PGMAP: A Privacy Guaranteed Mutual Authentication Protocol Conforming to EPC Class 1 Gen 2 Standards

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

Abstract

To resolve the security vulnerabilities and comply with EPC Class 1 Gen 2 UHF RFID (EPC C1G2) Standard at the same time, we present a Privacy Guaranteed Mutual Authentication Protocol (PGMAP). By utilizing the existing functions and memory bank of tag, we amend the processing sequence based on current EPC architecture. An auto-updating index number IDS is enrolled to provide privacy protection to EPC code and a set of light weight algorithms utilizing tag's PRNG are added for authentication. Several attacks to the existing security solutions can be effectively resolved in our protocol.

1. Introduction

With the booming prosperous of logistics and e-business market, Radio frequency Identification (RFID) technology is showing more and more importance. 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]. The efficiency gains from using RFID tags could substantially lower the cost of tagged items, which enable RFID techniques can be widely applied. The benefit of this technique is obviously. But the low cost demand for a RFID tag also restricts its calculation and storage capability, which lead to weakness in some aspects, such as security. 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 the information stored in tags. A powerful malicious reader can easily snoop, corrupt or manipulate upon the tags if within acceptable communication range (up to 10 meters for EPC C1G2). 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 are 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.

In this article, we propose a PGMAP to increase tag privacy protection and authentication functions while remain complying with the current EPC C1G2 Standard architecture. Based on utilizing the already been computation unit and memory storage 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 environment applications. An index-pseudonym IDS is used to replace EPC code during inventory process to prove privacy protection. And a set of light weight symmetric encryption algorithms are implemented for

This research was supported by LSCM R&D Funding Program under Grant No. ITP/028/07LP

978-0-7695-3395-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ICEBE.2008.16

Tag-Reader Mutual authentication. Several amendments are made to prevent Full Disclosure attack and De-synchronization attack introduced in previous work[3]. To implement these functions, processing sequence must consequently be changed. Our researched 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 threats of EPC C1G2 are reviewed in Section III. In section IV, theory and steps of PGMAP is introduced. Section V particularly analyzes the security attributes of our proposal. And section VI analysis implementation characters. Section VII will conclude this paper.

2. Related work

To the weakness on security of EPC C1G2 standard, a lot of researches have been carried out in the past several years.

2.1. Literature review

As in this heading, they should be Times 11-point boldface, initially capitalized, flush left, with one blank line before, and one after.

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[4]. 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. 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.

As [5] summarized, Privacy and security in RFID can be protected through some physical approaches, including simple RF shielding (e.g., aluminum foil), distance detection, interference with RFID singulation and physical disablement.

In the symmetric encryption scope, researchs mainly concentrate in developing cryptography potentially lightweight enough for inclusion in low-cost devices. A standard implementation of the Advanced Encryption Standard (AES), 20-30K gates are considered to be applicable in some kind of passive tags[1]. Feldhofer et al. have described an AES implementation designed specifically for RFID devices[6]. This implementation requires security resources exceeding those presently possible in EPC tags, but perhaps suitable for some of the enhancements we describe here. It is as yet unclear whether any of these recently proposed primitives are both strong enough and agile enough for use in low-cost RFID tags, but they represent an important continuing area of inquiry.

Most of the proposed solutions are based on the use of hash functions with a reduced number of gates, but although this proposal seems to be light enough to used in a low-cost RFID tag, the security of this hash scheme remains an open question. Even some prototype successfully taken used the traditional hash functions (MD5, SHA-1, SHA-2), the increased cost will out burden many applications in real environment[7]. Peris et al. introduced several topical hash based schemes, including hash lock, randomized hash lock and hash chain[8]. Articles [9, 10] improved them and provide binary tree based hash-chain. Hash based solutions can be used to solve the tracking and tracing problems in RFID to protect the privacy of tag side. But as these solutions still require extra hash function or memory storage, they still not suitable to be applied into the widely used EPC C1G2 tags.

