SCIS 2006 The 2006 Symposium on Cryptography and Information Security
Hiroshima, Japan, Jan. 17-20, 2006
The Institute of Electronics, Information and Communication Engineers

Enhancing Security of EPCglobal Gen-2 RFID Tag against Traceability and Cloning

Dang Nguyen Duc * Jaemin Park * Hyunrok Lee * Kwangjo Kim *

Abstract— In this paper, we present a synchronization-based communication protocol for RFID devices. We focus on the EPCGlobal Class-1 Gen-2 RFID tag which supports only simple cryptographic primitives like Pseudo-random Number Generator (PRNG) and Cyclic Redundancy Code (CRC). Our protocol is secure in a sense that it prevents the cloned tags and malicious readers from impersonating and abusing legitimate tags, respectively. In addition, our protocol provides that each RFID tag emits a different bit string (pseudonym) when receiving each and every reader's query. Therefore, it makes tracking activities and personal preferences of tag's owner impractical to provide the user's privacy.

Keywords: RFID, Privacy Protection, Cloning Risk

1 Introduction

Radio Frequency Identification (RFID) technology is envisioned as a replacement for Barcode counterpart and expected to be massively deployed in the near future. The advantage of RFID system over barcode system includes many-to-many communication (*i.e.*, one tag can be read by many readers and one reader can read many tags at once), wireless data transmission (versus optical communication, thus requiring light-of-sight, in case of Barcode) and its computing nature. Those major benefits enable much more wider range of applications including: supply chain management, library management, anti-counterfeiting banknotes, smart home appliances, *etc.*

Despite of many prospective applications, RFID technology also poses several security and privacy threats which could harm its global proliferation. Ironically, the security weakness of RFID technology comes from the most basic operation of a RFID tag, that is to wirelessly release a unique and static bit string (usually known as Electronic Product Code - EPC) identifying the object associated with the tag upon receiving the query request from readers. Using those unique EPC as reference, one (equipped with a compatible reader) can track the moving history, the personal preferences and the belongings of a tag's holder. Even worse, absence of authentication results in revealing EPC to malicious readers (referred to as skimming attack). Once capturing EPC, an attacker can duplicate genuine tags and use the cloned tag for a variety of malicious purposes. A natural solution to the security vulnerability stated before is to employ cryptographic protocol in RFID system. Unfortunately, the cost of manufacturing a tag has to be extremely low, *e.g.*, less than 30 cents (according to RFID journal [11], one RFID tag is expected to cost 5 cents by 2007). Therefore, the costful security protocols widely deployed in the cryptographic settings cannot be incorporated into a small chip with tightly constrained computational power.

To foster and publicize RFID technology, several organizations including EPCglobal and ISO have been actively working on RFID standardization. In particular, EPCglobal is a joint venture between EAN International (Europe) and UCC (USA) aiming at developing industry RFID standards. Since EPCglobal unifies the two biggest organizations responsible for Barcode technology, it has the potential to influence the standard for RFID technology at the global scale. One of the most important standards proposed by EPCglobal is the EPCglobal Class-1 Gen-2 RFID specification which defines the functionality and operation of a RFID tag [2]. Unfortunately, the EPCGlobal Class-1 Gen-2 RFID specification pays little attention to the security and privacy issues mentioned earlier. More specifically, RFID tag uses a very naive method to authenticate RFID reader by sending out a random number and requiring the reader to acknowledge that number. After that, a tag backscatters the associated EPC in clear text to the reader. This protocol obviously enables any malicious reader conforming to the standard to perform skimming attack to capture the EPC stored in the tag's memory. Another possibility for an attacker to steal an EPC is to sniffer communication channel between a tag and a reader. It is easy since wireless communication is inherently vulnerable to eavesdropping. As we previously pointed out, EPC is essentially all the attackers need to produce a fake tag (cloning) since it knows for sure that that the captured EPC will be recognized by legitimate readers. On the other hand, the standard is clearly susceptible to tracking since EPC is fixed and sent out unscrambled. We think that these security flaws in the standard will limit its adoption because people are very sensitive to

* International Research Center for Information Security (IRIS), Information and Communication University (ICU), 119 Munjiro, Yuseong-gu, Daejeon, 305-732 Republic of Korea, e-mail: {nguyenduc, jaeminpark, tank, kkj}@icu.ac.kr

security and privacy issues. In this work, we aim to fix that, not by reworking the standard but by using only current capabilities of a tag ratified in the standard. This will minimize the change to the standard and save a lot of work put in the standard so far.

