security module implementation.

References

1. Ari Juels, Ronald Rivest, and Michael Szydlo.: The blocker tag: Selective blocking of RFID tags for consumer privacy. In: Vijay Atluri (ed.): *Conference on Computer and Communications Security -ACM CCS*. ACM Press. Washington DC, USA (2003) 103-111
2. Stephen Weis.: Security and privacy in radio-frequency identification devices. Master thesis, Massachusetts Institute of Technology (MIT). Massachusetts, USA (2003)
3. Dirk Henrici and Paul Muller.: Hash-based enhancement of location privacy for radio frequency identification devices using varying identifiers. In: Ravi Sandhu and Roshan Thomas (eds.): *International Workshop on Pervasive Computing and Communication Security - PerSec 2004*. IEEE Computer Society. Orlando, Florida, USA (2004) 149-153
4. Miyako Ohkubo, Koutarou Suzuki, and Shingo inoshita.: Efficient hash-chain based RFID privacy protection scheme. In *International Conference on Ubiquitous Computing - Ubicomp'04*. Nottingham, England (2004)
5. Ari Juels.: Minimalist cryptography for low-cost RFID tags. In: Carlo Blundo and Stelvio Cimato (eds.): *International Conference on Security in Communication Networks - SCN 2004*. LNCS vol. 3352, Springer-Verlag. Amalfi, Italia (2004) 149-164
6. Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer.: Strong authentication for RFID systems using the AES algorithm. *Workshop on Cryptographic Hardware and Embedded Systems 2004*. LNCS Vol 3156. IACR, Springer-Verlag (2004) 357-370
7. Ari Juels and Stephen Weis: Authenticating pervasive devices with human protocols. In: Victor Shoup (ed.): *Advances in Cryptology - CRYPTO 2005*. LNCS vol.3126. IACR,

Springer-Verlag. Santa Barbara, California, USA (2005) 293-308
8. C.E. Shannon: A Mathematical Theory of Communication. Bell System Technical Journal vol. 27. (1948) 379-423, 623-656
9. L. Gong, R. Needham, and R. Yahalom.: Reasoning about Belief in Cryptographic Protocols. IEEE (1990)
10. Martin Feldhofer and Christian Rechberger.: A case against currently used hash functions in RFID protocols. Printed handout of Workshop on RFID Security *200*6 (2006)
11. Junichiro Saito, Jae-Cheol Ryou, and Kouichi Sakurai.: Enhancing privacy of universal re-encryption scheme for RFID tags. In: Laurence Jang, Minyi Guo, Guang Gao, and Niraj Jha, (eds.): *Embedded and UbiquitousComputing - EUC 2004*, Lecture Notes in Computer Science Vol 3207. Springer-Verlag. Aizu-Wakamatsu City, Japan (2004) 879-890
12. Hopper, N. and Blum, M.: A Secure Human-Computer Authentication Scheme. Technical Report, CMU-CS-00-139, Carnegie Mellon University (2000)
13. Selwyn Piramuthu.: HB and related light-weight authentication protocols for secure RFID tag/verifier authentication. In Collaborative Electronic Commerce Technology and Research 2006 (2006)
14. Jonathan Katz and Adam Smith.: Analyzing the HB and HB+ protocols in the "large error" case. Cryptology ePrint Archive, Report 2006/326(2006)
15. Gavin Lowe.: Breaking and Fixing the Needham Schroeder Public Key Protocol using FDR. In Tools and Algorithms for the Construction and Analysis of Systems. LNCS vol. 1055. Springer-Verlag (1996) 147-166
16. Tom Coffey, Reiner Dojen and Tomas Flanagan.: Formal verification : an imperative step in the design of security protocols. Elsevier(2003)
17. A Marithuria, R. Safavi-Naini, P. Nickolas.: Some remarks on the logic of Gong, Needham and Yahalom. Proc. of the International Computer Symposium, ROC vol.1. Hsinchu, Taiwan (1994) 303-308
18. Istvan Vajda and Levente Buttyan.: Lightweight authentication protocols for low-cost RFID tags. In *Second Workshop on Security in Ubiquitous Computing - Ubicomp 2003*. Seattle, WA, USA (2003)
19. Benessa Defend, Kevin Fu, and Ari Juels.: Cryptanalysis of two lightweight RFID authentication schemes. In *International Workshop on Pervasive Computing and communication Security - PerSec 2007*. IEEE Computer Society Press. New York, USA (2007)