To investigate the extremely lightweight security protocols, article [11] summarized a set of XOR based authentication protocols. In [12], Juels proposes a solution based on the use of pseudonyms, without using any hash function. LMAP and M2AP provide two light weight protocols based on the use of pseudonyms and XOR operations[1, 13]. 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 M2AP and try to break them through two active attacks[3]. 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 [13] 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[14]. From application perspective, article [15] provides another light weight tag-reader mutual authentication scheme complying to EPC standards. However, this paper doesn't consider privacy and vulnerability under the above attacks.

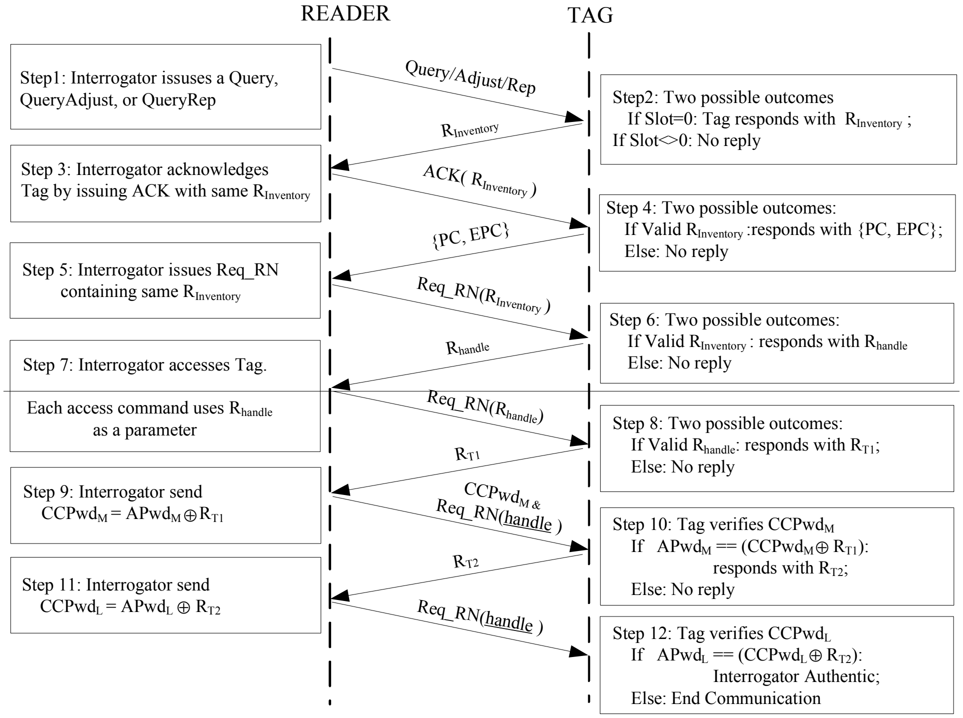
READER
TAG
Step1: Interrogator issuses a Query, QueryAdjust, or QueryRep
Query/Adjust/Rep
Step2: Two possible outcomes
If Slot=0: Tag responds with $R_{Inventory}$;
If Slot<>0: No reply
$R_{Inventory}$
Step 3: Interrogator acknowledges Tag by issuing ACK with same $R_{Inventory}$
ACK($R_{Inventory}$)
Step 4: Two possible outcomes:
If Valid $R_{Inventory}$:responds with {PC, EPC};
Else: No reply
{PC, EPC}
Step 5: Interrogator issues Req_RN containing same $R_{Inventory}$
Req_RN($R_{Inventory}$)
Step 6: Two possible outcomes:
If Valid $R_{Inventory}$: responds with R_{handle}
Else: No reply
R_{handle}
Step 7: Interrogator accesses Tag.
Each access command uses R_{handle} as a parameter
Req_RN(R_{handle})
Step 8: Two possible outcomes:
If Valid R_{handle}: responds with R_{T1};
Else: No reply
R_{T1}
Step 9: Interrogator send $CCPwd_M = APwd_M \oplus R_{T1}$
$CCPwd_M$ & Req_RN(handle)
Step 10: Tag verifies $CCPwd_M$
If $APwd_M == (CCPwd_M \oplus R_{T1})$: responds with R_{T2};
Else: No reply
R_{T2}
Step 11: Interrogator send $CCPwd_L = APwd_L \oplus R_{T2}$
Req_RN(handle)
Step 12: Tag verifies $CCPwd_L$
If $APwd_L == (CCPwd_L \oplus R_{T2})$: Interrogator Authentic;
Else: End Communication

