17

An Improved Forward Secrecy Protocol for Next Generation EPCGlogal Tag

L.M. Cheng, C.W. So and L.L. Cheng
City University of Hong Kong
Hong Kong

1. Introduction

Radio Frequency Identification (RFID) (Landt, 2001) is a prevalent technology that replaces barcode technology and it will be massively applied in both consumer and commercial products as the trend predicts. However, the computation power and memory of RFID including the EPCGlobal Gen-1 and Gen-2 RFID tags are restricted. These made the implementation of well-known cryptographic algorithms, both computational and memory intensive, on the tags not possible.

Although various cryptographic privacy enhancing technologies for RFID have been proposed, they include Hash-lock approaches, Digital Signature approaches, Encryption approaches, Time Stamping approaches, Pseudonyms approaches and Challenge-response approaches, the EPCGlobal tags continue to operate in a limited security protection.

The Hash-lock approach (Weis, 2003) is based on locking the tag using a hash of the key on the tag, where the key is stored in a back-end server. This approach assumes that tags can not be operated securely in a long isolated environment. This approach can be used for authentication, by matching the right hash of key. The cloning resistance is weak and enhanced techniques providing better privacy protection and scalability have been derived (An, 2005; Nohara, 2005; Wang, 2007).

Digital signatures approaches (Juels, 2003; Bono, 2005; de Dormale, 2005) provide better tracing and forgery resistance of RFID during authentication process. The approaches use a PKI encryption technique to avoid static identifiers and information to be read by others. Authentication is performed by verifying data on the tag is signed using a valid public-key digital signature to check the valididty of authentication.

Encryption approaches (Golle, 2004; Feldhofer, 2004; Ranasinghe, 2004; Ateniese, 2005) are similar to digital signature approaches except simplified private key standard or propriertay cryptographic algorithm is used.

Time stamping approaches (Glidden, 2004; Molnar ,2005; Tsudik, 2006; Ith, 2007) are the most popular approaches which provide a dynamic matching of time information that help avoiding replay attacks.

Pseudonyms approaches (Juels, 2004; Juel, 2006; Molnar, 2005; Avoine, 2005) is very similar to time stamping approaches except the dynamic information is scheduled from a pre-defined list of pseudo-random data called pseudonynms.

Challenge-response approaches (Ree , 2005; Dimitriou, 2006; Duc, 2006; Chien, 2007) are the most secured techniques developed from multi-pass authentication process to provide a wide range of security and privacy protection.

Source: Development and Implementation of RFID Technology, Book edited by: Cristina TURCU,
ISBN 978-3-902613-54-7, pp. 554, February 2009, I-Tech, Vienna, Austria

In actual situation, EPCGlogal Gen 1 (Garcia-Alfaro et al, 2008) tag implementation uses protocols that only require RFID readers to use the tags' unique serial numbers to identify the tags. Tags with the same ID will certainly confuse the reader. Gen 2 rectifies this problem by allowing reader to read tags even if two or more tags have the exact IDs. The unique sequence of communication makes anti-collision algorithm more robust and reduce the possibility of interference from other tags when a reader is talking to a certain tag.
In order to handle reading multiple tags reliably and securely, a number of security techniques have been deployed in Gen 2 (EPC, 2005; Roberti, 2005). They are Session Concept, Dense Readers Conditions, Enhanced Secured Protocols, Ghost Reads Improvements and Covered Coding. However, there are still some privacy concerns in both user and application levels left in the standards and a security loophole in defining and managing the key 'random number' that require attention in order to avoid possible privacy violation or information leakage by eavesdropping on the communication channels.

1.1 Fake tag ID (ghost reads) problems

Fake Tag ID (Ghost Reads) is the major reading problems with Gen 1 tag. Noise or glitches are a hurdle in adopting this valuable technology. As confirmed by the report from RFID Alliance Lab (Deavours, 2005), the Class 0 ghost read rate is about 1.3 per 1,000. These ghost reads can create havoc in many solutions. Gen 2 protocol has an edge on that and it comes with a very vigilant solution to tackle this problem using a 'Query' Concept. The 'Query Concept' establishes a mutual authentication flow to allow secure exchange data and to eliminate fake tag ID problem.
Strict timing constraints in Tag-Reader communications create an illusion of full duplex link. In fact, the communication is still operated in a half-duplex mode. Tag will not talk when it is listening to the reader commands but the timing constraints make sure that tag must response to reader command within a preset time. If the tag fails to response within the preset time, the task will be terminated and the entire process has to be started from beginning.

1.2 Password protection and effective randomness

A secure communication channel is essential for data transmitted over an air interface. In Gen 1 Class 1, an 8-bit password was used with a 'kill' command to safeguard the data. However, this 8 bit password is not secure and eases to break because of just 256 possible values. In Gen 1 Class 0, a 24-bit password is used which gives a better data protection against eavesdropping. Gen 2 with even better safeguard uses 32 bit password while offering 4 billion possible values and makes the brutally search for the correct password difficult and thus provides a very high level of secure communication that EPC tags never had before (Roberti, 2005).
To strengthen the password requirement in communications, a random number is used in Gen 2 to scramble the data commuted. Tags will generate and use a 16-bit pseudo-random number generator (PRNG) throughout the communication link session. Because the PRNG is close to truly random, the communication link is ensured to be safe. For example, having a tag population of up to 10,000, the probability that any two or more tags simultaneously generate the same sequence of RN16s (16-bit random number) is less than 0.1%. The 32 bit password protection in Gen 2 is further enhanced by using "cover coding" while EXORING the data with random numbers to mess up the data (EPC, 2005) during transmission, a matching random number is needed to recover the transmitted data at the receiver end.

