Privacy Guaranteed Mutual Authentication on EPCglobal Class 1 Gen 2 Scheme

Jiahao Wang[1, 3], Terry Ye[2], Edward C. Wong[3]
[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, China, wangjh@uestc.edu.cn
[2]Hong Kong R&D Centre for Logistics and Supply Chain Management Enabling Technologies, Hong Kong
[3] E-Business Technology Institute, the University of Hong Kong, Hong Kong

Abstract

Concerning the security weakness of EPC scheme especially on privacy concerned applications, an Anonymous Mutual Authentication Protocol is proposed for Light-weight security inauguration on Class 1 Gen 2 UHF RFID (EPC C1G2) scheme. By utilizing the existing functions and memory bank of tag, we amend the processing sequence based on current EPC architecture. And an auto-updating index number IDS is enrolled to provide privacy protection to EPC code. A light weight encryption algorithm utilizing tag's existing PRNG and keys are introduced for mutual authentication. Several attacks to the RFID solutions can be effectively resolved through our improvement.

Key words: *RFID, light weight encryption, mutual authentication, privacy*

1. Introduction

Radio Frequency Identification (RFID) is a technology capable of providing wireless identification of objects. This real time and wireless means of information exchanges between tags and readers enables emerging innovative applications in many areas, such as logistics, supply chain management, manufacturing, warehouse managements, et al. As the price of RFID tag is already quite cheap now, a standard EPC Class 1 Gen 2 UHF RFID tag be between 0.05and 0.1 € to be considered affordable[1] . But the low cost demand for a RFID tag also restricts its capability in some aspects, such as privacy and authentication. A powerful malicious reader can illegally snoop, corrupt or manipulate upon the tags if within acceptable communication range. Similarly, tracking of people would also become possible. These potential risks scare away potential adoption as was the case with the boycott of Benetton where the garment maker was forced to take off RFID tags from their clothes. And a scan of tags attached on products inside a container, warehouse, etc, may also lead to corporate espionage. In the medical systems, any snoop and temper of the medical card information can cause even more serious problem.

Although research literatures in RFID security already quite extensive and growing, most of them can not be easily applied into off-the-shelf tags. Among these researches, authentication and privacy perhaps the major focus in security aspect. Some current RFID tags employ cryptographic primitives, but they tend to be more expensive than EPC tags. And the Auto-ID Lab, the research arm of EPCglobal, also operates a special interest group try to proposed uses of EPC to combat counterfeiting of consumer items[2]. They review extensions to existing EPC architecture for security applications. Instead of incorporate cryptography into EPC C1G2 tags, they propose support for future, higher class EPC standards

Following the same thread, we propose a novel anonymous mutual authentication protocol to increase tag privacy protection and authentication functions while remaining compliant with the current EPC C1G2 Standard architecture in this article. Based on utilizing the already been computation unit and memory in EPC tag, we try to implement security functions to the current scheme while minimize the amendments to tag's hardware. This reservation is important to guarantee our improvement can be easily applied into real application. An index-pseudonym (IDS) is used to replace EPC code during inventory process to prove privacy protection. And a light weight symmetric encryption algorithms is implemented for Tag-Reader Mutual authentication. To realize these functions, processing sequence must be consequently changed.

978-0-7695-3398-8/08 $25.00 © 2008 IEEE
DOI 10.1109/ICYCS.2008.217

Our protocol is aimed to be an alternative to the creation of Class 2 EPC standard or as its basis.

Organization of this paper is as follows. In Section II, a literature review is provided. The security treats of EPC C1G2 is reviewed in Section III. In section IV, our protocol scheme and encryption algorithm are introduced. Section V particularly analysis the security performance of our proposal. And section VI analysis implementation characters. Section VII will conclude this paper.