Lots of researchers have proposed several lightweight cryptographic protocols to defend against security and privacy threats. Most of the proposed solutions make use of a hash function [6, 7, 8, 9, 12]. Even though the hash function can be efficiently implemented in low-power hardware, it is still beyond current capability of low-cost RFID tag. In particular, current EPCGlobal Class-1 Gen-2 RFID specification does not ratify cryptographic hash function like MD5 and SHA-1. Thus, we need to look for another solution which should use only the available functionalities of current RFID standards. In fact, Juels [3] suggested such a scheme to prevent the cloned tags from impersonating legitimate tags. However, his protocol did not take eavesdropping and privacy issues into consideration, therefore provides no protection against privacy invasion and secret information leakage. In this paper, we present another scheme targeting most of security features for a RFID system including authentication, traffic encryption, privacy protection as well. Our proposed scheme employs only PRNG and pre-shared secrets between tag and reader/backend server (*e.g.*, PIN, seed to PRNG). We call our scheme *synchronization-based* as ours requires session-key synchronization between tag and reader/backend server like Ohkubo *et. al.* hash-based scheme [6].

The rest of this paper is organized as follows: in Section II, we briefly review some background knowledge and related work. In Secion III, we then describe our proposed protocol followed by heuristic security analysis in Section IV. Finally, we conclude with the final remarks and future work.

2 Background and Related Work

2.1 RFID System

An RFID system consists of three components: RFID tag, RFID reader and backend server. A RFID tag is a small chip attached to an object. It emits a unique bit string serving as the object identity. A RFID reader can be a PDA, a mobile phone or any kind of devices capable of querying object identity stored in a RFID tag. Using object identity as a pointer, a RFID reader can later retrieve detail information about the object stored in backend server's database.

2.2 EPCglobal Class-1 Gen-2 RFID Specification

The latest RFID standard ratified by EPCglobal is named EPCglobal Class-1 Gen-2 RFID specification (Gen-2 RFID for short). We briefly summarize properties of Gen-2 RFID tag as follows [2]:

- Gen-2 RFID tag is passive, meaning that it receives power supply from readers.
- Gen-2 RFID tag communicates at UHF band (800-960 MHz) and its communication range is from 2 to 10m.
- Gen-2 RFID tag supports on-chip Pseudo-Random Number Generator (PRNG) and Cyclic Redundancy Code (CRC) computation.
- Gen-2 RFID's privacy protection mechanism is to make the tag permanently unusable once it receives the kill command with a valid 32-bit kill PIN (*e.g.*, tag can be *killed* at the point-of-sale).
- Read/Write to Gen-2 RFID tag's memory is allowed only after it is in secure mode (*i.e.*, after receiving access command with a valid 32-bit access PIN).

We would like to note that that privacy protection mechanism suggested in the specification is arguable overkilled. In many scenarios like tracking animal, smart home appliances, *etc.*, the tag should never be killed. Furthermore, for the supermarket management, the tag is likely to be helpful in many ways after items being purchased (*e.g.*, for warranty purpose). Therefore, in designing new protocol, we are going to avoid this kind of mechanism and to make a new use for the *kill* PIN. For the access PIN, we want to stress that it is useless from security point of view since the PIN (16-bit per times) is XORed with a 16-bit pseudo-random number sent by the tag in a session. Just by eavesdropping the 16-bit pseudo-random number and the XORed PIN, an attacker can easily recover the access PIN. Losing the access PIN is very dangerous because it allows a malicious reader to read/write the entire memory of a tag.

In this work, we aim at designing a new communication protocol with security properties for EPCglobal Class-1 Gen-2 RFID tag. However, note that our protocol can be applied to other RFID standards as well.