1.3 Security problems and possible attacks

With increasing mobility requirements, RFID readers are integrated in a handheld device or even in mobile phones. The low-cost tags are likely the factor for widespread adoption of the technology, deployment on such massive scale has created new threats to user and application privacy due to the powerful tracking capability of the tags (Luo et al, 2005). All UHF standards do provide a security mechanism for reading user memory but any reader can read the tag ID on fly. Security check must be imposed on the tag before transmitting the ID and a mechanism should be defined to recognize the trusted reader to address privacy concerns.

The transmission protocol (EPC, 2005) defines the mechanism to exchange instructions and data between the reader and the tag, in both directions. It is based on the concept of "interrogator (reader) talks first" and it simply means that every tag compliant to UHF standards will always answer to reader's query with its identification (ID) at very first hand. This makes the RFID technology susceptible and any intruder reader can track a tag. The attacker can obtain concrete product information associated with EPC/UID code. This product information is usually provided in the public network. Although current UHF protocols have 'kill' command/option which makes tag presently dead and can be executed before moving the tag in the hands of end-users but it is not the solution for most applications. Some applications require permanent tag tracking, for example, tags associated with objects for security purposes, personal identification systems, vehicle tracking system etc.

Another tag security issue relates to the scenario. Since the communication between a tag and a reader is by radio means, anyone can access the tag and obtain its output, i.e. attackers can eavesdrop on the communication channel between tags and readers, which is a cause of consumers' apprehension. Therefore, the authentication scheme employed in RFID must be able to protect the data passing between the tag and the reader, i.e. the scheme itself should have some kind of encryption capability (EPC, 2005).

Gen 2 provides a good mechanism for securing the data communication between the tag and reader. The exchange of cover-coding is first initiated by a random number request, i.e. RN16, from the tag. If a lower secured mechanism or plaintext only is used, eavesdrop on the communication channel will break the entire security process of the cover-coding. The generation and management of this 'random number' are vital for ensuring the security and integrity of the system but its size should be reconsidered and time of command to response should be restricted with precise values. So that, random number and time for command to response should be directly proportional. Although the random RN16 secures the communication link but its 16 bit size still makes it susceptible as generating or searching 65536 combinations is very easy with ordinary processors. The duration of command to response time makes it more vulnerable which means that reader A would start querying the tag but reader B (an intruder) can jump in the communication link with fake random numbers.

1.4 Possible solutions

Duc et al (Duc et al, 2006) proposed schemes for enhancing security of EPCglobal Gen-2 RFID tag against Traceability and Cloning. It enhances the weaknesses of Rhee (Rhee, 2005), Juels (Juels, 2006), and Dimitriou (Dimitrios, 2006) schemes, which are either not conform to EPCGlogal standard or unable to resist the privacy or/and DoS attack.

Duc et al's authentication relies on the synchronized session key between the tag, *T*, and the server, *S*, an adversary can initiate replay attack, man-in-the-middle attack and brute force attack that will cause DoS in the RFID system. If any one of the "end session" command was intercepted, the shared session key between *T* and *S* will be out of synchronization. As a result, *T* cannot be authenticated anymore. Thus, Duc et al.'s protocol is not able to resist the DoS attack, and it does not provide forward secrecy to the RFID system. Chien (Chien, 2007) provides an enhnacement to Duc's approaches by introducing a pair of old (previous) and new Session Keys, and a pair of old and new random number to avoid DoS attack but it cannot resist Man-in-the-Middle attack caused by a spoofed reader.
We start our discussion in Section 1 with a short introduction and Section 2 we present Duc et al's scheme as an enhancement to both Juels' and Dimittriou's schemes, and we elaborate how Duc's technique will fail. In Section 3, we will give a scenario of all possible threats in RFID environment. In Section 4, we will propose a new security protocol to close these security loopholes and the corresponding simulation results in Section 5. The security analysis of the newly proposed scheme will be given in Section 6. We will conclude new security RFID solutions in Section 7.

2. Duc et al's scheme review

Duc et al. proposed a communication scheme (Duc et al, 2006) to protect user privacy for RFID system. The scheme based on a synchronous session key between tags and back-end database server to authenticate each other. This mutual authenticate scheme takes the advantages of the hash properties of CRC function and a PRNG that are supported by EPCglobal Class-1 Gen-2 tags. The underlying idea is by using the same PRNG with the same seed at both tag and back-end database to generate the same session key on both side. To prevent tag send static message before update of the session key, a random number is added in the authentication process. Data will be encrypted by performing logic operation $\oplus$ with the session key before transmission. Session key will be updated after each successful authentication. The following paragraphs will briefly explain the protocol flow.

2.1 Symbol notations

T	-	RFID Tag
R	-	RFID Reader
S	-	Backend Database Server
r	-	Pseudo-Random Number Generated by Tag's PRNG
CRC(:)	-	CRC Function
PRNG(:)	-	PRNG Function
K_i	-	Session Key for i^{th} Session
A	-	Adversary

2.2 Initialization of tags and back-end database server

Initially during the manufacturing time, the tag has assembled with its EPC and the necessary parameters for the PRNG. A random seed number for PRNG and PIN is chosen and then stored into both *T*'s memory and *S* entry corresponding to the matching EPC. This is very important that each EPC must exactly match with its PRNG seed number and PIN, otherwise the tag can not be authenticated by the back-end server.

2.3 Communication channel between *R* and *S*

The scheme assumes that *R* is communicating with *S* in a secure channel, both *R* and *S* are able to perform standard cryptography authentication. *S* can send the EPC and data to *R* in an encrypted form. *S* can even depend on the privilege of *R*, to determent what kind of information can send to the reader.

Protocol flow

Figure 1 shows the protocol flow of Duc et al's scheme. The flow is illustrated as below:

Step 1. First of all, *R* sends a query request to *T*

Step 2. *T* generates a nonce *r* and form the message $M_{1T} = CRC(EPC \| r) \oplus K_i$ and $C = CRC(M_{1T} \oplus r)$. CRC in M_{1T} actually is acting like a hash function while in C is functioned as an error detection function. M_{1T}, C and *r* will then be sent to R.