2. Related work

In 2005, the Version 1.1.0 of EPC C1G2 standard was ratified both by EPCglobal and ISO, which harmonized the last version with the ISO 18000-6 Type C amendment[3]. In contrast to established HF RFID standards like ISO 14443 and ISO 15693 where security protocols have already been deployed, the widely applied EPC C1G2 tag only provides an Access and a Kill password (APwd and KPwd) to protect information stored in tags. And as the EPC C1G2 tags can practice outstanding far-field performance, with a communication range of up to 10 meters, it is not difficult to perform a Man-in-the-middle attack from powerful malicious readers. Anybody possessing a reader could read any passersby's tags, which can potentially reveal even the most private information of them. To the weakness on security of the standard, a lot of researches have been carried out in the past several years. The low cost demand for RFID tags forces them to be very resource limited. Typically, they can only store 5-10K logic gates. Within this gate counting, only between 250 and 3000 bits can be devoted to security functions. Some researches try to employ primitive cryptographic into RFID tags, including hash, symmetric or asymmetric based encryption algorithms. But these tags tend to be more expensive than EPC tags currently, and can only suitable for niche and high value product applications.

To investigate the extremely lightweight security protocols, article [4] summarized a set of XOR based authentication protocols. In [5], Juels proposes a solution based on the use of pseudonyms, without using any hash function. LMAP and M^2AP provide two light weight protocols based on the use of pseudonyms and XOR operations[1, 6]. The index-pseudonym refers to a table in which all the information about a tag is stored. Each tag has an associated key which be divided into four 96 bits parts. But these schemes are not sophisticated enough, Li and Wang analyzed some weakness of LMAP and M^2AP and try to break them through two active attacks[7]. The first one is named De-synchronization attack which can break the communication between the tag and the reader. The second is a man-in-the-middle attack called Full-disclosure attack, which can get the whole secret key of the tag. They give out solutions with 40% increase consumption of tag's memory, but which can very likely lead to DOS attack to tags. Article [6] also give out an extension version LMAP+ to countermeasure the weaknesses. But unfortunately, the problems are not well solved as they announced. By calculating the least significant bits of every key and secret, Mihaly etc show that LMAP can be easily broken through a few rounds of eavesdropping[8]. From application perspective, article [9] provides another light weight tag-reader mutual authentication scheme complying to EPC standards. However, this paper doesn't consider privacy and vulnerable under the above attacks.

3. Security threats

The EPC C1G2 standard can be considered as specification for low-cost RFID tags on off-the-shelf applications. Even this standard already be considered a great success after having been adopted by many RFID manufacturers, the quite simple security mechanism of EPC C1G2 constitutes an important pitfall. Except the problems mentioned in the former research, there are three major threats we try to resolve in this work.

Threat 1: Trace and Tracking: The tag's privacy is not considered in Class 1 Gen 2 standard, which can cause seriously problem to customers. As the RF signal usually transmit through open air media, and up to 10 meters for EPC C1G2 tag, it will be easy for an attacker to obtain the EPC code of a tag by simply eavesdrops the air channel. Shield external RF signals/noise physically (i.e. Faraday cage) is not applicable in many real application environments.

Threat 2: Malicious RFID Readers: Products labeled with tags reveal sensitive information when queried by readers, and they do it indiscriminately. Therefore, a powerful malicious reader can illegally snoop, corrupt or manipulate upon tags. For instance, a disgruntled or compromised employee with such readers can simply initiate Man-in-the-Middle Attack to eavesdrop and impersonate those random numbers and one-time-pads in the communication processing. Then, the attacker will be able to decode the cipher texts from the reader by performing the same operations as the tag.

Threat 3: RFID Tag Cloning: The EPC C1G2 standard provide solutions for tag to authenticate readers by examining the shared passwords between them. But there is no authentication to the tag from the reader side. This concision for the protocol leaves drawbacks in application. Any people know the data (e.g., EPC number) structure can probably generate fake tags and attached to counterfeit products. This threat can only be resolved through authentication methods. Even tags giving out genuine EPC numbers, they must still be authenticated by the reader.

