A Provable Secure Batch Authentication Scheme for EPCGen2 Tags

Jiageng Chen, Atsuko Miyaji, and Chunhua Su

School of Information Science,
Japan Advanced Institute of Science and Technology, Japan.
{jg-chen, miyaji, su}@jaist.ac.jp

Abstract. EPC Class1 Gen2 (EPCGen2) is an international industrial standards for low cost RFID system used in many applications such as supply chain and consumer service. While RFID technology offers convenience and being employed in various applications in our society, security and privacy issues are still the number one concern of most RFID applications today. In this paper, we study the problems occurring where a reader wants to authenticate and identify legitimate RFID EPCGen2 tags in a batch to guarantee the integrity of the products. Most of the EPCGen2 tags are passive and have limited computational ability to compute cryptographic functions. For this reason, to design a mechanism to protect low-cost EPCGen2 tags from security and privacy risks is a challenging task. We propose a provable secure batch authentication scheme for EPCGen2 tags using the pseudo-random number generator (PRNG) and cyclic redundancy check (CRC) code. Our ultra-lightweight scheme which integrates the operations of EPCGen2 and only relies on build-in CRC-16 and PRNG function with secret keys inside the tags. We formally analyze security and privacy of the proposed scheme by using mathematical modeling and proof. Our analysis shows that our scheme provides strong ability to prevent existing possible attacks.

Keywords: RFID, batch authentication, pseudo-random number generators, security protocol

1 Introduction

RFID system is widely applied in counterfeiting products and RFID-enable supply chain in recent years. An RFID system consists of RFID tags, an RFID reader, and sometimes a back-end server. The communication channel between the reader and the backend server is (usually) assumed to be secure while the (wireless) channel between the reader and the tag is insecure. As the RFID reader communicates with the tags using RF signals, RFID protocols may face various security threats such as location privacy, authentication, and re-synchronization between read and tags. EPCGen2 standardization which covers the whole RFID architecture, from tag data structure to network communication specifications. EPC tags are not provided of on-board batteries, but are passively powered

through radio-frequency waves. The problem of authenticating tags in large-scale RFID systems can be easily reduced to verifying each tag one by one. However, this is not efficient enough for the practical usage for the extremely busy supply chain. A number of different authentication protocols have been proposed to address supply chain application.

There is scope for securing low cost devices. It is obviously that the level of security may not be sufficient for sensitive applications. However there are many low cost applications where there is no alternative in the practical industrial applications. It is difficult for them to adapt the existing authentication protocols using cryptographic primitives which require a lot of computation cost and storage space. Thus, we need to find a novel way to guarantee the security and privacy of such low-cost EPCGen2 tags.

The scenario of our scheme: We recognize that there are situations in which one has to design security into systems with restricted capability so as to promote low-cost widespread usage. In this paper, we concern with the security of universal EPCGen2 application. We focus on such a supply chain scenario in which a batch of products are transported from one place to another. The receivers who can be retailers or transportation service providers want to confirm that the products are the original ones and none of them is lost during the outsourcing supply chain procedures such as Third party logistics which is one of the most dominating kind of supply chains that has been widely adopted by many companies.

1.1 Related works

To execute the authentication while maintaining the security and privacy-preserving features in RFID system have been research for years. For the lightweight tags, there are also many proposals such as using one-way hash functions by Song *et al.* [**?**], performing authentication by hashing random challenges, tag identity, and/or secret key into one message [**?**], etc. However, hardware implementations of hash functions such as SHA-1 and MD5 are generally considered too expensive to be implemented on low-cost EPCGen2 RFID tags. For this reason, lightweight solutions are needed for low-cost RFID tags such as EPCGen2. Lightweight authentication protocols aim to achieve fast and cost-efficient authentication through simple operations like bitwise XOR and binary addition. In 2006, Juels and Weis proposed a multi-round lightweight authentication protocol called HB+ [**?**], after that many improvements of the HB+ protocol such as Peinado in 2007 [**?**] and Gilbert *et al.* in 2008 [**?**].

In 2009, Sun and Ting presented the EPCGen2 protocol [**?**] for EPCGen2 standard in which each tag stores a string and shared with a back-end server. Burmester *et al.* demonstrated an attack to break this protocol in 2009 [**?**]. Until recent years, it still remains a challenging task to design a reasonable secure and efficient solution for EPCGen2 application. Recently, there are some practical works focusing on the security of lightweight solutions for EPCGen2 tags,

such as pseudo-random number generator for EPCGen2 [**?**] and CRC-based solutions [**?**]. In recent years, there are some batch authentication methods for RFID being proposed, Yang *et al.* [**?**] and Guo *et al.* [**?**] study the RFID batch authentication issue and propose the first probabilistic approach to meet the requirement of prompt and reliable batch authentications in large scale RFID applications. However, their solutions are not light-weight to be used for EPCGen2 tags authentication. In 2014, Qi *et al.* [**?**] proposed a batch recall protocol for RFID-enable supply chain and industry manufacturers, there is a so-called collector to recall the products based on public key technologies. In practical implementation, the public key cryptography-based solutions are still too expensive to be broadly applied to low-cost RFID tags.