Step 3. R forwards M_{1T} , C and r to S.

Step 4. For each tuples in *S*, it generates a message M_1 in the same way as M_{1T} in *T* until a match where $M_{1T} = M_1$ is found. If matched, *T* is successfully identified and authenticated. *S* forwards *T*'s information to *R*. If match failed, *S* will send a tag reject message to *T* via *R*. Information on *T* will be updated if *R* is authenticated to *T* with the generation of M_2. *S* uses the matched tuple's EPC, PIN and K_i to generate the message M_2, where $M_2 = CRC(EPC \| PIN \| r) \oplus K_i$. Finally, *S* will send the corresponding object data and M_2 to *T* via *R*.

Step 5. *T* generates a message M_{2T} to verify M_2 from *R*. *T* uses its EPC, PIN, *r* and K_i to generate the message M_{2T} same as M_2. If M_{2T} matches M_2, data exchange is XORING data with the session key K_i to encrypt or decrypt. However, if M_{2T} does not match M_2 , R is rejected and the session ends immediately. When data exchange is completed, *R* sends an "end session" message to both *S* and *T*. Both *S* and *T* updates the session key where $K_{i+1} = PRNG(K_i)$ and wait till a new session starts.

3. Possible attacks and vulnerabilities on Duc et al.'s scheme

Duc et al.'s protocol is not able to resist the DoS attack and it can not provide forward secrecy to the RFID system. Since this authentication reply on the synchronized session key between *T* and *S*, an adversary can initiate replay attack, man-in-the-middle attack and brute force attack and causes DoS in the RFID system. If any one of the "end session" command was intercepted, the shared session key between *T* and *S* will be out of synchronization. As a result, *T* can not be authenticated anymore. The above DoS attacks actually are based on this vulnerability, aiming at intercepting the delivery of the "end session" command sent from *R* to *T*.

3.1 Replay attack

An adversary can use a spoofed *R* to send a query request to tags, then record the replay messages M_{1T} and nonce *r* from *T*. Recorded message will replay with a session started with an authorized *R*, finally *S* will update its session key while *T*'s session key will remain unchanged. As the session key is out of synchronization between *T* and *S*, therefore *T* can not be authenticated anymore. This is one of the high level threats to the RFID system, as the replay attack can perform on a large numbers of *T* at a time.