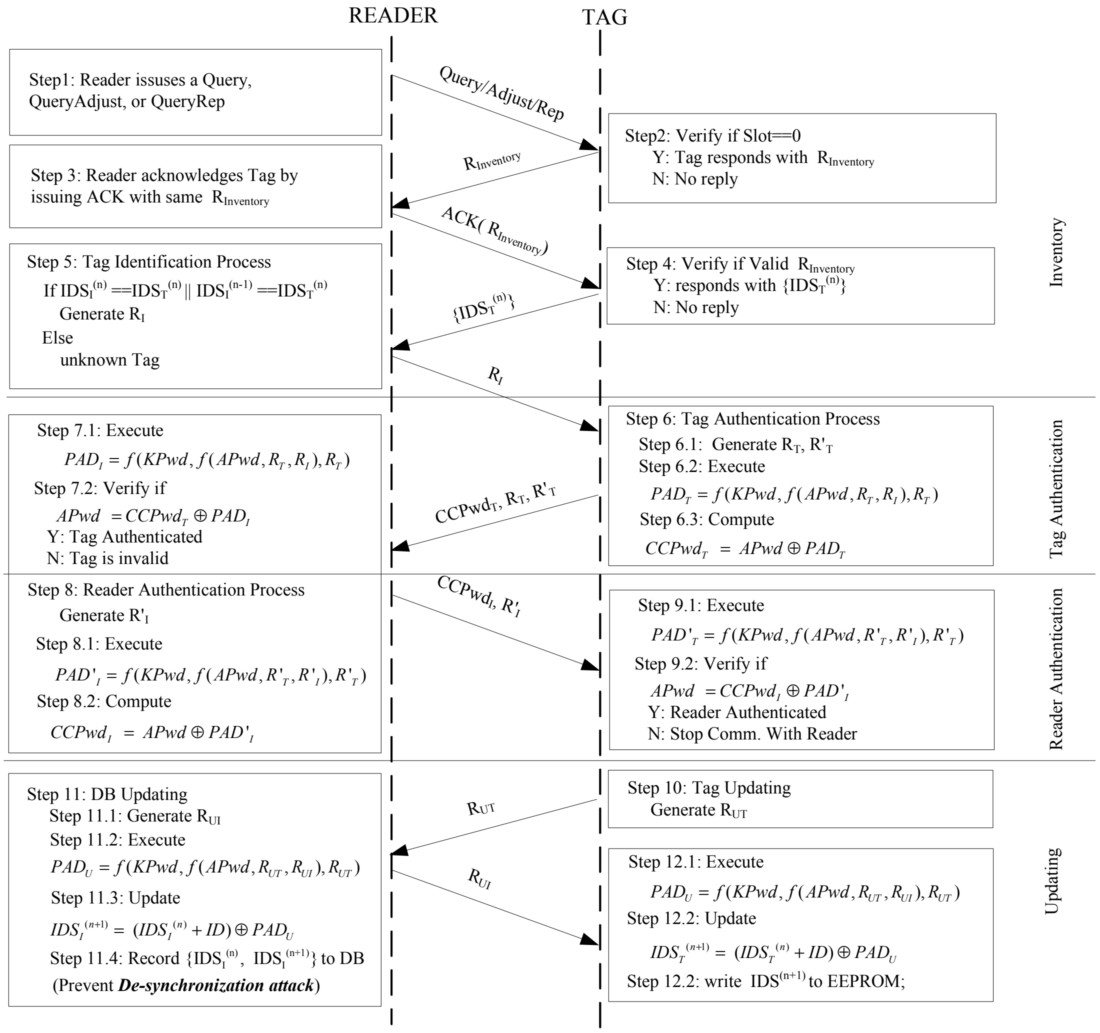
READER
TAG
Step1: Reader issuses a Query, QueryAdjust, or QueryRep
Query/Adjust/Rep
Step2: Verify if Slot==0
Y: Tag responds with $R_{Inventory}$
N: No reply
$R_{Inventory}$
Step 3: Reader acknowledges Tag by issuing ACK with same $R_{Inventory}$
ACK($R_{Inventory}$)
Step 4: Verify if Valid $R_{Inventory}$
Y: responds with $\{IDS_T^{(n)}\}$
N: No reply
$\{IDS_T^{(n)}\}$
Step 5: Tag Identification Process
If $IDS_I^{(n)} == IDS_T^{(n)} \| IDS_I^{(n-1)} == IDS_T^{(n)}$
Generate R_I
Else
unknown Tag
R_I
Inventory
Step 6: Tag Authentication Process
Step 6.1: Generate R_T, R'_T
Step 6.2: Execute
$PAD_T = f(KPwd, f(APwd, R_T, R_I), R_T)$
Step 6.3: Compute
$CCPwd_T = APwd \oplus PAD_T$
$CCPwd_T$, R_T, R'_T
Step 7.1: Execute
$PAD_I = f(KPwd, f(APwd, R_T, R_I), R_T)$
Step 7.2: Verify if
$APwd = CCPwd_T \oplus PAD_I$
Y: Tag Authenticated
N: Tag is invalid
Tag Authentication
Step 8: Reader Authentication Process
Generate R'_I
Step 8.1: Execute
$PAD'_I = f(KPwd, f(APwd, R'_T, R'_I), R'_T)$
Step 8.2: Compute
$CCPwd_I = APwd \oplus PAD'_I$
$CCPwd_I$, R'_I
Step 9.1: Execute
$PAD'_T = f(KPwd, f(APwd, R'_T, R'_I), R'_T)$
Step 9.2: Verify if
$APwd = CCPwd_I \oplus PAD'_I$
Y: Reader Authenticated
N: Stop Comm. With Reader
Reader Authentication
Step 10: Tag Updating
Generate R_{UT}
R_{UT}
Step 11: DB Updating
Step 11.1: Generate R_{UI}
Step 11.2: Execute
$PAD_U = f(KPwd, f(APwd, R_{UT}, R_{UI}), R_{UT})$
Step 11.3: Update
$IDS_I^{(n+1)} = (IDS_I^{(n)} + ID) \oplus PAD_U$
Step 11.4: Record $\{IDS_I^{(n)}, IDS_I^{(n+1)}\}$ to DB
(Prevent ***De-synchronization attack***)
R_{UI}
Step 12.1: Execute
$PAD_U = f(KPwd, f(APwd, R_{UT}, R_{UI}), R_{UT})$
Step 12.2: Update
$IDS_T^{(n+1)} = (IDS_T^{(n)} + ID) \oplus PAD_U$
Step 12.2: write $IDS^{(n+1)}$ to EEPROM;
Updating