1.2 Our contributions and organization

The security level of EPCGen2 heavily depends only on 16-bits PRNG which makes RFID protocol potentially vulnerable up to a certain point, for example, adversary can perform ciphertext-only attacks to exhaust the 16-bit range of the components of protocol flows. Due to such a reason, we have to revisit the security and privacy issues for EPCGen2 tags and find a way to do better based on the limited computation resource of EPCGen2 tags.

- We first propose a provable secure scheme for batch authentication of EPCGen2 RFID. The main contribution of our paper is that our scheme can apply to EPCGen2 tags without modifying steps or components of the standard.
- Besides the efficiency for RFID authentication, our scheme uses very little computational and memory resource which includes one PRNG and one CRC along with a few conditional xors computation. The seed to be input to the PRNG can be considered as if it is a key for the blocks cipher; in particular, we forgo the need for key-separation for each tag.
- Different from related works of RFID authentication, we provide security property between reader and back-end server. Our scheme can prevent unauthorized reader to attack against RFID system.
- We provide security and privacy proof and analysis to show the limits of the adversary who tries to compromise our scheme.

The rest of the paper is organized as follows: Section 2 provides some preliminaries to understand the technical details in our proposal. We propose our provable secure batch authentication scheme in Section 3. We also provide the security and privacy analysis and proof of our proposed scheme in Section 4. We draw conclusions in Section 5.

2 Preliminaries

In this section, we give brief introduction of RFID system and EPCGen2 tags, providing syntax definitions and security primitives such as pseudo-random number generator and cyclic redundancy check code.

2.1 Brief description RFID system and of EPCGen2 tags

Our batch authentication relies on tag's internal PRNG and cyclic redundancy check.

- Passive Tag: RFID tags can be classified into two types, active or passive depending on powering technique. While an active tag can generate power by itself, a passive tag is not able to supply a power by itself. Therefore the passive tag obtains power from the reading devices when it is within range of some reading devices.
- Reader: A reader can read and re-write the data in a tag. A reader can also obtain the tags contents through queries. After the reader queries to a tag and receives some information from the tag, the reader forwards the information to a back-end server.
- Back-end server : A back-end server is a computer which manages and stores various information for authenticating of each tag, so as to determine a tags identity from the information of a tag sent by an authenticated reader. But if the reader can have enough memories and computational ability, the back-end server is not a must in a RFID system.

EPCGen2 was adopted as 18000-6 international Standard by ISO/IEC. As a result, RFID system will be able to be recognized without confusion. EPCGen2 tag has properties as follows [**?**]: Tag is passive and communication range is 2-10m and it has on-chip Pseudo-Random Number Generator (PRNG) and Cyclic Redundancy Code (CRC). It also has two 32-bit PIN for kill command for disable the tag and access command to write into the tag or to read something in password fields.

2.2 Security and privacy requirement for RFID batch authentication

Security for Tag authentication: Theoretically, RFID authentication is insecure if there exists a polynomial-time adversary such that one tag session on a legitimate tag output OK but had no matching conversation with any reader session, with non-negligible probability. That means the adversary can forge the tags and pass the authentication processing. RFID tags may contain sensitive information about the carrier in which the information should not be revealed to anyone, especially to an attacker. In other words, tags should first authenticate the reader validation before sending private data. Meanwhile, readers should also be able to authenticate tags to prevent counterfeit tags.

Tag Privacy: In a typical RFID system, when an RFID reader queries an RFID tag, it responds by sending its identifier to the reader; the reader can then request further details by sending this identifier to a server. If unauthorized readers can also get a tag identifier, then they may be able to determine the additional information related to the tag. For example, if the information associated with

a tag attached to a passport, ID-card or medical record could be obtained by any reader, then the damage would be very serious. To protect against such information leakage, RFID systems need to be controlled so that only authorized readers are able to access the information associated with a tag.