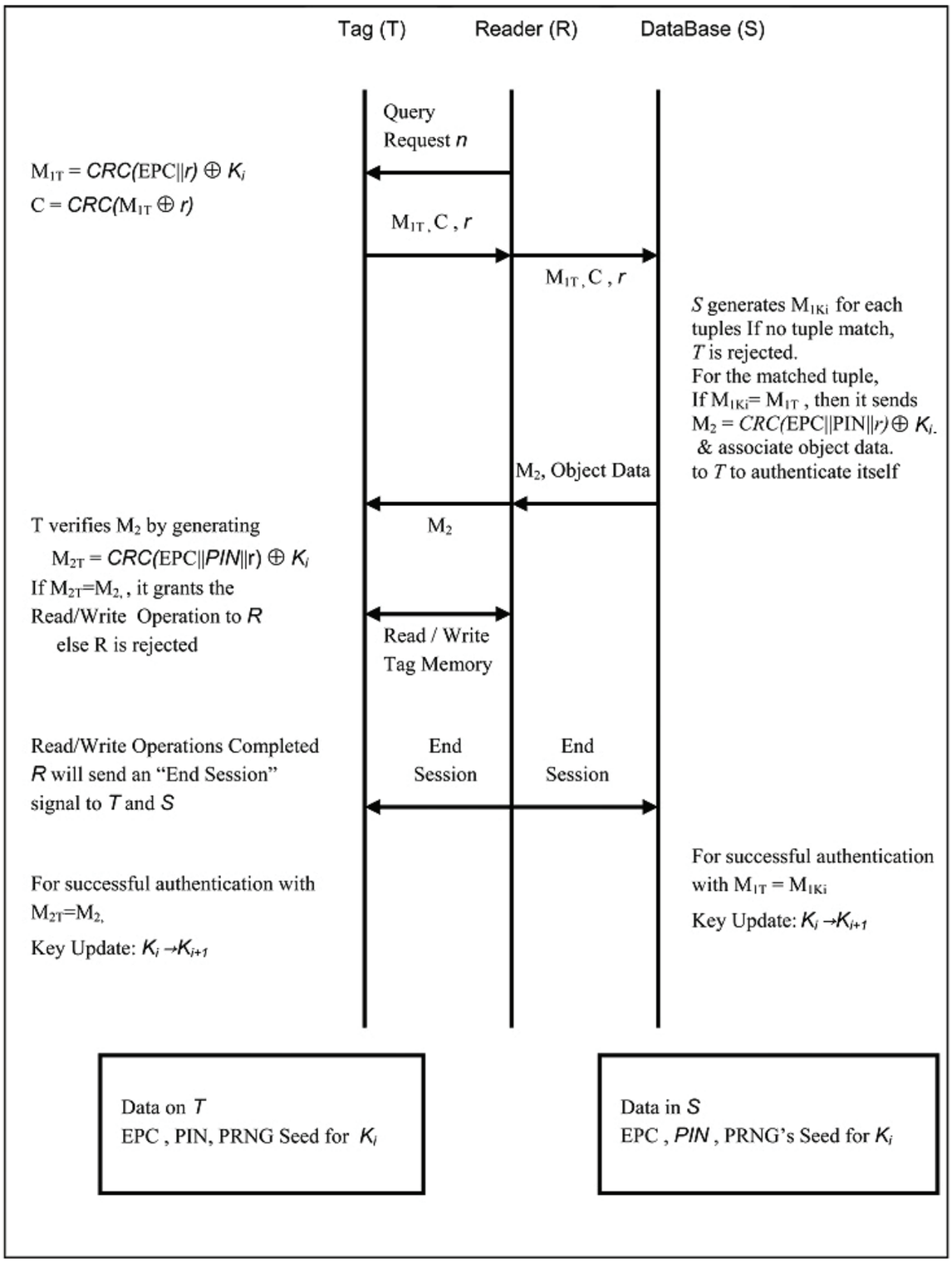

Fig. 1. Duc et al's Protocol

3.2 Man-in-the-middle attack

Man-in-the-middle attack is very similar to replay attack, an adversary acts like a hub to store and forward messages between *R* and *T*. However, an adversary will intercept the command "end session" from *R* to *T*, to make the session key out of synchronization. Man-in-the-middle attack is a high level threat to the RFID system too, as it can also perform on a large numbers of tag at a time.

3.3 Brute force attack

Since tag-to-reader authentication relies on the correspondence between nonce *r*, EPC and session key. It is important to take note that, before the update of the session K_i in a successful authentication, the session key will remain unchanged, while the EPC is always a constant. Therefore, the variance of M_{1T} is basically determined by *r*. An adversary can take this property to initiate a brute force attack on the message M_{1T}. A random message M is chosen, then an adversary can send along with different *r* in each session until a reply of M_2 from reader. As the length of *r* is 16 bits only, the maximum trial times for *r* in a particular M is only 65536. The probabilities that the random message finds a match in *S* is mainly depends on the number of tuples exist in *S*. This is a very dangerous attack to the whole system, as the message length of M_{1T} is 16 bits only, an adversary can send all the combination of M_{1T} and *r* to *R*, it only cost 2^{32} trial times to match all the tuples exists in the database.

3.4 Forward secrecy

If the tag is compromised, an adversary can obtain the EPC, PIN, K_i . From the eavesdropped communication data, we can trace the past communication record between *T* and *R* by computing the respective M_{1T} and M_2 with the obtained parameters. For instance, an adversary can take $M_{1T} \oplus M_2$ from the past communication, that can eliminate the session key and remain only the $CRC\,(\text{EPC} \oplus r\,) \oplus CRC\,(\text{EPC} \| \text{PIN} \| r\,)$. Then we may use the obtained parameter from *T* and generate with *r* to trace the past communication of *T* from the eavesdropped past communication data.

4. Proposed new protocol

With the understanding of the possible attacks and vulnerabilities in Duc et al.'s security scheme, a new security scheme that improves the security performance for RFID system is proposed.

The major differences between the proposed scheme and Duc et al.'s scheme are the additional random number challenge from the reader, and the database will keep the old session key for each tag, update the access PIN after each successful authentication and acknowledgement of M_2 from *T*. The flow of the proposed protocol is further explained below.

The tag T was manufactured and assembled with its corresponding EPC with preset parameter for the pseudo random number generator PRNG. A random seed number for PRNG and PIN was chosen and was stored into both *T*'s memory and backend data server S with entry corresponding to the matching EPC. The database will store the session key K_{i-1} and PIN_{i-1} after the first authentication. The communication between Reader *R* and server *S* was through a secure channel of which cryptographic algorithm can be used in

authentication and for the object data exchange. The protocol below can also provide secured communications between *R* and *T* even for an insecure wireless channel.

4.1 Proposed protocol flow

The protocol flow for proposed scheme is shown in Figure 2. The protocol sequences are as follows.

Step 1. *R* generates a 16-bit random number *n* by its Pseudo Random Number Generator (PRNG) and sends it together with Query Request message to *T*.

Step 2. *T* generates a 16-bit random number *r* by its PRNG and the message $M_{1T} = CRC\,(EPC \| n \| r) \oplus K_i$ and the error checksum code $C = CRC\,(M_{1T} \oplus n \| r)$ and sends M_{1T}, C and *r* to *R*. [N.B. ⊕ is an exclusive OR function]

Step 3. *R* checks $C = CRC\,(M_{1T} \oplus n \| r)$ and detects any transmission error in the channel and *R* forwards M_{1T}, C , *r* and *n* to *S* if no error was found, otherwise, the tag is rejected.

Step 4. *S* generates $M_{1Ki} = CRC\,(EPC \| n \| r) \oplus K_i$ and $M_{1K(i-1)} = CRC\,(EPC \| n \| r) \oplus K_{i-1}$ for each tuples in *S*.

Step 5. If no tuple matches for $M_{1Ki} = M_{1T}$ or $M_{1T} = M_{1K(i-1)}$, the tag is rejected.

Step 6. If $M_{1T} = M_{1K(i-1)}$, it reveals that the session key is out of synchronization. The following steps are then executed.

Out of Synchronization Flow:

Step 6.1 *S* generates $M_2 = M_{2K(i-1)} = CRC\,(EPC \| PIN_{i-1} \| n \| r) \oplus K_{i-1}$ and sends it to *T* via *R*.

Step 6.2 *S* then informs *R* to send the "end session" command to *T*

Step 6.3 T updates its K_i and PIN_i after receiving the "end session" command, *S* continues to keep both K_i and PIN_i and K_{i-1} and PIN_{i-1} unchanged. In this session, *R* will not perform any read and write operation to *T* .

Step 6.4 Finally, *R* will re-initiate a new session with *T* using an updated session key.

Step 7. If the tuple is matched where $M_{1T} = M_{1Ki}$, ***S*** generates $M_2 = M_{2Ki} = CRC\,(EPC \| PIN_i \| n \| r) \oplus K_i$. *S* sends M_2 and the associated object data to *R*. *R* then forwards only M_2 to *T*.