Figure 1. Anonymous Mutual Authentication Protocol

For the above reasons, tag's PIN or EPC code mast be masked and transmitted through secure channel to solve these problems.

4. Anonymous Mutual Authentication Protocol

Fortunately, EPC schemes leave spaces for optimization. To resolve the security weakness mentioned in the last section, a lot of researches have been carried out recently accompanied with the booming implementation of RFID in various applications. Based on our prior work[10], Here we propose a security intensified solution rely on the already been capabilities of EPC C1G2 tag. To facilitate the description, table 1 list some notations we used in this paper.

Table 1. Notations

Notation	*Descriptions*
$R_{Inventory}$	16bit Random No. used for singulate a tag
R_T,R'_T,R_{UT}	16bit Random No. Generated by Tag
R_I,R'_I, R_{UI}	16bit Random No. Generated by Reader
IDS	16bit Index Pseudonym Random No.
$APwd_M$/$KPwd_M$	16 MSBs of APwd/KPwd
$APwd_L$/$KPwd_L$	16 LSBs of APwd/KPwd
n	The serial number of current round

4.1. Protocol description

Here we modify the processing sequences of EPC scheme to implement privacy and mutual-authentication functions. We assume that both the backward and the forward channels can be eavesdropped by an attacker, despite their asymmetry. A light weight encryption processes are added to tag. And the whole processing sequence can be split into four main stages, named inventory, tag authentication, reader authentication and updating phases. Figure 1 particularly describes the processing sequences.

Steps 1-5 details tag inventory process. Among which, steps 1-3 are exactly same as EPC C1G2 standard. The communication must be initiated by readers due to the fact that low cost tags are passive. If the query received by a Tag within right Slot, it will generate a 16 bits random number $R_{Inventory}$ through its PRNG and reply to the reader. Then, the $R_{Inventory}$ be used in ACK by the reader. After receive corresponding ACK in step 4, the selected tag will send its $IDS_T^{(n)}$ to the reader instead of PC or EPC code. We use $IDS_T^{(n)}$ and $IDS_I^{(n)}$ represent the index send from tag and reader respectively. They may be de-synchronized under attack and they are updated after each successive conversation to guarantee tag's privacy. In step 5, reader will scan index rows $IDS_I^{(n)}$ and $IDS_I^{(n-1)}$ from database for corresponding $IDS_T^{(n)}$. Normally, a record including all the necessary information about the tag can be found, including EPC code, passwords and et al. Otherwise, the tag is unrecognizable to the system.