Figure 1. Security of EPC Class 1 Gen 2 UHF RFID Protocol

In 2007, Chien et al. try to solve the mutual authentication problem through taking use of the tag's Cyclic Redundancy Code (CRC) function[16]. But such CRC solutions were proved by Pedro et al. few months later that they are not suitable for solving the existing problems[17].

2.2. Our innovation

The research literature in RFID security is already quite extensive and growing. Based on utilizing existing calculation capability and memory of EPC tag, we focus on protocols to realize privacy protection and mutual authentication on the off-the-shelf products. We consider ways to create RFID tags that perform cryptographic functionality while remaining compliant with both the EPC C1G2 standard and conformance specification.

To sum up, the main contributions of this paper are as follows. (1) In this work, we design a standard compliant protocol to minimize amendments to hardware architecture of tag. This compatibility is important to guarantee our innovation can be easily applied. (2) A set of light weight symmetric encryption algorithms is implement which take use existing functions in EPC tag, such as 16 bits Pseudo-Random Number Generator (PRNG), bitwise XOR (⊕), bitwise OR (∨), bitwise AND (∧), addition mod 2^m (+), APwd and KPwd, et al. (3) Index-pseudonym (IDS) is used to replace EPC code during inventory process to prove privacy protection. (4) Mutual authentication function added based on EPC architecture. (5) Several amendments are made to prevent Full Disclosure attack and De-synchronization attack in previous works.

No cryptographic hash functions/keys are used within tags our protocol. And different with the former research, we contribute to the state of arts of researches on solving security weakness in light weight encryptions by taking use of EPC tag's PRNG. Meanwhile, we try our best on optimization to reduce costs and design complexity. Our solution is more efficiency and applicable to the widely used EPC tag.

3. EPC Class 1 Gen 2 UHF RFID Protocol[4]

The EPC C1G2 standard can be considered as specification for low-cost RFID tags on off-the-shelf applications. Although it represents a great advance for the establishing of RFID technology, the security level of this standard is extremely low. To facilitate the description, table 1 list the notations we used in this paper.

Table 1. Notations

Notation	*Descriptions*
$R_{Inventory}$	16bit Random No. used for singulate a tag
R_{handle}	16bit Random No. used for represent a tag
R_{T}	16bit Random No. Generated by Tag
R_{I}	16bit Random No. Generated by Reader
IDS	16bit Index Pseudonym Random No.
$APwd_{\mathrm{M}}$	16 MSBs of APwd
$APwd_{\mathrm{L}}$	16 LSBs of APwd
n	The serial number of current round
$\|\|$	Concatenates its right operand to the end of its left operand
$\oplus$	Bitwise XOR operation
$\vee$	Bitwise OR operation
$\wedge$	Bitwise AND operation

3.1. Security Assessment of EPC Class1 Gen2 UHF RFID Protocol

From the view of security, the processing sequence of EPC C1G2 protocol mainly including two functions, inventory and access, as divided by the dashed line in the figure 1.

A tag will generate 4 random numbers in the process. The use of Kill password is similar as the above sequences. To guarantee the obscure efficiency, the protocol request a reader shall not use handle or reuse a random number for cover-coding purposes.

From the view of security aspect, the EPC C1G2 tags only support on-chip a 16 bits PRNG , a 16 bits CRC, two 32 bits APwd and KPwd. Besides, the reserved memory can be locked by the manufacturer, to prevent unauthorized read or modification. But tag memory is still susceptible to physical attacks.

3.2. Security treats and requirements

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.

For the above reasons, tag's PIN or EPC code mast be masked and transmitted through secure channel to solve the security problems. Meanwhile, cover code should be transferred through a secure way for obscure the password during communication.