Step 8. *T* verifies M_2 by computing $M_{2T} = CRC\,(EPC \| PIN_i \| n \| r) \oplus K_i$, if $M_{2T} = M_2$, i.e. *R* is authenticated, reading and writing *T*'s memory is granted to *R*; otherwise, the request from *R* is rejected.

(Note: Data exchange between *T* and *R* is encrypted and decrypted by exclusive OR operation ⊕ with the session key K_i.)

Step 9. When *R* has finished the reading and writing operation to *T*, *R* sends an "end session" command to both *R* and *S* to trigger the key update process. Both *T* and *S* will update K_i and PIN_i using $K_i = PRNG(K_{i-1})$ and $PIN_i = PRNG(PIN_{i-1})$. A session is completed at this stage.

5. Simulation

In order to find out the average appearing time of M_{1T} for a given tag, a simulation programme was built using VB.net and MySQL data base to test both Duc's and our proposed protocols. The number of occurrences of M_{1T} in each successful key update session was measured and used for comparison. The tests was based on 65,535 random entries using a PRNG satisfied Gen-2 requirements with a repeated period of 59,092.

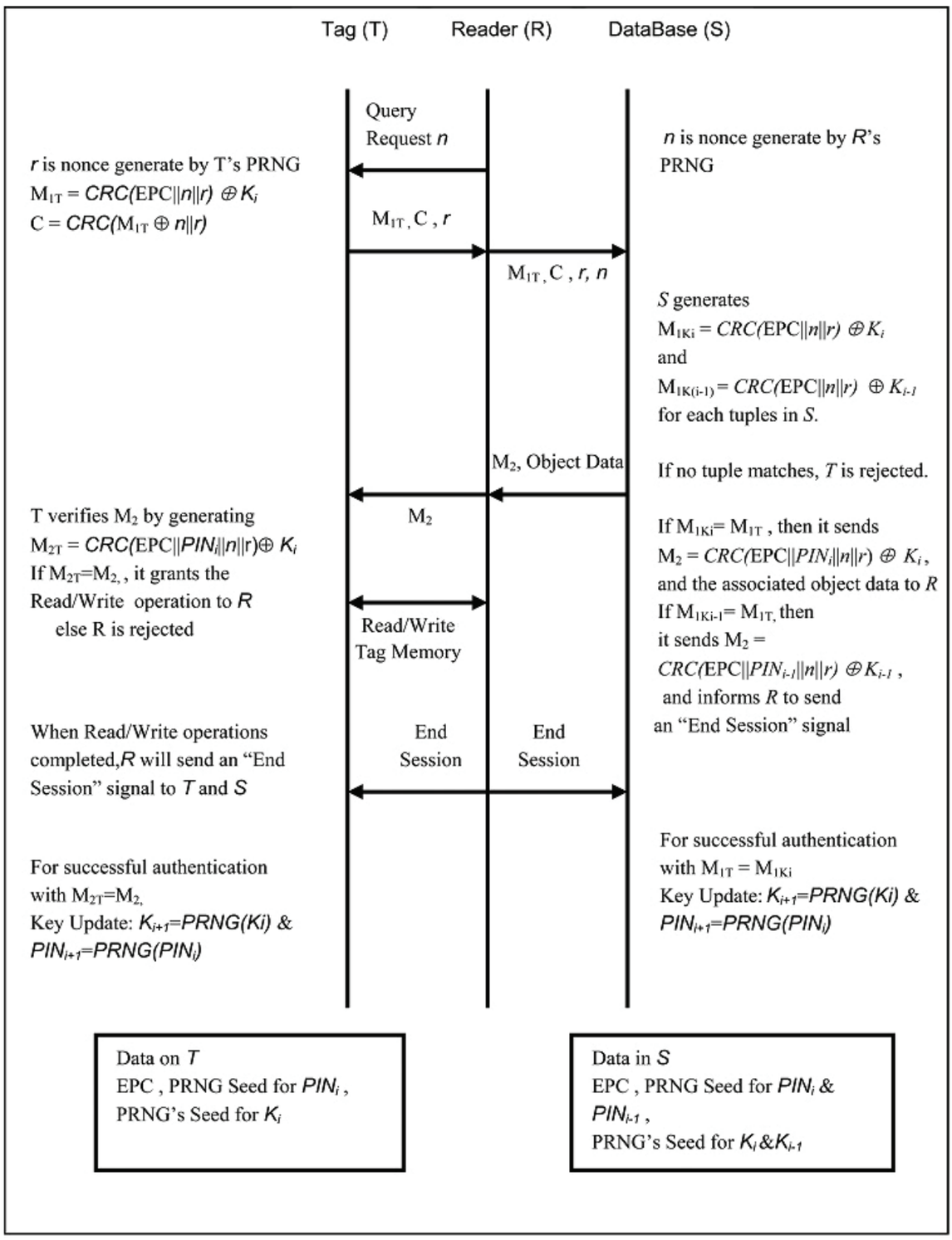

Fig. 2. New Proprosed Protocol

5.1 Pseudo-random number generation

There are four essential parameters for each tag which includes a 96-bit EPC, 32-bit PIN, 16-bit K_i and PRNG's parameters. EPC , PIN_i and seed for K_i are randomly chosen from a PRNG. The random number should satisfy the Gen-2 tag requirements.

A popular class of PRNG, linear congruential generator in the form of $X_{i+1} = (aX_i+c) \bmod m$ was selected; where a, c and m defined the PRNG parmeters, X_i was the seed. In our simulation, a = 61979, c = 0 and m = 59093 were used. A repeating period of 59092 was obtained and satisfied the requirement of Gen-2 tag.