2.3 Mathematical definitions

Binary Fields: All the communication executed between reader and tags can be represented as an element of GF(2^n) as a polynomial over the field $GF(2)$ of degree less than n. The set $\{0,1\}^n$ of bit strings can be considered as the finite field GF(2^n) consisting of 2^n elements. A string $a_{n-1}a_{n-2}\cdots a_1a_0 \in \{0,1\}^n$ corresponds to the polynomial $a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0 \in GF(2^n)$. The addition in the field is just the addition of polynomials over GF(2) (that is, bitwise XOR, denoted by $\oplus$. To define multiplication in the field, we fix an irreducible polynomial f(x) of degree n over the field GF(2). Given two elements $a(x), b(x) \in GF(2^n)$, their product is defined as $a(x)b(x)$ mod $f(x)$-polynomial multiplication over the field $GF(2)$ reduced modulo $f(x)$. We simply write $a(x)b(x)$ and $a(x) \cdot b(x)$ to mean the product in the field $GF(2^n)$. We use the Arabic number to represent the polynomials. For example. "2" means x, "3" means $x+1$, and "7" means x^2+x+1. When we write multiplications such as $2 \cdot 3$ and 7^2, we mean those in the field GF(2^n).

Pseudo-random Number Generator: A pseudo-random number generator (PRNG) can be defined as an function for generating a sequence of numbers that approximates the properties of random numbers. A deterministic function $G : \{0,1\}^d \rightarrow \{0,1\}^m$ is a (t,ϵ) is a pseudo-random number generator (PRNG) if $d < m$, $G(x)$ and U_m (U_m is a m-bit truly random string) are (t,ϵ) indistinguishable. In this paper, we model the PRNG as a deterministic function $G() : \mathcal{K} \times \{0,1\}^{\mathrm{m}} \rightarrow \{0,1\}^{\mathrm{g}}$.

CRC code Cyclic Redundancy Check code (CRC) is a kind of calibration method that checks the correctness of data communication. In original usage of CRC, the sender sents m-bit information data represented as a polynomial $T(x)$, and the receiver receives the data as $D(x)$. For a given n, CRC code use a polynomial $g(x)$ as a generator, the sender moves $T(x)$ left to k bits, and then makes XOR operation with $g(x)$. The remainder is the check number $r(x)$. A CRC is called an n-bit CRC when its check value is n-bits. Such a polynomial has highest degree n, and hence $n+1$ terms (the polynomial has a length of $n+1$). The remainder has length n. The CRC has a name of the form $CRC-n$.

3 Our Batch Authentication Scheme

There are some essential requirements of a good batch authentication scheme for large-scale RFID systems. First, the authentication scheme should be efficient.

Second, the authentication result should be informative to support various application demands. Knowing that whether there exist counterfeits in a batch of tags or not is far from adequate, since the administrator of RFID systems may still resort to per-tag authentication to count how many counterfeits in a number of tags. Our protocol is designed for EPCGen2 RFID tags; therefore, the requirement for implementing our protocol will not overload the capabilities of the tags.

Our Proposal is inspired by the parallelized message authentication code [**?**] and online cipher design [**?**] in which a cipher taking input of arbitrary length and it can output ciphertext blocks as it is receiving the plaintext blocks. Specifically, the i-th ciphertext block should only depend on the key and the first i plaintext blocks. The system we built consists of EPCGen2 compliant RFID tags and an EPCGen2 compliant RFID reader. Our protocol are limited to computing an XOR sum of the RIFD internal processed data and using two extra CRC computation calls. The sketch of our batch authentication scheme can be referred to Fig 1.

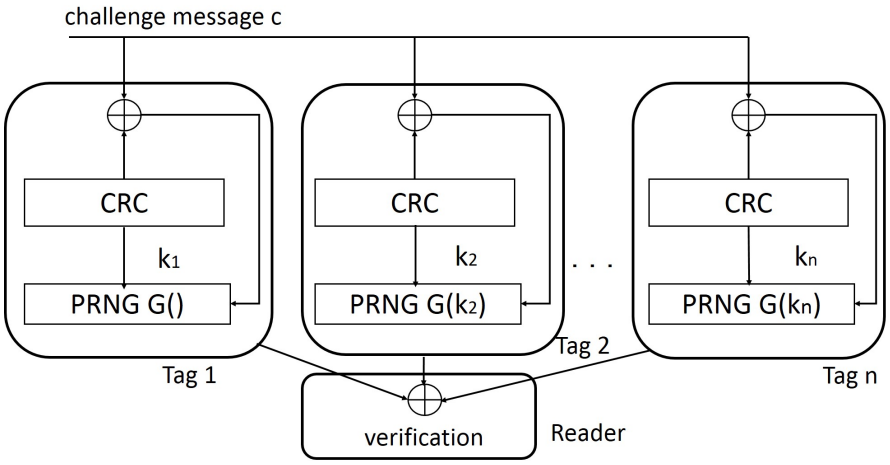

Fig. 1: Parallelized computation for the responds of tags

3.1 Initialization

The back-end server storages tags' identification information for the batch authentication, such as EPC for each tag. So that it can identify a tag from the information sent to an authenticated reader for tag. Reader generates different random challenge message and sends tags during each different session. We provide a table of notions which are used in our scheme as follows:

We consider an RFID system comprising of a single legitimate reader and a set of n tags $T = \{t_1, \cdots, t_n\}$, with some polynomials as security parameters.