Steps 6-9 detail the mutual authentication process which composed by two message exchanging operations. During the tag authentication step, an encrypted pad $CCPwd_T$ is transmitted from tag to reader by using APwd as the shared key. It can be authenticated if the reader side can successively find the same keys to validate the pad. The reader can also be authenticated with the same process in steps 8-9 except using different random numbers R'_T and R'_I. The encryption algorithm will be explained in the next section.

Steps 10-12 describe the updating process. The IDS shall be updated in a secure form for protecting the tag's privacy. A new pare of random numbers R_{UI} and R_{UT} separately generated in both side be used to generate a secret pad PAD_U Accompanied with the last $IDS_I^{(n)}$ and ID, a new $IDS^{(n+1)}$ can be generate to prevent De-synchronization attack in below way.

$$IDS^{(n+1)} = (IDS^{(n)} + ID) \oplus PAD_U$$

Both $IDS_I^{(n)}$, $IDS_I^{(n+1)}$ will be store in database, while only $IDS_I^{(n+1)}$ should be write to tag's EEPROM for next conversation.

4.2. Encryption algorithm

Based on the algorithms in article [9], here we implement an extreme light weight encryption algorithm in the mutual authentication process. As APwd = $APwd_M$ || $APwd_L$ and KPwd = $KPwd_M$ || $KPwd_L$, a function f is used for encryption with 16 bits calculate capability, and two 16 bits pads (PAD_L and PAD_M) are generated by utilizing the Access and Kill passwords as $PAD = f(KPwd, f(APwd, R_T, R_I), R_T)$. Since only the tag and the reader pare storing the same passwords can generate the same pads, they can be verified by exchanging two pairs of random numbers R_T and R_I. The encryption processes are show as follows.

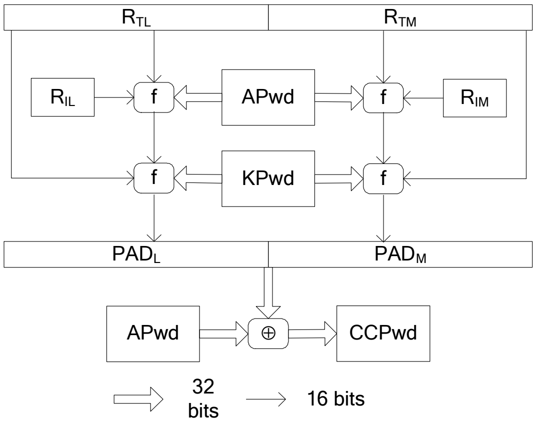

Figure 2. Encryption Algorithm

To explain the process in figure 2, we can represent the 16 bits random number x, y as:

$$x = ht_1 ht_2 ht_3 ht_4 \ , \qquad ht \in \{0,1,2,\cdots,9,A,B,\cdots,F\}$$

$$y = hm_1 hm_2 hm_3 hm_4 \ , \quad hm \in \{0,1,2,\cdots,9,A,B,\cdots,F\}$$

Let us use *K* represent the 32 bits APwd or KPwd, such as :

$$K = a_0 a_1 a_2 \ldots\ldots a_{29} a_{30} a_{31}$$

Consequently, we can calculate *f* as:

$$f(K,x,y)$$

$$= a_{ht1} a_{ht2} a_{ht3} a_{ht4} || a_{ht1+16} a_{ht2+16} a_{ht3+16} a_{ht4+16} ||$$

$$a_{hm1} a_{hm2} a_{ht3} a_{hm4} || \ a_{hm1+16} a_{hm2+16} a_{hm3+16} a_{hm4+16} \quad \text{[Base 2]}$$

$$= \ hv_1 hv_2 hv_3 hv_4 \qquad \text{[Base 16, where } hv \in \{0,1,2,\cdots,9,A,B,\cdots,F\} \text{]}$$

After the encryption process, the overall 32 bits PAD execute XOR with the APwd to encrypt or decrypt the cover-code chunk CCPwd through the equation:

$$CCPwd = APwd \oplus PAD \ .$$

5. Security analysis

Considering the EPC C1G2 standard, the range of the communication channel between a particular tag and a reader may be up to 10 meters. An active attacker may continuously eavesdrop on the signals. Based on the fact that the APwd and KPwd are only 32 bits each in the EPC standard, a simple brute-force attack or other active attacks on the tag, can crack them. Our proposal is not aims to establish a fully secured operation in RFID applications, but we try to develop a scheme simple, low cost, and adhering to EPC standards. We tried to solve the security services for Tag-Reader mutual authentication and privacy correlate problems through a light weight and practically scheme aspect.