5.2 Protocol simulation programme

Figure 3 shows the layout of the simulation programme layout. The main screen in the centre simulates the protocol flow. The grid view on the left simulates the population of tags, while the right hand side grid view simulatse tag's information tuples maintained in the database. On the right hand corner, the programme performs simulation for M_{1T} out of a given trial times before a successful session key update for a selected tag in the tag's grid view. In order to find out how the CRC function affect the average appearing times out of a given trial, the programme can simulate the generation of M_{1T} for both CRC-16-CCITT used in current Gen-2 tag and CRC-32.

5.3 Duc et al's protocol under man-in-the-middle-attack

An adversary appears in between the tag and the reader, emulating a store and forward hubs. It forwards query request from the reader and then sends M_{1T} , C , r to the reader received from tag, and forward M_2 to tag like an ordinary authentication process. However, after the mutual authentication, it blocks the "end session" command send from reader. As a result, the tag can not be authenticated anymore. Since the tag remains its session key and PIN unchanged while back-end database server updates them with PRNG. The simulation programme shows the tag is rejected in the next authentication. This Man-In-The-Middle Attack simulation shows that the Duc et al's proposed RFID protocol can collapse.

5.4 Simulation on average appearing time M_{1T} and effect of CRC

A tag was randomly chosen to loop recursively to generate 65535 trials for M_{1T} in the simulation programme. The main difference of M_{1T} between the two protocols was the introduction of an additional random number challenge n generated from reader .

The simulation found that M_{1T} was equalled to T_D =1.5148, T_M =1.5312 and T_M =1.5234 for Duc et al and for the new proposed schemes using n=16 bit and n=32 bit respectively, where T_D was the Average Appearing Time out of 65535 Trial for Duc et al.'s Protocol, and T_M was the Average Appearing Time out of 65535 Trial for the Proposed Protocol.

5.5 Simulation result analysis

The simulation result reveals that the period of CRC output in M_{1T} does not follow the period of r. Since r is the only changing parameters in the M_{1T} throughout the trial, it is expected that T_D should approximate equal to r's period. The period of r is found to be 59092 time, therefore the average appearing times out of 65535 trial should be around 1.1. However, T_D is found to be around 1.5 which has a significant difference from r's period.

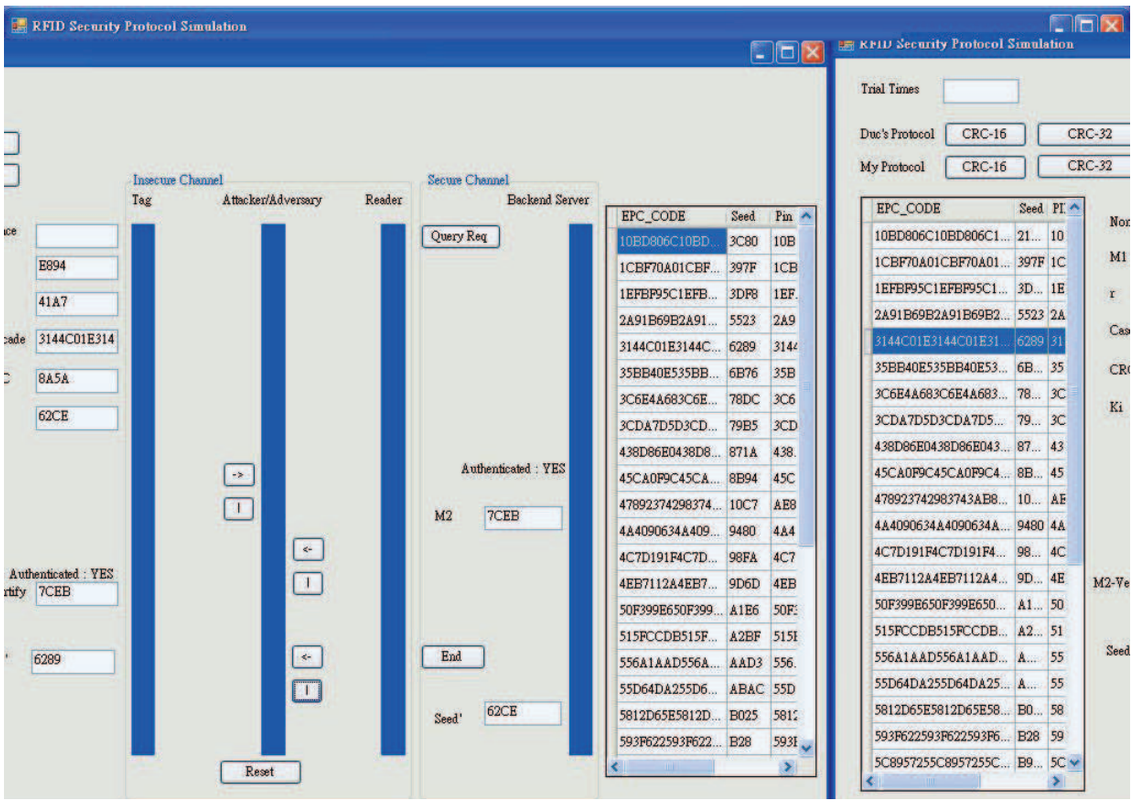

Fig. 3. Protocol Simulation Programme

The simulation results also show that T_M and T_D behaved the same and thus an addition of random number challenge will not reduce T_M, although the combined number from *n* and *r* is found to have no repetition out of the 65535 trials.

Further tests on the effect of CRC length on T_D and T_M was conducted using 32-bit CRC, the results T_D =1.000168 and T_M =1.000015 were obtained using n=16 bit respectively. This concludes that CRC affects the behavior of M_{1T} but not the n.

Since both T_D and T_M have dropped from around 1.5 to around 1 by using 16-bit CRC and 32-bit CRC. It can be concluded that the repetitive of PRNG is caused by the CRC bit size used. Since a 32-bit CRC is not available in the existing RFID standards, its vulernability against replay attack is weak and thus needs enhancement in the next/new generation tags.