4. Scheme of PGMAP

To resolve the security weakness in EPC C1G2, the standard leave rooms to strengthen for optional commands or class 2 tag. Here based on our prior work [7], we propose a new PGMAP utilizing the already been capabilities on EPC C1G2 tag.

We assume that both the backward and the forward channels can be eavesdropped by an attacker, despite their asymmetry. Some light weight encryption processes are added to tag. And the whole processing sequence can be split into three main stages, named inventory phases, mutual authentication phases and updating phases. Figure 2 particularly describes the processing sequences.

(1) Inventory phases

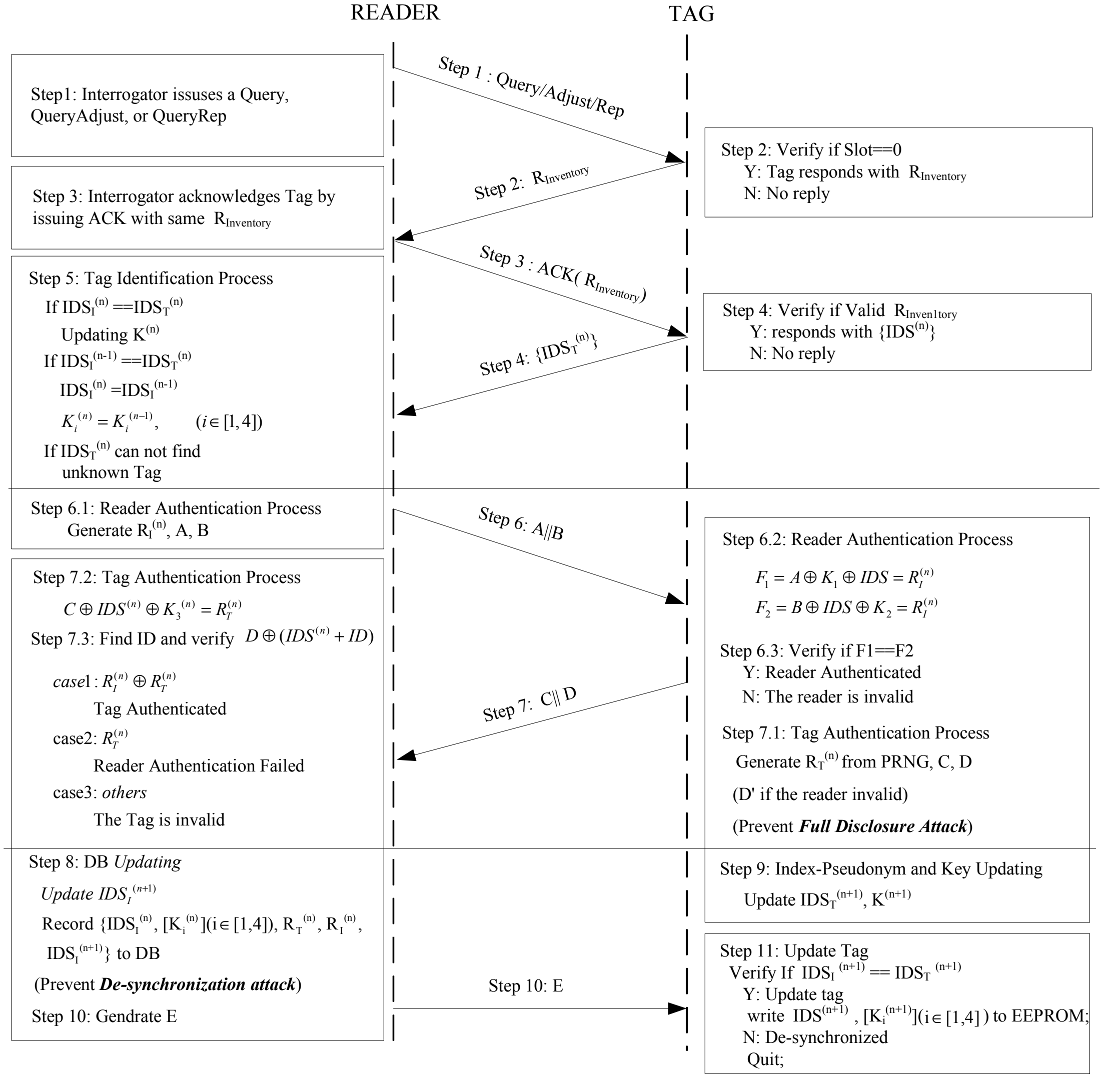
READER
TAG
Step1: Interrogator issuses a Query, QueryAdjust, or QueryRep
Step 1 : Query/Adjust/Rep
Step 2: Verify if Slot==0
Y: Tag responds with $R_{Inventory}$
N: No reply
Step 2: $R_{Inventory}$
Step 3: Interrogator acknowledges Tag by issuing ACK with same $R_{Inventory}$
Step 3 : ACK($R_{Inventory}$)
Step 4: Verify if Valid $R_{Inven1tory}$
Y: responds with $\{IDS^{(n)}\}$
N: No reply
Step 4: $\{IDS_T^{(n)}\}$
Step 5: Tag Identification Process
If $IDS_I^{(n)} == IDS_T^{(n)}$
Updating $K^{(n)}$
If $IDS_I^{(n-1)} == IDS_T^{(n)}$
$IDS_I^{(n)} = IDS_I^{(n-1)}$
$K_i^{(n)} = K_i^{(n-1)}, \quad (i \in [1,4])$
If $IDS_T^{(n)}$ can not find
unknown Tag
Step 6.1: Reader Authentication Process
Generate $R_I^{(n)}$, A, B
Step 6: A||B
Step 6.2: Reader Authentication Process
$F_1 = A \oplus K_1 \oplus IDS = R_I^{(n)}$
$F_2 = B \oplus IDS \oplus K_2 = R_I^{(n)}$
Step 6.3: Verify if F1==F2
Y: Reader Authenticated
N: The reader is invalid
Step 7.1: Tag Authentication Process
Generate $R_T^{(n)}$ from PRNG, C, D
(D' if the reader invalid)
(Prevent ***Full Disclosure Attack***)
Step 7: C|| D
Step 7.2: Tag Authentication Process
$C \oplus IDS^{(n)} \oplus K_3^{(n)} = R_T^{(n)}$
Step 7.3: Find ID and verify $D \oplus (IDS^{(n)} + ID)$
$case1: R_I^{(n)} \oplus R_T^{(n)}$
Tag Authenticated
case2: $R_T^{(n)}$
Reader Authentication Failed
case3: *others*
The Tag is invalid
Step 8: DB *Updating*
Update $IDS_I^{(n+1)}$
Record $\{IDS_I^{(n)}, [K_i^{(n)}](i \in [1,4]), R_T^{(n)}, R_I^{(n)}, IDS_I^{(n+1)}\}$ to DB
(Prevent ***De-synchronization attack***)
Step 10: Gendrate E
Step 10: E
Step 9: Index-Pseudonym and Key Updating
Update $IDS_T^{(n+1)}, K^{(n+1)}$
Step 11: Update Tag
Verify If $IDS_I^{(n+1)} == IDS_T^{(n+1)}$
Y: Update tag
write $IDS^{(n+1)}$, $[K_i^{(n+1)}](i \in [1,4])$ to EEPROM;
N: De-synchronized
Quit;

Figure 2. Proposed Privacy Guaranteed Mutual Authentication Protocol

Steps 1-5 details tag inventory process. Among which, steps 1-3 are exactly same as EPC C1G2 scheme. 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. The index-pseudonym IDS is the index of a table (a row) where all the information about a tag is stored. 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. The record items including $\{IDS^{(n)}, IDS^{(n+1)}, R_T^{(n)}, R_I^{(n)}, [K_i^{(n)}](i \in [1,4])\}$. At the reader side, the keys need to be setup at step 5 in each conversation. Similar to LMAP, we take use the following algorithms for key updating:

$$K_1^{(n)} = K_1^{(n-1)} \oplus R_T^{(n-1)} \oplus (K_3^{(n-1)} + ID) \tag{1}$$

$$K_2^{(n)} = K_2^{(n-1)} \oplus R_T^{(n-1)} \oplus (K_4^{(n-1)} + ID) \tag{2}$$

$$K_3^{(n)} = (K_3^{(n-1)} \oplus R_I^{(n-1)}) + (K_1^{(n-1)} \oplus ID) \tag{3}$$

$$K_4^{(n)} = (K_4^{(n-1)} \oplus R_I^{(n-1)}) + (K_2^{(n-1)} \oplus ID) \tag{4}$$

If no $IDS_I^{(n)}$ match and a $IDS_I^{(n-1)}$ be found, that means something wrong in the last conversation round during updating phases and caused a de-synchronization between the reader and the tag. In this situation, we simply reuse the keys in last version to resynchronize the record in both sides. And further than just defend De-synchronization attack, we can also defend DOS attacks, which may aimed on disable tags and remain unsolved on the previous works. The security evaluation will be given out in the next section.

(2) Mutual authentication phases

Steps 6-7 details Mutual Authentication process. The authentication function is composed by two message exchanging processes. After keys setup, step 6 details tag authentication process which includes 3 parts. In step 6.1, reader generate a new random numbers $R_I^{(n)}$. Accompanied with $K_1^{(n)}$ and $K_2^{(n)}$, it is used to generate messages A and B as:

$$A = IDS^{(n)} \oplus K_1^{(n)} \oplus R_I^{(n)} \quad (5)$$

$$B = IDS^{(n)} \vee K_2^{(n)} + R_I^{(n)} \quad (6)$$

Then, they are combined as the signature of the reader, A||B, and send to tag. If they possessing the same $K_1^{(n)}$ and $K_2^{(n)}$, $R_I^{(n)}$ can be distilled by reverse computation in tag and the reader can be authorized.

In step 7.1, a $R_T^{(n)}$ is generated through tag's PRNG and used in messages C and D (D′ if reader is invalid). As these calculations are carried out on the tag, we utilize the already been calculating units in the tag. The encryption functions are as follows:

$$C = IDS^{(n)} + K_3^{(n)} + R_T^{(n)} \quad (7)$$

$$D = (IDS^{(n)} + ID) \oplus R_T^{(n)} \oplus R_I^{(n)} \quad (8)$$

$$D' = (IDS^{(n)} + ID) \oplus R_T^{(n)} \quad (9)$$

After reader receive C||D, it firstly distills $R_T^{(n)}$ from C at step 7.2. Thereafter, it checks the validity of the tag by calculates $D \oplus (IDS^{(n)} + ID)$. Upon failure, the reader will initiate a re-authenticated process and the record of retry times shall be increase 1 for the tag. This record is used for defend brute force attack. A tag will be temporarily or permanently forbidden if its retry times exceed certain threshold.

(3) Updating phases

After the reader and the tag have been mutually authenticated, the IDS and keys updating processes are carried out in a secure form in steps 8-11 for protecting the tag's privacy. The random numbers $R_I^{(n)}$ and $R_T^{(n)}$ will be used again to prevent Full Disclosure Attack. The different is, only $IDS_I^{(n+1)}$ updated on reader side. And the reader will store all the parameters at current conversation to database, including $IDS_I^{(n+1)}$, $IDS^{(n)}$, $R_T^{(n)}$, $R_I^{(n)}$ and $[K_i^{(n)}](i\in[1,4])$. And the updating calculations of keys are leaved to the next conversation. The updating process of IDS is as:

$$IDS^{(n+1)} = (IDS^{(n)} + (R_T^{(n)} \oplus K_4^{(n)})) \oplus ID \quad (10)$$

And it is embedded in message E to send to tag as updating notice.

$$E = (IDS_I^{(n+1)} + ID) \oplus R_T^{(n)} \oplus R_I^{(n)} \quad (11)$$

To the tag side, after send out he authentication signature C||D at step 7, it directly start calculate the updating messages in step 9, including $IDS_T^{(n+1)}$ and $[K_i^{(n+1)}](i\in[1,4])$. To update it immediately is because a Class 1 tag only has restricted computation capability and powered by backscattering energy sent by the reader device. All the time consuming calculation must be accomplished before it lose power supply from the reader. The updating algorithms are same as those in reader side. Those updated $IDS^{(n+1)}$ and $[K_i^{(n+1)}](i\in[1,4])$ will be write to tag's EEPROM if $IDS_I^{(n+1)}$ equal to $IDS_T^{(n+1)}$. Otherwise, the tag will cease updating to prevent DOS attack.

5. Security analysis

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 XOR (⊕), bitwise OR (∨), bitwise AND (∧), and addition mod 2^m (+) operations. Our protocol also takes use of the index-pseudonym (IDS) to prove privacy protection. The IDS index a record storing four associated keys, K_1, K_2, K_3 and K_4. All these parameters are with the length of 16 bits to accompany EPC C1G2 standard. 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 $IDS^{(n)}$, thus an eavesdropper can only get random wraps in this process. The real tag ID is sent through the encrypted message D in the following conversation. To prevent the possible violation of the location privacy of a tag owner, $IDS^{(n)}$ will be updated after each successful communication session with a valid reader by calculating $IDS^{(n+1)}$ separately in both side. Further more, by update the $IDS^{(n)}$ and the 4 keys after the mutual authentication, a future security compromise on an RFID tag will not reveal data previously transmitted and forward security can be guaranteed.

(2) Mutual Authentication:

For authenticating the reader to the tag, a validate reader mast have the proper privilege to access the database and distill its current 4 keys $K_i^{(n)}$ ($i\in[1,4]$). By enrolling mutual authentication and encryption functions to the EPC standard, we can fend off many threats like exposed tag's passwords, malicious snooping readers, disgruntled employee, Cloned Tag, man-in-the-middle attacks, et al.

(3) Prevent *De-synchronization* Attack:

As described in article [3], an active attacker can initiate a man-in-the-middle attack at first. Then, he or she pick a bit from the same position of messages A to D and perform ⊕ operation to change the random numbers R_I or R_T during the authentication or updating process. According to the equations from A to D, both sides may have certain possibility to accept the amended numbers. For example, R_T is generated by tag's PRNG, reader side may be fraud if R_T tempered. And it may finally influence the updating result E. as described in step 10 in figure 2. R_I and R_T must be successfully fraud on the same time before the final updating message E can be accept be the tag. Different from former work, our solution implement mutual authentication. Successful attack can also influence the storage in reader side. Table 2 depicts the corresponding result after attacking different messages.

Table 2. Updated storages after attack

Attacks	*Reader storage*	*Tag storage*	*Success*
A, B	[IDS, K_1,K_2,K_3,K_4,]	[IDS, K_1,K_2,K_3',K_4',]	50%
C, D	[IDS', K_1',K_2',K_3,K_4,]	[IDS, K_1,K_2,K_3,K_4,]	50%
A,B,C	[IDS', K_1',K_2',K_3,K_4,]	[IDS, K_1,K_2,K_3',K_4',]	25%

Different from former work, successful attack can also influence the storage of reader or DB record in our protocol. To solve the problem, article [3] try to storing status information in tag's memory. Although they can prevent the multiple trials from attacker and distinguish abnormal tag, an attacker still has high probability to cause DOS attack to prevent the tag from successive update. To increase the robustness of the entire system, storing all the necessary information at the database side still should be the best solution. And a tag's availability can be discerned by recording its retry times at server side.

The keys values are depend on which IDS can be found in step 5. By doing so, even the corresponding tag did not update its IDS and keys in last round, it still can be identified in the next round. The failure of updating may either caused by unmatched IDS_T and IDS_I in step 11 or by losing power. Then, we can prevent the De-synchronization attack in our protocol with no extra addition storage in EEPROM.

(4) Prevent *Full Disclosure* Attack:

Full disclosure attack is a simple but effective method to break the XOR based encryption methods, such as LMAP and M^2AP. It is based on repeatedly run the incomplete protocol many times (96 trials in LMAP) at the tag side by changing the j-th bit of A and B respectively. By judging the response message received from tag, the full bit values of R_I can be disclosed. After get R_I, disclosing the rest parameter can be much easier.

To counter this attack, we use the message pare C||D in step 7 to act as the replication from tag to reader. If the reader is successively authenticated, the reply will assigned D. otherwise, a failure notice message D′ will be generated to compose the reply. And because a random number R_T is contained, which generated from the tag's PRNG in each trial. The attacker can not get any clue on from the reply. The attacker can only wait the response of the reader to decide whether the change of the j-th bit is accepted. However, the repeating trials by the attacker can easily be identified and insulated by a stateful reader. Then, we can defend this attack through a simpler way.

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 XOR (⊕), bitwise OR (∨), bitwise AND (∧), and addiction mod 2^m (+) in our protocol. 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. Totally 4 encryption and decryption algorithms and 5 updating algorithms need to be added into tag's processing sequence.

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. The EPC, TID, and User 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. Here we divide these memories into 4 pairs, $APwd_M$, $APwd_L$, $KPwd_M$ and $KPwd_L$, each represents a 16 bits password share and used as K_1 to

K_4 in our protocol. Different with LMAP and M^2AP whose key's length are 24 bits, our solution shorted the length of the keys to 16 bits to accompany with the half length of the passwords in EPC C1G2 standard. For the same reason, we also use the 16 bit PC from EPC memory to act as ID in our protocol. Shorter keys may decrease the crypto strength under brute force attack. Considering the cost of increasing hardware of tag, it is still a worthy 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 add a 16 bits IDS. And four 16 bits keys, K_1 to K_4, are also added which need 64 bits at all. Totally, we need increase 80 bits rewritable memory storage space from User memory. EPC memory and TID memory are leaving unchanged, which still can be achieved after authentication. To facilitate authentication process, the length of IDS is set equal to the length of random number or half of password length.

7. Conclusion

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 a PGMAP in this paper concerning security attributes of EPC C1G2 standard. We take advantage of the existing functions on the EPC C1G2 tag and used its PRNG. As the random numbers and keys are all 16 bits in our protocol, our protocol can be easily merged into the EPC C1G2 scheme. Our solution may be not fully secure but it is simple, cost-effective, and light-weight to be implemented on tag. Through the three phases in our protocol, we can thwart many existing threats. Our improvements try to avoid extra hardware requirement in EPC C1G2 tag, which guarantee PGMAP can be easily applied to real application environments. Important related problems, such as implementation performance and security verification, will be addressed in future reports.

8. 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] 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.

[4] 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/.

[5] D. Bailey and A. Juels, "Shoehorning Security into the EPC Tag Standard," in Security and Cryptography for Networks, vol. 4116/2006: Springer Berlin / Heidelberg, 2006, pp. 303-320.

[6] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong Authentication for RFID Systems Using the AES Algorithm," in Cryptographic Hardware and Embedded Systems - CHES 2004, 2004, pp. 357-370.

[7] Z. Luo and T. Chan, "A Lightweight Mutual Authentication Protocol for RFID Networks," presented at ICEBE, 2005.

[8] J. C. H.-C. Pedro Peris-Lopez, Juan Estevez-Tapiador, Arturo Ribagorda, "RFID Systems: A Survey on Security Threats and Proposed Solutions," Lecture Notes in Computer Science, vol. 4217, pp. 159-170, 2006.

[9] T. Dimitriou, "A Secure and Efficient RFID Protocol that could make Big Brother (partially) Obsolete," presented at Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06), 2006.

[10] L. Lu, J. Han, L. Hu, Y. Liu, and L. M. Ni, "Dynamic Key-Updating: Privacy-Preserving Authentication for RFID Systems," presented at Fifth IEEE International Conference in Pervasive Computing and Communications (IEEE PerCom), White Plains, NY, USA, March 2007.

[11] 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.

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

[13] 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.

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

[15] 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.

[16] C. Hung-Yu and C. Che-Hao, "Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards," Comput. Stand. Interfaces, vol. 29, pp. 254-259, 2007.

[17] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "Cryptanalysis of a Novel Authentication Protocol Conforming to EPC-C1G2 standard," presented at RFIDSec 2007, 2007.