Notions	Descriptions
t_i	i-th tag in a set of tags T
k_i	the secret key shared between reader and i-th tag
$\oplus$	exclusive-or operation
$\|\|$	concatenation of two inputs
$\Pr[A\|B]$	the probability of event A given B
G()	16-bit pseudo-random number generator
CRC()	cyclic redundancy check computation

Table 1. Notions used in our scheme

We assume that reader and each tag share a secret string for the authentication, here we denote it as k_i. Our scheme is based the parallelizable computation for each tag and allows reader to gather all the responds from tags for the batch authentication. We use the framework of XE and XEX constructions in [**?**,**?**], which makes our constructions and secure proofs simple and easy to be analyzed. For the secure communication and between reader and back-end server, we apply a block cipher $E()$ with shared key between server and reader as kr.

Typically, each tag is a passive transponder identified by a unique ID and has only limited memory which can be used to store only several keys and/or state information. The reader is composed of one or more transceivers and a backend processing subsystem. In this paper, we assume that the reader is secure, which means that an adversary cannot obtain any information about the RFID system from the legitimate reader except the information obtained from RFID communications and tags (in other words, the legitimate reader is a "black-boxh to an adversary).

3.2 Tag's internal processing

The internal processing of tag involves two function calls, one is for pseudo-random number generator and the other is for CRC computation.

For a particular construction PRNG $G()$ which consists of three major algorithms (setup, next, refresh)for the pseudo-random number generation and internal state update, we let $Pr[A(m,H)^{\mathrm{I(G)}} = 1]$ denote the probability that adversary $\mathcal{A}$ outputs the bit 1 after interacting as above with the system. Here I(G) stands for the ideal random process and note that we only use G in this game to answer queries that are made while the compromised flag is set to true. The details of the our PRNG are given as follows.

- setup: it is a probabilistic algorithm that outputs some parameters related to the secret key k_i of tag T_i for the generator.
- refresh: it is a deterministic algorithm that, given k_i of tag T_i, a state $ST \in \{0,1\}^n$ and an input $I \in \{0,1\}^p$, outputs a new state $ST' = \mathsf{refresh}(ST, I) = refresh(k_i, ST, I) \in \{0,1\}^n$

– next: it is a deterministic algorithm that, given k_i and a state $ST \in \{0,1\}^n$, outputs a pair $(ST'; R) == (k_i, ST)$ where $ST' \in \{0,1\}^n$ is the new state and a pseudo-random number $r \in \{0,1\}^g$ is the output.

According to EPCGen2 standard [**?**], the CRC-16 algorithm maps arbitrary length inputs onto 16-bit outputs as follows: an n-bit input p is first replaced by a binary polynomial $p(x)$ of degree $n-1$, and then reduced modulo a specific polynomial $g(x)$ of degree 16 to a polynomial remainder $r(x) : p(x) = q(x)g(x) + r(x)$. The remainder has degree less than 16 and corresponds to a 16-bit number. For EPCGen2, the polynomial $g(x)$ is the irreducible polynomial: $x^{16}+x^{12}+x^5+1$ (over the finite field GF(2) of two elements). CRC-16 will detect burst errors of 16-bits or less, any odd number of errors less than 16, and error patterns of length 2. For the $CRC()$ function, we use it to generate the dummy mask for individual tag. $CRC()$ is an efficient checksum algorithm and the input to $CRC()$ is divided into groups, each has 16 bits. Each 16-bit group will be encoded one by one. The output of each is a 16-bit encoded data. Each output will be combined together to generate the mask.

We assume a fixed, polynomial-size tag set $T = \{t_1, \cdots, t_n\}$ and a reader RD as the elements for an RFID system: $S = \{RD, T\}$. As to model the communication between tags and reader, we assume that the update process of new internal state and secret-key, by an uncorrupted tag in a session run, automatically overwrites (i.e., erases) its old internal state and secret-key to the PRNG. Each uncorrupted tag and reader use fresh and independent random coins (generated on the fly) in each session, in case it is an randomized algorithm enhanced by their internal PRNG. We assume that the random coins used in each session are erased once the session is completed (whether successfully finished or aborted).

3.3 Batch authentication

Formally, we consider an RFID system comprising of a single legitimate reader and a set of n tags $T = \{t_1, \cdots, t_n\}$, where l is a polynomial in a security parameter p. The reader and the tags can be modeled as probabilistic polynomial time interactive Turing machines. The RFID system (RD, T) is setup by a procedure, denoted Setup(p, l). This setup procedure generates the system parameter such internal keys for PRNG and the key for encryption. It may also setup an initial backend database DB for R to store necessary information for identifying and authenticating tags. We use $para = (w, k_1, ..., k_n)$ to denote the RFID system parameters. We assume that in the RFID system, the reader is secure; in other words, the legitimate reader is a "black-boxh to an adversary.

The reader collects all the responding massage from all tags and make aggregative computation to generate a authenticated massage and send it back to back-end server for further processing. The supply chain service providers and product manufacturers write a fix batch identification value w into each tag and using it to compute $CRC(w)$. In every tag, the value of w is different from each other. We use $CRC(w)$ to generate different mask value to avoid the collision in the batch authentication. The sketch our scheme is illustrated in Fig. 2. At each

new single session between reader and tags, let $S = G(0^{16})$ and the PRNG and CRC will be resumed to initial state.

Batch Authentication Protocol for RFID EPCGen2 Tags

- Reader $\rightarrow$ tags : The reader sends a challenge message c to all tag to initiate a session.
- Tags $t_i \rightarrow$ reader : Let $V[0] \leftarrow CRC(w) \oplus S)$, and $\Delta_0 \leftarrow 3 \cdot CRC(w)$ $\Delta_1 \leftarrow 2 \cdot CRC(w)$.
  After receiving the challenge message, for each tag $t_i \in T$ parse k_i with c runs its internal PRNG $a_i = G_i(c \oplus k_i)$.
  $V[i] \leftarrow G(a_i \oplus \Delta_0) \oplus V[i-1]$
  $R[i] \leftarrow G(V[i]) \oplus \Delta_1$
  $\Delta_0 \leftarrow 2\Delta_0, \Delta_1 \leftarrow 2\Delta_1$
  The tag t_i sends $R[i]$ to the reader.
- Reader $\rightarrow$ back-end server: The reader gathers all the message from the tags and computes $M = E_k(E_k(\Sigma \oplus 2^{d-1}3^2S) \oplus R) \oplus 2^{d-1}7S$ and sends M to the back-end server.
  Here, $R = R[1] \oplus R[2] \oplus \cdots \oplus R[n]$ and $\Sigma \stackrel{def}{=} k_1 \oplus k_2 \oplus \cdots \oplus k_n$.
- Server processing: The back-end server receives M from the reader who gathers all the response from tags. The server perform verification by checking if $R \oplus E_{kr}(\Sigma \oplus 2^{d-1}3^2S) = E_{kr}^{-1}(M \oplus 2^{d-1}7S)$, where the tag is rejected if the equality is not true. After that, it can recover the further information of the authenticated tags.

Fig. 2: Batch Authentication Scheme

The masking method in [**?**,**?**] enables us to produce many different values of the mask Δ from just one secret value $\Delta = CRC(w)$. Namely, the masks are produced as $\Delta = 2\alpha 3\beta 7\gamma \cdot R$ for varying indices of α, β and γ. To do this, we need to choose our irreducible polynomial $f(x)$ carefully. First, $f(x)$ needs to be primitive for the implementation of $CRC()$ inside of tags, and is able to generate the whole multiplicative group. Second, we make sure that $\log 2 \cdot 3$ and $\log 2 \cdot 7$ are both lager enough. Third, we check if $\log 2 \cdot 3$ and $\log 2 \cdot 7$ are defer enough (modulo 2^{n-1}). We impose these conditions to ensure that values $2\alpha 3\beta 7\gamma$ do not collide or become equal to 1. Combining CRC-16 with pseudo-random number generator can prevent the batch collision and tampering attacks effectively, it avoids occupying the resource of back-end server, and reduce the time complexity.

3.4 Post-authentication processing

By reconsidering the solution of batch authentication in another perspective, we find it is not always necessary to ensure the genuineness of every single product in a batch. It is acceptable if we guarantee the percentage of deactivated is sufficiently small.

Typically, the reader and the tag would exchange data after completing the authentication process. These data are sometimes considered private; for example, the tag used in a hospital would contain the records of its carrier. The threat of eavesdropping attacks makes the tag carriers feel insecure about transmitting sensitive data. To address this problem, we construct a mechanism to establish a session key and use it to encrypt the sensitive data. We suggest that reader and tags use the key stream generated by PRNG to encrypt the messages. Without the secret key the adversary cannot decrypt the message break the encrypted messages.

4 Security and Privacy Proofs of Tags Authentication

In this section, we provide the security and privacy proof for our batch authentication scheme. For the security analysis, we show the security bound for an adversary to forge the batch security. For the privacy analysis, we show that the adversary has limited advantage to distinguishable two communication session between reader and tags. This implicates that the adversary cannot do malicious tracing against the tags. Here, we model the adversary as a polynomial probabilistic Turing machine which tries to break the security and privacy of our batch authentication scheme.

The core security element is the PRNG which provides the minimum security property as follows which is shown in the existing result in [**?**,**?**]:

1. Probability of PRNG: The probability that a 16 bits pseudo-random number drawn from the PRNG has value r is bounded by: $0.8/2^{16} < \Pr(G(k_i) = r) < 1.25/2^{16}$.
2. Drawing identical sequences: For a tag population of up to 10,000 tags, the probability that any two or more tags simultaneously draw the same sequence of 16-bit pseudo-random number is $< 0.1\%$, regardless of when the tags are energized.
3. Next random number prediction: A random number which is generated by a tag PRNG is not predictable with probability better than 0.025%, given the outcomes of all prior draws.

There is an important point here is that the adversary cannot attack our scheme using the similar forging attack in message authentication codes using the arbitrary length of messages to get a collision. The number of the tags are determined pre-authentication. We prove our security and privacy as follows. The interaction between an adversary $\mathcal{A}$ and the protocol participants occurs only via oracle, which model the adversary capabilities in a real attack. During

the execution, the adversary may create several instances of a participant. Let U_i denote the instance i of a participant $U \in \{RD, T\}$

Adversary runs a $\texttt{Setup}(t_i, k_i)$ is a setup procedure which generates key k_i for a tag T_i and sets the tagfs initial internal state st_0. It also associates the tag T_i with its unique ID as well as other necessary information such as tag key and/or tag state information as a record in the database of reader.

4.1 The security of our scheme

In our analysis, we modify the security definition of our scheme from Rogaway *et al.* [**?**,**?**]. The security notions also can be found in [**?**]. For our batch authentication, the security refers to unforgablity of the aggregated tags which can pass the authentication. Let $\{0,1\}^n$ denote the set of strings whose length is a positive multiple of n bits. Here, we model the pseudo-random number generator as $G : \mathcal{K} \times \{0,1\}^* \to \{0,1\}^g$ is a function such that it is a permutation on every tag's n bits output, we can replace our PRNG calls with random permutations. Second, we show that the batch authentication behaves exactly the same as the ideal functionality for the security, as long as certain successful attacks from adversary do not occur. The successful attack here means collisions of tag state values, and the proof amounts to evaluating the probabilities of these attacks.

Let $\mathcal{A}$ be an adversary trying to distinguish $G()$ from the family all tweakable permutations with the same tweak space. Say that A runs in time t and makes exactly q queries. Without loss of generality assume that $\mathcal{A}$ is deterministic. Here, we define key space for PRNG as K, the adversary can make random sample from the key space $\mathcal{K}$ to attack the PRNG and forge tags response.

Definition 1. *We define the security using the advantage of an adversary $\mathcal{A}$.*

$$\mathbf{Adv}_G(\mathcal{A}) = \Pr\left[k \xleftarrow{\mathcal{R}} \mathcal{K} : \mathcal{A}^{G(k)} = 1\right] - \Pr\left[\pi \xleftarrow{\mathcal{R}} Perm(n) : \mathcal{A}^{\pi(\cdot)} = 1\right] \quad (1)$$

The above is the probability that adversary A outputs 1 when given an oracle for $G(k, \cdot)$, minus the probability that A outputs 1 when given an oracle for $\pi()$, where k is selected at random from K and π is selected at random from Perm(n).

If the adversary knows no additional information, the success probability is surely $1/2^g$. If the adversary acquires function for $CRC()$ by compromising a tag or a reader, it will have some advantages in constructing the codewords.

The adversary will try to generate the collision to get the same value of $V[j]$ coming from different secret value inside RFID tags $k_1, k_2, ..., k_n$ and $k'_1, k'_2, ..., k'_n$. At the beginning, the adversary chooses $k'_1, k'_2, ..., k'_n$ from the key space $\mathcal{K}$, and uses the PRNG oracle to generate the collision to pass the batch authentication. We can see that the outputs may be the same for a common mask for CRC-16. So an PRNG $G(k_i)$ yields a pseudo-random generator of $i - th$ tags, where the permutation is determined by the input (i.e. the secret key of tag k_i). Let $Perm(n)$ be the set of all such permutations to be distinguished from PRNG $G()$. We notice that unless the collision occurs, the adversary cannot distinguish an output of G from an output of Perm(n).

We want to model adversary's behavior to forge tags. Adversary show find some collisions in order to forge the targets tags without knowing their internal secret keys. Tagcoll(n) measures the probability of getting a collision when the adversary sends challenge message to n tags. The tag collision means a collision among the values $R[1], R[2], \cdots, R[n]$, where $R[0] = 0^n$ and each k_i is the PRNG input associated to tag i. Informally, CollisionM(n) measures the probability of finding collision at the finalized computation for readeracross two different batches of tags, T and T', each having n tags. This can be a "nontrivial" collision. That is, consider the $2n$ points at which the PRNG is applied in processing the finalized M and M'.

The adversary also can choose n elements at random key k_i' and w' and then there is the point to get the same output as the legal tags (the PRNG is applied at this point). There are n responses from original tags $R[1], \cdots, R[n]$, other responses from other faked tags $R'[1], \cdots, R'[n]$, Adversary can get some pairs of the collision of finalized tag could coincide for a "trivialh reason: namely, we know that $R[i] = R'[j]$ if $k_i' = k_j'$ and $M = M'$. We say that there is a nontrivial collision between T and T' if some other $R[i]$ and $R[j]$ happened to coincide. Note that M-collisions include collisions with 0^n, while CollisionM(n)s do not. Also, both collisions do not include collisions within a single tag (or collisions with all zero input) because both of these possibilities are taken care of by way of n-collisions.

We can further to apply the PMAC security proof for our scheme. We can make use of the theorem for parallel MAC construction in [**?**] and claim that for aggregated M from n tags, the adversary queries all n tags and then the advantage $\mathbf{Adv}_G(\mathcal{A})$ is less than $n^2/2^g$, here g is the output length of PRNG. For EPCGen2 tags, the advantage is less than $n^2/2^{16}$.

4.2 The privacy of our scheme

Adversary who try to beak the privacy of tags should execute as the following three phrases. The attack intentionally desynchronizes the tag from the reader by sending the tag some messages.

1. Learning: An adversary sends m number of queries Q_i for $1 \leq i \leq m$ to a batch@of targeting tags, and records the tagfs response $R[i]for1 \leq i \leq m$. Since the adversary is impersonating the reader, thus each time it will not pass the check by the tag, and so each time the tag would update its stored secret as $k_i = G(k_i)$, from which ti will be derived in the next session.
2. Challenge: Query m times to random tags in $\{t_1, \cdots t_n\}$ and obtain their response R and M
3. Guess: Check if $t = t_i$. If so, then the adversary knows this was the tag it queried during the learning phase i.e. $T_b = T$. Else, it knows that $T_b = T'$.

Intuitively, an adversary can trace location of tags if response of tag is always the same or similar pattern for each session. We can see that dummy masks generated by CRC is similar to the all XE an XEX construction can be modeled

using the techniques in. For each query from reader, tags' response are different even the challenge message c is the same. This property guarantees the privacy of our scheme. It is easy to prove that for individual tag, the privacy is well preserved.

More formally, we discuss the batch privacy which preserve the privacy for the whole batch of tags. We use $\mathbf{Adv}_G^{\mathcal{A}}(t, q, n, l)$ we denote the maximum advantage taken over all distinguishers that run in time t and make q queries, each of at most l out of totally n tags. Based on the security proof of [**?**], we can claim that for two pseudo-random function $f_1 : \{0,1\}^* \to \{0,1\}^g$ and $f_2 : \{0,1\}^* \to \{0,1\}^g$, the distinguishing advantage is at most $n^2/2^g$.

For the batch of RFID tags, adversary cannot distinguish from two queries if the collision does not occur. As same in the security analysis, adversary can access a PRNG oracle and use it to distinguish the queries. After get some priori knowledge by sending the query, the advantage of adversary can be bounded as $\mathbf{Adv}_G^{\mathcal{A}}(t, q, n, l) \leq \frac{39(n+q)^2}{2^g} + \mathbf{Adv}_{\mathrm{Perm}()}^{\mathcal{A}}(t, 4(n+q)) + \frac{(l+2)(q-1)^2}{2^g}$ according to the proof of [**?**].

5 Conclusion and future works

The EPCGen2 standardized tags focuses on reliability and efficiency while proving only a very basic security level, which is at risk of security and privacy breach. To overcome such risks is particularly challenging because the only security tool that is available in this standard is a 16-bit PRNG. In this paper, we proposed a scheme for EPCGen2 tags batch authentication which are provably secure. In this paper we have studied the recently proposed EPCGen2 related schemes and made some arguments on how to achieve maximum security and privacy levels supported by this standard. We proposed a batch authentication RFID protocol that provides strong anonymity and that complies with the EPCGen2 standard. Finally, we examine the successful probability for an adversary to forge a batch of tags and distinguish every responses in different session between reader and tags. In the future work, we want to extend our scheme to more sophisticated and practical scenarios, such as reader corruption, tag cloning (or more feasibly, protocols to prevent swapping attacks, tag group authentication, anonymizer-enabled RFID systems, and tag ownership transfer.

References

1. E. Andreeva, A. Bogdanov, A. Luykx, B. Mennink, E. Tischhauser and K. Yasuda."Parallelizable and Authenticated Online Ciphers". in Advances in Cryptology ASIACRYPT 2013 LNCS Vol. 8269, pp 424-443, 2013.
2. M. Burmester, B. de Medeiros, J. Munilla, and A. Peinado, "Secure EPCGen2 compliant radio frequency identification," Ad-Hoc, Mobile and Wireless Networks, vol. 5793, pp. 227-240, 2009.
3. J. Black and P. Rogaway. "A Block-Cipher Mode of Operation for Parallelizable Message Authentication". In Advances in Cryptology EUROCRYPT02, pp.384-397, 2002.

4. EPCglobal Inc., Class 1 Generation 2 UHF RFID protocol for communication at 860Mhz-960Mhz version 1.2.0, 2008.
5. Fleischmann, E., Forler, C., Lucks, S.: McOE: "A Family of Almost Foolproof On-Line Authenticated Encryption Schemes". In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 196?215. Springer, Heidelberg,2012.
6. W. Gong, K. Liu, X. Miao, Q. Ma, Z. Yang, and Y. Liu. "Informative counting: Fine-grained batch authentication for large-scale rfid systems". In ACM MobiHoc, 2013.
7. Gao, L., Ma, M., Shu, Y., Wei, Y. "An ultralightweight RFID authentication protocol with CRC and permutation". Journal of Network and Computer Applications, 2013.
8. H. Gilbert, M. Robshaw, and Y. Seurin, "HB#: increasing the security and efficiency of HB+," in Proceedings of the 27th International Conference on Theory and Applications of Cryptographic Techniques, pp. 361-378, 2008.
9. A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols". in Advances in Cryptology CRYPTO 2005, vol. 3621 of Lecture Notes in Computer Science, pp. 293-308, 2006.
10. J. Melia-Segui, J. Garcia-Alfaro, J. Herrera-Joancomarti. "Analysis and improvement of a pseudorandom number generator for EPC Gen2 tags". In Curtmola, R. et al. (Eds.), LNCS, Financial cryptography and data security 2010 workshops, pp. 34-46, 2012.
11. K. Mandal, X. Fan, and G. Gong, "Warbler. A Lightweight Pseudorandom Number Generator for EPC C1 Gen2 Passive RFID Tags", International Journal of RFID Security and Cryptography (IJRFIDSC), Vol. 2, Issue 1-4, pp. 82-91, 2013.
12. J. Munilla and A. Peinado, "HB-MP: a further step in the HB-family of lightweight authentication protocols," Computer Networks, vol. 51, no. 9, pp. 2262-2267, 2007.
13. P. Rogaway. "Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC". In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp.16-31. Springer, Heidelberg, 2004.
14. D. Ranasinghe, D. Engels, and P. Cole, "Low-cost RFID systems: confronting security and privacy," in Proceedings of the Auto-ID Labs ResearchWorkshop, pp.54-77, 2004.
15. Rogaway, P., Zhang, H. "Online Ciphers from Tweakable Blockciphers". In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 237-249. Springer, Heidelberg, 2011.
16. B. Song and C. J. Mitchell, "RFID authentication protocol for low-cost tags," in Proceedings of the 1st ACM Conference on Wireless Network Security, pp. 140-147, April 2008.
17. H. M. Sun and W. C. Ting, "Gen2-based RFID authentication protocol for security and privacy", IEEE Transactions on Mobile Computing, vol.8, no.8, pp.1052-1062, 2009.
18. L. Yang, J. Han, Y. Qi, Y. Liu, "Identification-free batch authentication for RFID tags", Proceedings of the 18th IEEE International Conference on Network Protocols, p.154-163, October, 2010.
19. S. Qi, Y. Zheng, M. Li, L. Lu, Y. Liu, "COLLECTOR: A Secure RFID-Enabled Batch Recall Protocol", In IEEE INFOCOM, Canada, April 2014.