6. Security & complexity analysis

6.1 Security analysis

Table 1 provides a security performance comparison of the Duc et al and this proposed schemes related to tag anonymity, data privacy, mutual authentication, forward secrecy, key attack, DoS attack and replay attack.

Tag Anonymity

Tag will never emit static ID, a new random number is chosen from Reader and Tag in each session to ensure tag anonymity.

	Duc et al.'s Protocol	**Proposed Protocol**
Backend Server's Complexity	N O(CRC)	2 N O(CRC)
Tag's Complexity	2CRC + 2PRNG	3CRC + 3PRNG
Reader's Complexity	Send, receive and forward	Send, receive and forward + 1PRNG
Reader Authentication	Yes	Yes
Tag Authentication	Two Phrase	Three Phrase
Spoof Reader Attack	No	Yes
Resist to Dos Attack	No	Yes
Resist to Replay Attack	No	Yes
M_{1T} Collision in Database	No	Yes
Forward Secrecy	No	Yes

Table 1. Security and Complexity Comparison

Where, N is Number of tuples in Back-End Database Server
O(CRC) is the Computational complexity of CRC algorithm.

Data Privacy

Tag never sends any plain text data through insecure channel, data is always encrypted by a session key with nonce. Reader can use cryptography algorithm to exchange data between back-end database server. Therefore, data privacy is protected.

Mutual Authentication

The new protocol performs both tag-to-reader and reader-to-tag authentication. Database authenticates the tag by verifying the message M_{1T}. Tag verifies M_2 generated by database. This mutual authentication scheme ensures data exchange will be granted to authenticated parties only.

Forward Secrecy

Even if the tag is compromised at some time later, as the PIN and session key is updated after each successful authentication, an adversary can not trace and track the compromised tag from the past eavesdropped communication data. Therefore, the forward secrecy is protected.

Key Attack

The shared secret session keys are chosen randomly for each tag and they are different from each other. Exposure for a single key will therefore not expose other's tags secret information.

DoS Attack

The database will maintain six values including the old session key and old PIN for each tag. Even though the tag is out of synchronization with the database, it can still

communicate with the database, by performing a session key and PIN update process to synchronize with database. Although it may increase the communication cost, it can ensure that M_{1T} will not subject to any replay attack.

Replay Attack

The random number challenge from the reader can effectively prevent replay attack from the spoofed tag. The generation of M_{1T} has involved the random number from reader, therefore an adversary can not replay M_{1T} from an eavesdropped communication between spoofed reader and tag.

6.2 Complexity analysis

The proposed security scheme complexity is studied according to its computation, storage requirement and authentication phrase.

Computation Complexity

To communicate with the tag, the reader requires only a PRNG and cryptography algorithm to authenticate to allow the transfer of data between reader and back-end database server. The requirements are feasible in the current generation reader. In the authentication process, reader actually acts like a store and forward hub between back-end database server and the tag with the computation complexity are mainly handled by the back-end database server.

The authentication between the tag and reader two CRCs and one PRNG to generate the message M_{1T}. The Reader authentication process requires one CRC and one XOR operation to verify M_2. The key and PIN update process requires two PRNGs. A total of three CRCs and three PRNGs being used in the whole authentication protocol.

The database generates both M_{1Ki} and M_{1Ki-1} for each tuple, so the computation complexity is equal to 2N (2CRC + PRNG), where N is number of tuples in database.

Storage Requirement

For the tag, it is required to store 3 parameter, i.e. EPC , PIN_i and K_i . For the database, the storage requirement is the same as Duc's scheme and it is required to store five values for each tag. In addition, it requires to store the tag's EPC, PIN_i, PIN_{i-1} and PRNG's seed for K_i, K_{i-1}.

6.3 Authentication phrases

The proposed security scheme is a three-phrase mutual authentication protocol.

Phrase one: Random number challenge from reader.

Phrase Two: Tag generates M_{1T} to authenticate itself to reader.

Phrase Three: Back-end database server generates M_2 which included tag's access PIN to authenticate itself to tag, in order to grant the read and write right to reader.

7. Conclusions

In this chapter, we have evaluated Duc et al.'s security schemes under different attacks and pointed out its vulnerabilities, including DoS attacks, forward secrecy weakness and reader-

to-tag authentication collision weakness. To overcomes these weaknesses, a new protocol is proposed. In our new proposed scheme, it distributes the authentication computational complexity or loading to the back-end database server and the reader and keeping the complexity in the tag unchanged. The scheme also conform to existing EPCglobal Gen-2 specification.

The simulation results conclude that the average appearing time of M_{1T} is affected by the CRC function but not only the random number input. This unwanted repetitiveness can be avoided by using a CRC-32 function instead, so hopefully it will be implemented in the next available generation of RFID tag in the near future.

8. References

An, Younghwa; Oh, Soohyun (2005). RFID System for User's Privacy Protection, *Asia-Pacific Conference on Communications*, pp. 516- 519.

Ateniese, G.; Camenisch, J.; de Medeiros, B. (2005). Untraceable RFID tags via insubvertible encryption, Proceedings of the 12th ACM conference on Computer and *Communications Security,* pp. 92 – 101.

Avoine, G.; Oechslin, P. (2005). A scalable and provably secure hash based RFID protocol, *Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'05)*, pp. 110-114.

Bono, S.; Green, M.; Stubblefield, A.; Juels, A.; Rubin, A. (2005). Security Analysis of a Cryptographically-Enabled RFID Device, 14th USENIX Security Symposium.

Chien, H.Y.; Che-Hao Chen, C.H. (2007). Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards, *Computer Standards & Interfaces Volume 29, Issue 2*, pp. 254-259

de Dormale G. M.; Ambroise, R; Bol, D; Quisquater, J. J. (2005). Low-Cost Elliptic Curve Digital Signature Coprocessor for Smart Cards, Proceedings of the IEEE 17th International Conference on Application-specific Systems, Architectures and Processors (ASAP'06), pp. 347-353.