2.3 Pseudo-random Number Generator

PRNG is the most frequently used primitive in cryptography as well as in computer science, electrical engineering, statistics, *etc.* In a common setting, a PRNG is modeled as a deterministic function whose next output is computed from previous outputs (usually the last output). The output sequence starts from a (randomly chosen) seed number. The security strength of a PRNG depends on the period and probability distribution of the output sequence. A popular class of PRNG has the congruential form of $x_i = ax_{i-1} + b \bmod N$ where x_0 is the seed number and a, b and N are PRNG's parameters [10]. In this paper, we will use a PRNG to share a new session key between RFID tag and reader for each and every session. In the EPCglobal Class-1 Gen-2 RFID specification, the RFID tag is capable of generating 16-bit pseudo-random number with the following properties [2]:

- The probability that a single 16-bit number j is drawn shall be bounded by $\frac{0.8}{2^{16}} < \mathbf{Prob}(j) < \frac{1.25}{2^{16}}$.

- Among a number of 10,000 tags, the chance that any two tags simultaneously generate the same 16-bit pseudo-random number is less than 0.1%.
- The probability of guessing the next pseudo-random number generated by a tag is less than 0.025% under the assumption that all previous outputs are known to an attacker.

Since Gen-2 standard requires only 16-bit pseudo-random number, the security margin (*i.e.*, success probability of adversary) of a security protocol using such PRNG is usually bounded by $\frac{1}{2^{16}}$. We suggest that Gen-2 standard should support 32-bit PRNG to take full advantage of 32-bit PIN currently supported by Gen-2 specification. It is because XORing two halfs of a 32-bit PIN with the same 16-bit nonce in one session provides no better security than using the full 16-bit PIN.

2.4 Checksum Code

In our proposed protocol, we also make use of checksum code to provide security and resolve possible collisions at backend server's database. A checksum code is often used to check the integrity of data being sent or received. The popular cryptographic checksum codes are cryptographic hash function, MAC and HMAC. In this paper, we will make use of a well-known, efficient (yet less cryptographically strong) checksum algorithm, namely CRC [13]. This kind of checksum code is currently ratified in EPCglobal Class-1 Gen-2 RFID specification, version 1.09 [2]. CRC algorithm treats binary data as a polynomial whose coefficients are in GF(2) (*i.e.,* 1 or 0). For n-bit CRC, an irreducible and primitive polynomial of n degree (called CRC polynomial) over GF(2) should be chosen. The CRC checksum is then computed as a remainder of the division of the original data by the CRC polynomial. For example, the polynomial $x + 1$ is a CRC polynomial resulting in 1-bit CRC checksum equivalent to parity bit. In EPCglobal Class-2 Gen-2 specification, a 16-bit CRC checksum is used to detect error in transmitted data and the corresponding CRC polynomial of degree 16 is $x^{16} + x^{12} + x^5 + 1$. Even though calculating CRC checksum involves polynomial division, it actually can be implemented very efficiently by using *shift register* in hardware and *look-up table* in software. Generally, if CRC is setup properly, we can expect that the probability of collision on n-bit CRC checksum is about $\frac{1}{2^n}$. Of course, we can always use cryptographically secure checksum algorithms as well.

3 Our Proposed Protocol

Main Idea. We first think of protecting data transmitted between the tag and reader against eavesdropping. The obvious way is to utilize encryption/decryption and the most simple encryption function that we are aware of is XORing which is popularly used in a stream cipher). The problem now turns to key management issue: that is to ensure that a new encryption key is used in every session. Solving this issue turns out to be a solution to privacy protection as well since RFID tag can Xor EPC with different key in every session, thus, prevent malicious readers from tracking the tag. And we suggest that the simplest, yet most efficient way of key sharing in this scenario is to use the same PRNG with the same seed at both RFID tag side and reader/backend server side. The session key can be computed by generating a new pseudo-random number from current session key after every session. This computation is required to be done at both RFID tag and reader/backend server in a synchronous way. Otherwise, subsequent traffic cannot be understood by both sides.

The next security problem that we need to solve is authentication. We argue that, in most cases, a reader just needs to know EPC stored in a tag and then eventually contact the backend server to get/update information about the object carrying the tag. Keeping this in mind, we propose that reader-to-tag authentication can be delegated to tag-to-backend server authentication. More specifically, reader can only receive EPC from RFID tag in an encrypted form. It needs to authenticate itself till backend server first, and then, depending on its privileges, backend server can decide what kind of information to send back to reader (for example, in case of a public reader, only information describing what the referenced object is; and in case of a manufacturer's reader, actual EPC and PIN associated with that tag can be sent). Actual reader-to-tag authentication needs to be carried out when reader wants to access (read/write) other sections of tag's memory bank. To do so, we can use PIN-based approach just like in the original Gen-2 RFID specification.

We also would like to note that, there exists another scheme that allows a reader to be able to decipher EPC without help from backend server for several sessions [7]. We have a different view in this regard. We believe that, in a ubiquitous environment, connectivity is abundant and exercising practical security and simplicity are a decisive fact to the successful adoption of new technology. In addition, we think that backend server's database can be partitioned in a hierarchical way, thereby reducing overhead at each backend server. This scenario naturally fits in both DNS-like hierarchical structure of EPCglobal Object Naming System (ONS) and real-life situations (for example, each department in a company manages its own inventories, thus, should have its own backend server). We want to stress that our proposed scheme is simple and provides reasonable security strength within the bound of the low-cost RFID tag's functionalities.

Notations. Before describing our protocol in detail, we give the definition of notations that we use in the description of our protocol.

Notation	*Interpretation*
$\mathcal{T}$	RFID Tag
$\mathcal{R}$	RFID Reader
$\mathcal{S}$	Backend Server
EPC	Electronic Product Code
$f(.)$	PRNG Function
$CRC(.)$	CRC Function
K_i	Session Key for i-th Session
PIN	Long-term Secret Shared between $\mathcal{T}$ and $\mathcal{S}$
r	A Pseudo-random Number
$\mathcal{A}$:	Action at entity $\mathcal{A}$
$\mathcal{A} \rightarrow \mathcal{B}$	Communication from $\mathcal{A}$ to $\mathcal{B}$
$\mathcal{A} \leftrightarrow \mathcal{B}$	Interaction between $\mathcal{A}$ and $\mathcal{B}$

Our Proposed Protocol. During the manufacturing time, manufacturer setups a tag by assigning EPC and other parameters. Then, it chooses a random seed number *seed* and store $K_1 = f(seed)$ to tag's memory and backend server's database entry corresponding to matching EPC. A random PIN (say, *access* PIN defined in Gen-2 specification) is also stored in both tag's memory and backend server database in a similar way. We also assume that the backend server is highly trustful.

The tag query protocol is as follows:

- $\mathcal{R} \rightarrow \mathcal{T}$: Query request.
- $\mathcal{T}$: Compute $M_1 = CRC(\text{EPC} \parallel r) \oplus K_i$ and $C = CRC(M_1 \oplus r)$ where r is a nonce.
- $\mathcal{T} \rightarrow \mathcal{R}$: M_1, C and r.
- $\mathcal{R} \leftrightarrow \mathcal{S}$: $\mathcal{R}$ and $\mathcal{S}$ authenticate each other and then $\mathcal{R}$ forwards M_1, C and r to $\mathcal{S}$.
- $\mathcal{S}$: For each tuple (EPC, K_i) in backend server's database, $\mathcal{S}$: verifies that $M_1 \oplus K_i$ equals $CRC(\text{EPC} \parallel r)$ and $C = CRC(M_1 \oplus r)$. If no tuple (EPC, K_i) is found, the tag is rejected. Otherwise, we assume that a tuple (EPC, K_i) is passed the check by $\mathcal{S}$:.
- $\mathcal{S} \rightarrow \mathcal{R}$: Detail information about object associated with the tag according to reader's privileges.

The above protocol can easily allow $\mathcal{R}$ to operate in batch mode, such that $\mathcal{R}$ collects multiple M_1's from various tags and send all to $\mathcal{S}$ at once.

If $\mathcal{R}$ desires to perform read/write operations to $\mathcal{T}$'s memory, it requests an authentication token M_2 from $\mathcal{S}$ where $M_2 = CRC(\text{EPC} \parallel \text{PIN} \parallel r) \oplus K_i$. Then, $\mathcal{R}$ sends M_2 to $\mathcal{T}$. $\mathcal{T}$ receives M_2 and computes its own version of M_2 based on its knowledge (of PIN, r, EPC and K_i). If two are not matched, $\mathcal{T}$ rejects $\mathcal{R}$'s request and accepts otherwise.

Lastly, when a session ends, $\mathcal{R}$ informs both $\mathcal{T}$ and $\mathcal{S}$ so that they can update their K_i for the next session by computing $K_{i+1} = f(K_i)$. Note that, this key updating process must be carried out perfectly. Otherwise, subsequent sessions will result in failure. To prevent other parties from either unintentionally or intentionally interfering key updating process (in case of intentional interference, we call it database desynchronization attack), $\mathcal{R}$ might need to inform the $\mathcal{T}$ and $\mathcal{S}$ with a one-time password. The password can be another PIN (*e.g.*, *kill* PIN as we don't need *kill* PIN ratified in Gen-2 specification). To make this password one-time, we needs to XOR it with a pseudo-random number, say r'. r' is sent to tags (together with 'Query Request' message) that reader is communicating with. The reader can signal 'End Session' to all communicating tags at once by broadcasting a single message $CRC(r' \oplus PIN')$.

4 Security and Complexity Analysis

In this section, we briefly give a security analysis of our proposed scheme. We claim that our scheme achieves the following security properties:

- *Tag-to-Reader authentication*: Without the knowledge of EPC and the session key K_i, a cloned tag might try to send a random message M_1, and hopes that it will be recognized by the backend server. However, the checksum $CRC(\text{EPC} \parallel r)$ forces the cloned tag to know the correspondence between EPC and the session key K_i. Furthermore, the other checksum, $CRC(M_1 \oplus r)$ also lowers the chance of fooling the backend server. By the way, the main purpose of $CRC(M_1 \oplus r)$ is to resolve possible collision on M_1 at backend server's database. According to birthday paradox, if the session key is k-bit long, the chance that there exists collision on M_1 at backend server's database is roughly $\frac{1}{2^{k/2}}$.
- *Reader-to-Tag authentication*: This type of authentication is delegated to Reader-to-Server authentication where we can make use of advanced authentication protocols in the cryptographic literature. Furthermore, a valid PIN is required if reader wants to access tag's memory (note that PIN is sent from reader to tag in a scrambled form).
- *Privacy protection*: Tag never directly emits EPC in a plaintext form. Each and every session, tag sends out a different bit string because of new session key and nonce r. Therefore, it is infeasible for malicious parties to use a compatible reader to track tag holder's activities, movement, belongings and preferences. Note that r is very important to provide privacy protection since the session key is updated only when a session ends successfully. Therefore, if an attacker interrogates a tag and end that session prematurely, without knowing the nonce r, the tag will emit the same M_1 in successive sessions.

The obvious way to improve security strength of our scheme is to employ cryptographic hash functions such as SHA-1 and MD-5 instead of CRC since the cryptographic hash function has much better anti-collision property. We can also use a cryptographic hash function for updating session key (*i.e.*, hash chain approach)

so that forward secrecy is guaranteed as cryptographic hash function is presumably one-way.

Our scheme also exhibits efficient computational complexity. We compare complexity and security features of our protocol with Juels' protocol in [3] which also targets Gen-2 RFID specification.

	Juels' Protocol	Our Protocol
Backend Server's Complexity	$O(N)$	$O(N)O(CRC)$
Tag's Complexity	$O(q)$	$1CRC + 2PRNG$
Reader's Complexity	$O(q)$	$O(1)$
Reader Authentication	◯	◯
Tag Authentication	◯	◯
Untraceability	×	◯

Note: N - the number of tags in tag population; $O(CRC)$ - computational complexity of CRC algorithm; q - the number of PIN-test round in Juels' protocol resulting in $1/2^q$ security margin; × - not supported; ◯ - supported; Complexity of authentication protocol between Tag and Server is not counted.

Regarding the implementation, our scheme requires extra memory space for session key K_i and possibly some more for tag's working memory. However, as specified in [2], a Gen-2 tag defines a spare memory bank for user-specific purposes. We think that K_i can be stored on that location so that no extra cost for storage would be incurred. The hardware to implement our protocol logic should also need minimal extra cost because the core functions required by our protocol including XOR, CRC and PRNG are already ratified in the Gen-2 specification.

5 Conclusion

We have presented a simple communication protocol for RFID devices, especially EPCglobal Class-1 Gen-2 RFID devices. Our protocol achieves desirable security features of a RFID system including: implicit reader-to-tag authentication, explicit tag-to-reader authentication, traffic encryption and privacy protection (against tracking). Our scheme makes use of only PRNG and CRC which are all ratified in current Gen-2 RFID specification. Moreover, there should be little overhead to adapt our protocol into the Gen-2 RFID specification.

Comparing to Juels' protocol, our suggested protocol offers more security features and better performance at the tag and reader sides. While Juels' protocol requires a tag and a reader to invoke q rounds of communication and PIN testing, our protocol has only one round of communication. We think that reducing computational and communication burden on the RFID tag is very crucial for the sake of the low-cost RFID tag. From security point of view, our scheme is also a more viable solution to the security threats than Juels' scheme. It is because Juels' scheme does not solve the privacy invasion issue which is considered to be the most serious problem faced by RFID technology.

For future work, computational complexity of our scheme at backend server side needs improvement. In addition, the ownership transfer of tag and multiple reading are not currently considered in our protocol.

Acknowledgement

This work was partially supported by a grant No.R12-2003-004-01004-0 from Ministry of Commerce, Industry and Energy.

References

[1] EPCglobal Inc., `http://www.epcglobalinc.org/`.

[2] EPCglobal Inc., "Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.09", Available at `http://www.epcglobalinc.org/standards_technology/specifications.html`.

[3] Ari Juels, "Strenthening EPC Tag against Cloning", *To Appear in the Proceedings of WiSe'05.*

[4] Ari Juels, "RFID Security and Privacy: A Research Survey", *To Appear in the Proceedings of IEEE JSAC'06.*

[5] D. Molnar, A. Soppera and D. Wagner, "A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of a RFID Tag ", *To Appear In the Proceedings of Selected Areas in Cryptography (SAC)'05.*

[6] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita, "Efficient Hash-Chain Based RFID Privacy Protection Scheme ", *In the Proceedings of International Conference on Ubiquitous Computing, Workshop Privacy*, September 2004.

[7] Gildas Avoine, Etienne Dysli, and Philippe Oechslin, "Reducing Time Complexity in RFID System", *To Appear In the Proceedings of Selected Areas in Cryptography (SAC)'05.*

[8] Jeongkyu Yang, "Security and Privacy on Authentication Protocol for Low-cost Radio Frequency Identification", Master Thesis, Available at `http://caislab.icu.ac.kr/Paper/thesis_files/2005/thesis_jkyang.pdf`, January 2005.

[9] Gildas Avoine and Philippe Oechslin, "A Scalable and Provably Secure Hash-Based RFID Protocol", *In the Proceedings of Workshop on Pervasive Computing and Communications Security - PerSec'05*, March 2005.

[10] NIST, "Random Number Generation and Testing", Available at `http://csrc.nist.gov/rng/`.

[11] RFID Journal, `http://www.rfidjournal.com/`.

[12] Tassos Dimitriou, "A Lightweight RFID Protocol to Protect against Traceability and Cloning Attacks", *In the Proceedings of SecureComm'05*, September 2005.

[13] Wikipedia, "Cyclic Redundancy Check", Available at http://en.wikipedia.org/wiki/Cyclic_redundancy_check.