To protect the privacy of a tag, our protocol involves a tag identity updating phases after each successful conversation. Unlike those hash and asymmetric key based solutions, our protocol implements an extremely light weight scheme based only on bitwise operations. Our protocol takes use of a 16 bits index-pseudonym (IDS) to provide privacy protection. The IDS also be used as the index of a table where all the information about a tag is stored including secret keys. By implementing these functions, we can at least solve the following problems in the current system.

(1) Privacy

In addition to the original standard, we increase the privacy protection at the very beginning of each communication sessions. Instead of sending its ID through open channel, a tag will answer reader's query by replying its current indx-pseudonym, $IDS^{(n)}$, and an eavesdropper can only get random wraps in this process. To prevent the possible violation of the location privacy of a tag owner, $IDS^{(n)}$ will be updated after each successful communication session, and $IDS^{(n+1)}$ will be calculated separately in both side. As the updated $IDS^{(n+1)}$ do not appear on the insecure open channel during updating process, it makes difficulties for an attacker to identify and track the tag. The tag ID and corresponding user confidentiality can be kept secure to guarantee users privacy maximally. Further more, by update the $IDS^{(n)}$ after the mutual authentication, a future security compromise on an RFID tag will not reveal data previously transmitted and forward Security can be guaranteed.

But before a successive updating of IDS, there still has possibility that the attacker can distinct and follow a user for some while. For instance, an attacker may sends hello messages to the tag to receives the respond $IDS^{(n)}$, then he stops the authentication process. As the tag may have chance lose contact with legitimate readers and do not updated its $IDS^{(n)}$ for a certain period, the attacker can repeats the above action and get the same $IDS^{(n)}$ in this interval.

(2) Mutual Authentication

By authenticating in step 7, a fake tag or a malicious tag which does not posses the correct keys can be eliminated. We can prevent attackers cheating backend system through cloned tags or tag impersonation attack by this way. And for authenticating the reader to the tag, a validate reader mast have the proper privilege to access the database and distill the required passwords, APwd and KPwd. There have been a lot of techniques to prevent such kind of information leaking in backend systems. And consequently, any malicious readers try to execute Genuine Reader Impersonation Attack to cheat tags can be prevented.

And as unique random numbers, $R_{I,}$ R'_I and R_T, R'_T are used in each communication sessions, we can finely defend replay attacks in our protocol. And by enrolling mutual authentication and encryption functions to the EPC scheme, we can fend off many threats like malicious snooping readers, disgruntled employee, Cloned Tag, man-in-the-middle attacks, et al.

(3) Prevent *De-synchronization Attack*

To protect an RFID tag's privacy, our protocol involved a tag identity update process after each successive protocol round. The IDS take over all the identification functions of the original tag's ID (EPC or PC) in the first part of privacy protected protocols and the keys values are depend on which IDS can be found in step 5. So the synchronization of secret information between the database and tags are very important to guarantee their following protocol runs.

These kinds of attacks always try to break the synchronization at both sides or leave the protocol incomplete. As described in article [7], an active attacker can initiate a man-in-the-middle attack to leave the update process wrong in tag side by change the random numbers R_{UT} or R_{UI}, or just simply block off the update messages from reader to tag side. For example, R_{UT} is generated by tag's PRNG, reader side may be fraud if R_{UT} have been tempered, and finally led to different updating results in both sides. And furthermore, this attack has high probability to cause DOS attack to tags by preventing them from successive updating.

Here we store all the current and newly generated IDS to database to insure even the tag did not update its $IDS_T^{(n)}$ in the last round still can be recognizable. Our protocol can solve the attack with no extra addition storing in tag's EEPROM.

6. Implementation analysis

Considering EPC C1G2 tags are very computationally constrained devices, we only take use of the existing functions in the tag. Here we only use bitwise operations. As they have already been implemented in the existing EPC C1G2 tags on off-the-shelf products, there will be no extra gate counting needed. To implement our protocol, we need to redesign the processing sequence on both the tag and the reader side.

For the memory storage, we consider its use as an input/output medium capable of interfacing with a set of crypto operation within the tag. And we also try to take use of the existing memories of EPC C1G2 tag to avoid extra storage costs. Our solution utilizes the Reserved memory bank in EPC C1G2 tag where containing two 32 bits passwords, APwd and KPwd. The kill and access passwords are individually lockable, as EPC, TID, and User memory. These memory banks are always readable regardless of their lock status. The reserved memory mast be read/write unlocked to provide updating capability in our protocol. The encryption algorithm is based on 16 bits operation, which divides these memories into 4 pairs, $APwd_M$, $APwd_L$, $KPwd_M$ and $KPwd_L$. We also use the 16 bit PC from EPC memory to act as ID in our protocol. Here we take use of the existing keys to avoid extra hardware amendment and achieve trade-off between security and applicability. Our proposed scheme can still be applicable and more strengthened, if the length of APwd and KPwd be extended in active tags or enhanced tags used for high value items.

Besides the existing storages, our protocols need to add a 16 bits rewritable memory storage space from User memory for IDS. EPC memory and TID memory are leaving unchanged, which still can be achieved after authentication.

7. Conclusion

To summarize, many former proposals are based on the hypothesis that low-cost tags can not generate random numbers, and they make almost all the computational load fall on the reader side. Based on the latest research achievements, we provide an Anonymous Mutual Authentication Protocol in this paper concerning security attributes of EPC C1G2 standard. To alleviate these flaws, we take advantage of the EPC C1G2 standard and utilized its already existing function PRNG. As the random numbers and keys are all 16 bits, our algorithms are designed accompany to the hardware restriction to insure it can be easily applied to the existing applications. Our solution may be not fully secure but it is simple, cost-effective, and light-weight to be implemented on tag. Through the four phases in our protocol, we can thwart many existing threats such as malicious readers, man-in-the-middle attacks, Cloned Tag, full disclosure, track and tracing, et al. Important related problems, such as implementation performance and security verification, will be addressed in future reports.

Acknowledgment

This research project has been supported by the LSCM R&D Funding Program under Grant No. ITP/028/07LP. The authors would like to thank Professor Joshua Huang and Dr Zongwei Luo for their comments and suggestions.

References

[1] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "M<sup>2</sup>AP: A minimalist mutual-authentication protocol for low-cost RFID tags," presented at Proceedings of Ubiquitous Intelligence and Computing UIC'06, Wuhan, China, 2006.

[2] T. Staake, F. Thiesse, and E. Fleisch, "Extending the EPC network - the potential of RFID in anti-counterfeiting," ACM Symposium on Applied Computing, pp. 1607-1612, 2005.

[3] EPCglobal Ratified Standard. EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz - 960MHz Version 1.0.10, http://www.epcglobalinc.org/standards/.

[4] I. a. Vajda and L. Butty'an, "Lightweight Authentication Protocols for Low-Cost RFID Tags," presented at Second Workshop on Security in Ubiquitous Computing -- Ubicomp 2003, Seattle, WA, USA, 2003.

[5] A. Juels, "Minimalist cryptography for low-cost RFID tags," Lecture Notes in Computer Science, vol. 3352, pp. 149-164, 2005.

[6] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags. ," presented at Proceeding of 2nd Workshop on RFID Security, 2006.

[7] H. Chien and C. Huang, "Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols," ACM SIGOPS Operating Systems Review, vol. 41, pp. 83 - 86, 2007.

[8] M. Barasz, B. Boros, P. Ligeti, K. Loja, and D. Nagy, "Breaking LMAP," presented at RFIDsecurity'07, 2007.

[9] D. M. Konidala, Z. Kim, and K. Kim, "A Simple and Cost-Effective RFID Tag-Reader Mutual Authentication Scheme," presented at Pre-Proc. of International Conference on RFID Security 2007 (RFIDSec 07), Malaga, Spain, 2007.

[10] Z. Luo, T. Chan, J. Li, “A Lightweight Mutual Authentication Protocol for RFID Networks,” ICEBE 2005, pp. 620-625, 2005.