Deavours, D.D; Ramakrishnan, K.M; Syed, A. (2005). Technical Report ITTC-FY2006-TR-40980-01, October 2005

Dimitrios, T. (2006). A Secure and Efficient RFID Protocol that could make Big Brother(partially) Obsolete, *Proc. Intern. Conf. on Pervasive Computing and Communications, PerCom2006*, Pisa, Italy.

Duc, D.N.; Park, J.; Lee, H.; Kim, K .(2006). Enhancing Security of EPCglobal Gen-2 RFIDTag against Traceability and Cloning, *SCIS 2006*, Hiroshima, Japan.

EPC (2005). EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID, EPCglobal Inc., January 2005.

Feldhofer, M.; Dominikus, S.; Wolkerstorfer J. (2004). Strong authentication for RFID systems using the AES algorithm; Workshop on Cryptographic Hardware and Embedded Systems–CHES,L*ecture Notes in Computer Science, Vol.* 3156, pp. 357-370.

Garcia-Alfaro, Joaquin; Barbeau, Michel; Kranakis, Evangelos (2008). Analysis of Threats to the Security of EPC Networks, *Communication Networks and Services Research.*

Glidden, R.; Bockorick, C.; Cooper, S.; Diorio, C.; Dressler, D.; Gutnik, V.; Hagen, C.; Hara, D.; Hass, T.; Humes, T.; Hyde, J.; Oliver, R.; Onen, O.; Pesavento, A.; Sundstrom, K.; Thomas, M. (2004). Design of ultra-low-cost UHF RFID tags for supply chain applications, IEEE Communications Magazine, *Volume: 42, Issue 8*, pp 140-151.

Golle, P.; Jakobsson, M.; Juels, A.; Syverson, P. (2004).Universal Re-encryption for Mixnets, *Topics in Cryptology – CT-RSA 2004*, The Cryptographers' Track at the RSA Conference, Lecture Notes in Computer Science, Volume 2964, pp. 1988.

Ith, P.; Oyama, Y.; Inomata, A.; Okamoto E. (2007). Implementation of ID-based signature in RFID system; 2007. APCC 2007. Asia-Pacific Conference on Communications,,pp. 233-236.

Juels, A.(2004). Minimalist cryptography for low-cost RFID tag, Conference on Security in Communication Networks – SCN'04, LNCS, Amalfi, Italia, September (2004) Springer-Verlag

Juels, A. (2006). RFID Security and Privacy: a Research Survey, *IEEE Journal on Selected Areas in Communications*, 24 (2), pp. 381-284.

Juels, A.; Pappu., R. (2003). Squealing Euros: Privacy Protection in RFID-Enabled Banknotes, Financial Cryptography – FC'03. Lecture Notes in Computer Science, Volume 2742, Springer-Verlag, Le Gosier, Guadeloupe, FrenchWest Indies, pp. 103—121.

Landt, J. Shrouds of Time, The history of RFID, *AIM Publication*, Ver. 1.0, October 1, 2001. *Conference (cnsr)*, pp. 67-74.

Luo, Zongwei; Chan, T.; Li, J.S. (2005). A lightweight mutual authentication protocol for RFID networks, *IEEE International Conference on e-Business Engineering*, pp. 620 – 625.

Molnar, D.; Soppera, A.; Wagner, D. (2005). A scalable, delegatable, pseudonym protocol enabling ownership transfer of RFID tags, *Handout of the Ecrypt Workshop on RFID and Lightweight Crypto, July*.

Nohara, Y.; Inoue, S.; Baba, K.; Yasuura, H. (2005). Quantitative evaluation of unlinkable ID matching schemes, *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pp. 55-60.

Ranasinghe, D.; Engels, D.; Cole, P. (2004). Security and Privacy: Modest Proposals for Low-Cost RFID Systems, *Auto-ID Labs Research Workshop*.

Rfid (2006): http://www.rfidalliancelab.org/

Rhee, K.; Kwak, J.; Kim, S.; Won D. (2005). Challenge-Response Based RFID Authentication Protocol for Distributed Database Environment, Security and Pursvarive Computing, Lecture Notes in Computer Science Volume 3450, pp. 70-84.

Roberti, M. (2005). Understanding the EPC Gen 2 Protocol, *RFID Journal Special Report*.

Tsudik, G. (2006). Yet another trivial RFID authentication protocol, International Conference on Pervasive Computing and Communications – PerCom 2006, Pisa, Italy, March

Weis, S.; Sarma, S.; Rivest, R.; Engels, D. (2003). Security and privacy aspects of low-cost radio frequency identification systems, *International Conference on Security in Pervasive Computing -SPC 2003, Lecture Notes in Computer Science, Vol. 2802. Springer-Verlag, Berlin Heidelberg New York*, pp. 454-469.

Wang, Xiao-hua; Zhou, Xiao-guang; Sun, Bai-sheng (2007). An Improved Security Solution of RFID system, International Conference on Wireless Communications, Networking and Mobile Computing, pp. 2081-2084.

Development and Implementation of RFID Technology

Edited by Cristina Turcu

ISBN 978-3-902613-54-7
Hard cover, 450 pages
Publisher I-Tech Education and Publishing
Published online 01, January, 2009
Published in print edition January, 2009

The book generously covers a wide range of aspects and issues related to RFID systems, namely the design of RFID antennas, RFID readers and the variety of tags (e.g. UHF tags for sensing applications, surface acoustic wave RFID tags, smart RFID tags), complex RFID systems, security and privacy issues in RFID applications, as well as the selection of encryption algorithms. The book offers new insights, solutions and ideas for the design of efficient RFID architectures and applications. While not pretending to be comprehensive, its wide coverage may be appropriate not only for RFID novices but also for experienced technical professionals and RFID aficionados.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen