CG+ RFID Authentication Protocol Revisited

Fereidoun Moradi*[1] and Hamid Mala[2]
[1] *School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden*
[2] *School of Computer Engineering, University of Isfahan, Isfahan, Iran*
fereidoun.moradi@mdh.se
h.mala@eng.ui.ac.ir
(This manuscript is under review.)

Abstract. RFID system utilizes radio frequency to transmit information among Tags and Readers which lets adversaries to effortlessly eavesdrop the information over the wireless channel. In this regard, several authentication protocols have been proposed with their focus on lightweight computations while preserving strong security. In 2012, a lightweight RFID authentication protocol suitable for VANETs was introduced by Caballero-Gil et al. They acclaimed the scheme is robust against security and privacy attacks, but then in 2015 and 2019, two information leakage vulnerabilities discovered on the protocol. Afterwards, researchers fixed weaknesses of the protocol and proposed immune versions, whereas two CG+ and CG++ redesigned schemes are results of their endeavors. In this study, we point out that CG+ scheme is still insecure against a full disclosure attack with the time complexity of $O(2^{16})$. We verify the correctness of the presented attack through Python implementation and then by eliminating the shortcomings, we present a modified scheme. Finally, we intuitively represent that the revised protocol preserves required security along with the feasible computational and communication overhead compared to previous proposals.

Keywords: RFID · Mutual authentication · Disclosure attack.

1 Introduction

In current smart world, Vehicular Ad hoc NETworks (VANETs) have attracted increasing attentions from both academia and industry fields. With the extensive VANETs deployment in transportation systems, driving experience can be drastically improved. In vehicular communication environments, Radio Frequency Identification (RFID) system is typically used where the tag is resided on the vehicle and the reader is located on the road. Through VANET, road safety is improved and road traffic is optimized. Regarding the ever-increasing diversity of attacks on VANETs, it is essential to make sure that life-critical information can not be illegally inserted or modified by an adversary, and the privacy of the drivers and passengers should be preserved [1–3]. The EPC

Class-1 Generation-2 standard [4] defines a strict framework including low-cost resided tag functions and operations. This standard describes the allowable operations on tags which are restricted to some simple operations such as Cyclic Redundancy check Code (CRC), Pseudo Random Number Generator ($PRNG$), and bitwise XOR [5]. Hence, to enhance the security of an RFID system and reduce the complexity, it is necessary to propose an efficient authentication protocol regarding to EPC-C1G2 standard specification [6].

In 2012, Caballero-Gil et al. [7] described a new lightweight scheme compliant with EPC-C1G2 standard for mutual authentication between a reader and tags that fulfills resource limitation of tags and minimal interaction between parties. Furthermore, the proposal is based on the trust on the back-end server because all shared secrets are maintained only by the tag and the back-end server. Afterward, in 2015, Moradi et al. [8] showed that the scheme is vulnerable to de-synchronization attack and suffers from the information leakage. They redesigned the authentication protocol and named CG+ protocol. In 2019, Lijun Gao [9] analyzed CG+ protocol and showed that the protocol is not immune against the replay and key guessing attack. He improved the scheme through applying multiple merging operations and introduced CG++ as a modified protocol version with the acceptable computational and communication overhead.

In this paper, we show an efficient disclosure attack against CG+ protocol [8] where an adversary passively reveal all secret parameters. The contribution shows that the CG+ has a drawback that is still insecure against full secret disclosure attack which discloses all secrets of the protocol. The main time complexity of this attack is about 2^{16} offline $PRNG$ evaluations which can be easily afforded by an ordinary adversary. Then, by removing the CG+ flaws we propose a new version which provides significant security compared to its two predecessors. Furthermore, the comparison result shows the performance attribute of our modification is much better than CG++ scheme.

Paper Organization The rest of this paper is organized as follows. We first review some existing RFID authentication protocols in Section 2. Next, in Section 3 the preliminaries of CG+ and its review are represented. The disclosure attack against CG+ is presented in Section 4. In Section 5 we describe the improved protocol and investigate its security. Finally, the paper is concluded in Section 6.

2 Related Work

To address the security and privacy concerns in lightweight RFID applications several solutions have been proposed in the literature. We discuss some of the schemes as follows.

Pang et al. [10] proposed a novel RFID authentication protocol that claimed their protocol does not have weaknesses and effectively resist the tag information leakage. Soon after, researchers in [11, 12] noted the protocol contains several flaws and all secret values can be revealed by 2^{17} $PRNG$ evaluations. Afterward,

authors in [13] exploited an efficient way to toggle one bit of the transmitted random value and presented a de-synchronization attack against the revised version of the protocol.

Habibi et al. [14] pointed out that all past and next transactions of a compromised tag will be traceable by an adversary who disclosed all secrets. Sun et al. [15] proposed an attack scenario where the adversary abuses the process of the key updating to break the protocol. Later, Mujahid et al. [16] proposed a new cryptographic primitive (pseudo-kasami code) to be applied in the RFID authentication process. The primitive enhances the diffusion properties of the protocol messages and makes the hamming weight of the secrets unpredictable. However, Safkhani et al. [17] reported on a vulnerability found in the protocol and presented a desynchronization attack against the protocol. Afterwards, they have also identified disclosure attack in two mutual authentication protocols of IoT system [18].

In 2017, Guo et al. [19] presented a privacy-preserving authorized RFID authentication protocol. Regarding the approach, the designated readers are located in a place where each tag performs the consistent time calculation for mutual authentication. They mathematically analyzed security and privacy of the work and demonstrated the tags cannot be traced, even if tags are corrupted by adversaries. In 2018, Gope et al. [20] studied a lightweight authentication protocol for distributed IoT applications. The proposed protocol not only protects the forward security and anonymity, but also hide the location of RFID embedded devices. However, they used a combination of XOR and complex hash functions which can not be used for passive tags. In 2019, Fan et al. [21] provided a lightweight authentication scheme for cloud-based RFID healthcare systems. The scheme is based on quadratic reSIDuals [22] and a pseudo random number generator, which meets the privacy and security requirements of the cloud-based RFID system with fewer resources.

3 Review of CG+

Here, we briefly describe CG+. The notations used in CG+ have been represented in Table I. The designers of this protocol have supposed that each tag T_i in this protocol keeps a record including $\{ID_{T_i}, SSK_{T_i}\}$ and the server also keeps the tag corresponding pair, including both the new and the old ones in its database. It is also assumed that both reader and tag are able to use a secure16-bit pseudo-random number generator $PRNG$. The protocol has two phases: registration and authentication. In the registration phase, in database of server, the old and the new version of variables are set to the same value, $ID_{T_i}^{old} = ID_{T_i}^{new}$, $SSK_{T_i}^{old}$=$SSK_{T_i}^{new}$, and after registration, the parties can mutually communicate with each other. The protocol is described step by step as follows.

1. First, the reader chooses a random seed s to initialize $PRNG$ in order to produce the 16-bit value N_1, and sends it to tag T_i.

TABLE I. Notations

Notations	Description
T_i:	i^{th} RFID tag.
R_j:	j^{th} RFID reader.
$s, s_1 \text{or} s_2$:	A 16-bit seed chosen by the reader.
N_1:	A 16-bit value built by the PRNG-function of reader.
N_2:	A 16-bit random number generated by the tag.
ID_{T_i}:	The 16-bit identity of tag T_i.
SSK_{T_i}:	The 16-bit secret key shared between the server and tag T_i.
K:	The shared session key.
$\oplus$:	Bit-wise XOR operation.
$B \leftarrow A$:	To assign the value of A to B.
$PRNG$:	The pseudo-random number generator with 16-bit output length.

2. Upon receiving N_1, the tag generates a random number N_2, computes A and B, and sends these values to the reader.
3. Then, the reader receives A and B, and sends them along with N_1 to the server.
4. The server receives A, B and N_1, and does as follows.
   - For any entry in database, the server extracts N_2 from $B \oplus ID_{T_i}^{new}$ and computes $A' \leftarrow PRNG(ID_{T_i}^{x} \oplus \ N_1) \oplus PRNG(SSK_{T_i}^{x} \oplus N_2)$, where $x \in \{old, new\}$ and then compares it with the received value A to identify and authenticate the tag T_i. If the tag is successfully authenticated, the server sends $SSK_{T_i} \oplus PRNG(N_2)$ to the reader.
   - Afterward, the server updates corresponding record in database, because of successful authentication of tag T_i.
5. After receiving $SSK_{T_i} \oplus PRNG(N_2)$, the reader does XOR operation between the seed originally chosen by itself in step 1 and the received message from the server, then sends $C \leftarrow s \oplus SSK_{T_i}^{x} \oplus PRNG(N_2)$ to the tag.
6. The tag receives C and extracts s by XORing C and $SSK_{T_i} \oplus PRNG(N_2)$, then checks whether it corresponds to the initially received N_1. After successful authentication on the server, the tag updates its records.

Finally, both the reader and the tag can generate the same secret session key K of length 16, through the XOR operation between the s chosen by the reader and the message A sent by the tag, $K \leftarrow s \oplus A$.

4 Secret Disclosure Attack Against CG+

The authentication protocols should meet the confidentiality requirement. In the RFID context, it means that a protocol must leak no information to an unauthorized party and this concept is related to limiting access to secret values of tags such as identity and cryptographic keys. Also, it is indicated that $PRNG$ of the protocol must satisfy the statistical requirements such as good distribution and collision resistance that are imposed by the EPC Gen2 standard. Most $PRNGs$ are build on algorithms involving some kind of recursive

method starting from a base value that is determined by an input seed. By the way, we observe that the CG+ confidentiality can be violated by the following description.

The main observation which is the milestone of our attack is the step 2 where the public random number over the insecure channel is computed through a 16-bit $PRNG$ with only a short 16-bit seed. Hence, an adversary can easily pre-compute a dictionary of all possible values of X in $PRNG(X) = Y$, where X and Y are 16-bit values, and sort this dictionary with respect to the Y values. Then, upon observing, for example, any value $N_1 = PRNG(s)$, the adversary finds s by only one access to the dictionary. In the following, it is pointed out that the CG+ still cannot resist against the disclosure attack with a complexity of 2×2^{16}. This attack consists of two phases and its scenario has been depicted in Algorithm 1.

Learning phase //online operations
1 Eavesdrops the all values of one session, N_1, A, B, C;
Attack phase //offline operations
2 **For** $i \leftarrow 0$ *to* $2^{16} - 1$ **do**;
3 $s \leftarrow i$;
4 **IF** $N_1 = PRTNG\,(s)$ **then**
5 **End for**
6 Return i; //Return i as s value
7 **For** $j \leftarrow 0$ *to* $2^{16} - 1$ **do**
8 $N_2 \leftarrow j$;
9 $ID_{T_i} \leftarrow N_2 \oplus B$;
10 $SSK_{T_i} \leftarrow C \oplus PRNG(N_2) \oplus s$;
11 **IF** $A = PRNG(ID_{T_i} \oplus N_1) \oplus PRNG(SSK_{T_i} \oplus N_2)$ **then**
12 **End for**
13 Return j; //Return j as N_2 value
14 $ID_{T_i} \leftarrow N_2 \oplus B$; //Extract the ID_{T_i} value
15 $SSK_{T_i} \leftarrow C \oplus PRNG(N_2) \oplus s$; //Extract the SSK_{T_i} value

---

Algorithm 1. The Disclosure Attack Against CG+

Learning Phase: In this phase of the attack, the adversary waits until a legal transaction of the protocol begins and the reader transmits N_1 to T_i.

1. The adversary eavesdrops and stores the all messages transmitted in public, including:
   N_1,
   $A = PRNG(ID_{T_i} \oplus N_1) \oplus PRNG(SSK_{T_i} \oplus N_2)$,
   $B = N_2 \oplus ID_{T_i}$,
   $C = s \oplus SSK^x_{T_i} \oplus PRNG(N_2)$.

Secret Recovery Phase: In this phase of the attack, the adversary uses the eavesdropped values and does operations as below:

1. Refers to the row with index N_1 of the pre-computed dictionary, as for example depicted in Table II, to obtain the value s.

TABLE II. A typical 16-bit dictionary used in the proposed attack

$PRNG(x)$	$Index = x$
0000	568D
0001	44F3
0002	DF84
...	...
FFFF	F18C

2. *For* $j = 0, \ldots, 2^{16} - 1$ the adversary does as follows:
   $N_2 \leftarrow j$,
   $ID_{T_i} \leftarrow N_2 \oplus B$,
   $SSK_{T_i} \leftarrow C \oplus PRNG(N_2) \oplus s$.
   If $A = PRNG(ID_{T_i} \oplus N_1) \oplus PRNG(SSK_{T_i} \oplus N_2)$ holds, then the adversary returns j as N_2.

3. Now, it applies the following operations to reveal ID_{T_i} and SSK_{T_i}.
   $ID_{T_i} \leftarrow N_2 \oplus B$,
   $SSK_{T_i} \leftarrow C \oplus PRNG(N_2) \oplus s$.

In the proposed attack, the tag's all secret parameters are extracted. Given secret parameters, it is easy to apply any other attacks against the protocol. From the data complexity point of view, the attacker requires to eavesdrop only one session of the protocol and the time complexity is 2×2^{16} offline $PRNG$ evaluations in the secret recovery phase. Hence, the exact security level of CG+ is only $O(2^{16})$, the same as its predecessor Caballero-Gil et al. scheme. We present simple Python implementation of the attack in [23] to verify the correctness of our attack. The values are 16-bit digits and $CGpluse(L)$ is defined for simulating protocol behavior. The $PRNG$ is based on a hash function and initialized with a seed, then produces a 16-bit pseudo-random number.

5 Improving CG+

In this section, we aim to make the resulting protocol immune against the attack described in the previous section. The main flaw of this kind of protocols is the $PRNG$ function with short length input. The natural solution could be increasing the size of $PRNG$ function, e.g. using a 64-bit $PRNG$ or replacing 16-bit $PRNG$ functions with lightweight block ciphers such as SIMON and SPECK [24]. However, the current EPC standard does not support such functions and we are limited to the current cryptographic primitives available in the EPC standard including 16-bit CRC and 16-bit $PRNG$. Therefore, to eliminate this weakness, the constructions of some messages are revised and then we discuss why the revised protocol is secure. The brief description of these changes are presented as follows.

– In the initiation phase, EPC_s as a more secret variable has been stored in the database and tags, to increase security when XOR operation is done on messages B.

– The first and the last messages have been changed, such that the adversary cannot do exhaustive search on them. Two seeds are chosen in calculation of N_1 and the value C is replaced by the two equations $E \leftarrow s_1 \oplus C$ and $F \leftarrow s_2 \oplus D$.

5.1 The Revised Protocol

In this subsection, we present and clarify the revised version of CG+, which is secure against the mentioned attack and other attacks in the context. In the revised protocol we use the same notations in Table I, and only one identifier is added. The revised protocol supposes that each tag T_i keeps a pair $\{ID_{T_i},\ SSK_{T_i}\}$ and $EPC_{s_{T_i}}$ in its memory where $EPC_{s_{T_i}}$ is 96-bit EPC code that is divided into six 16-bit blocks and then these six blocks are XORed.

$$EPC_{s_{T_i}} \leftarrow EPC_{b_0} \dots EPC_{b_{15}} \oplus EPC_{b_{16}} \dots EPC_{b_{31}} \oplus EPC_{b_{32}} \dots EPC_{b_{47}} \oplus EPC_{b_{48}} \dots EPC_{b_{63}} \oplus EPC_{b_{64}} \dots EPC_{b_{79}} \oplus EPC_{b_{80}} \dots EPC_{b_{95}}$$

The server also keeps a record of data for each tag T_i including $ID^{old}_{T_i}$, $SSK^{old}_{T_i}$, $ID^{new}_{T_i}$, $SSK^{new}_{T_i}$, $EPC_{s_{T_i}}$. The revised mutual authentication protocol is described step by step as follows.

1. The reader begins a session, chooses two random seeds s_1 and s_2 in order to calculate the 16-bit value $N_1 \leftarrow PRNG\,(s_1) \oplus PRNG(s_2)$, and sends it to tag T_i.
2. Once the tag received N_1, it generates its random number N_2, computes A and B as bellow and sends them to the reader.
   $A \leftarrow PRNG\,(ID_{T_i} \oplus\ N_1) \oplus PRNG\,(SSK_{T_i} \oplus\ N_2)$,
   $B \leftarrow N_2 \oplus ID_{T_i} \oplus EPC_{s_{T_i}}$
3. The reader receives A and B, and sends them along with N_1 to the server.
4. After receiving A, B and N_1, the server does as below.
   (a) Looks up its database for any entry in database.
   (b) Obtains $N_2 \leftarrow B \oplus ID^x_{T_i} \oplus EPC_{s_{T_i}}$ and computes $A' \leftarrow PRNG(ID^x_{T_i} \oplus N_1) \oplus PRNG(SSK^x_{T_i} \oplus N_2)$ then checks whether $A = A'$ holds or not.
   (c) Repeats the search until the matched tag is found. If it cannot find a matched tag, the session aborts.
   (d) After successful authentication of tag the T_i, it computes and sends $C \leftarrow SSK^x_{T_i} \oplus\ PRNG(N_2)$ and $D \leftarrow ID^x_{T_i} \oplus\ PRNG(N_2)$ to the reader and then updates its parameters.
5. Finally, after receiving C and D, the reader computes E and F as $E \leftarrow s_1 \oplus C$ and $F \leftarrow s_2 \oplus D$, and sends them to tag T_i.
6. The tag extracts s'_1 and s'_2 via $E \oplus SSK^x_{T_i} \oplus\ PRNG(N_2)$ and $F \oplus ID^x_{T_i} \oplus PRNG(N_2)$. Then, it generates $N'_1 \leftarrow PRNG\,(s_1) \oplus PRNG(s_2)$ and compares N'_1 with the received value N_1. If it holds, the tag authenticates the server successfully, then updates the records.
   $ID_{T_i} \leftarrow PRNG(ID_{T_i})$
   $SSK_{T_i} \leftarrow PRNG(SSK_{T_i})$

At the end of the execution of these steps, the secret session key K of length 16 can be generated through the XOR operation between s_1, s_2 and the message A sent by the tag, $K = s_1 \oplus s_2 \oplus A$.

5.2 Security Evaluation the Revised Protocol

Our revision on CG+ improves its security and eliminates the information leakage weakness. In the following, we give a discussion on why the revised protocol is secure.

Information leakage prevention. Revising the calculation of N_1 and C fixes the main flaw of CG+. The reason is the following observation. Given PRNG(X) and PRNG(Y), for X and Y are not equal, one needs $O(2^{16})$ off-line PRNG evaluations to determine X and Y, while given PRNG(X) $\oplus$PRNG(Y) it is not possible to determine X and Y uniquely and after $O(2^{16})$ PRNG evaluations, we encounter 2^{16} possible values for each of X and Y. In the revised protocol, each session has six secrets including $\{s_1, s_2, N_2, ID_{T_i}, SSK_{T_i}, EPC_{s_{T_i}}\}$.

To apply the proposed attack scenario on this scheme, the attacker just needs to guess three unknown values out of the aforementioned secret parameters. The complexity of this attack is eavesdropping of one session of the protocol, and then running a new session by impersonating the reader. But the latter step requires $O(2^{32})$ off-line computations which is the optimal security bound for EPC-C1G2 compliant tag. This attack has been depicted in Algorithm 2 and its source code of this attack can be found in [23].

The complexity of the attack is not lower than the claimed security level of the EPC-C1G2 standard. Therefore it cannot threaten the security of the improved protocol. We proceed the security analysis of the improved protocol based on different attack strategies e.g. tag/reader impersonation, traceability and de-synchronization which have been introduced in the context of RFID authentication protocols.

Tag Impersonation Attack Prevention. Integrity is a requirement that should be satisfied by an RFID authentication protocol. This attribute assures the originality of transmitted data and guarantees that it is not manipulated in transition, either randomly or by hostile activity. In these protocols integrity can be violated by some attacks such as replay attack. To impersonate the tag, the adversary must successfully generate a valid A and B. However, these messages are produced based on unknown parameters where $SSK_{T_i}$and ID_{T_i} are updated after each successful session and in each run N_2 is chosen randomly. So it is impossible for the adversary to deceive the reader through replay attack.

Reader Impersonation Attack Prevention. Similar to tag impersonation attack, to impersonate the reader, the adversary should return valid values E and F. Replay attack does not work and the best strategy for the adversary to impersonate the reader could be sending a random value to the tag. However, since the reader has to produce two records for sending, the adversary's success probability in each try is bounded by 2^{-32}.

De-synchronization Attack Prevention. The most dangerous threat is a de-synchronization attack, in which an adversary tries to make inconsistency

Learning phase //online operations
Eavesdrops the all values of one session, N_1, A, B, E, F and stops the last massages of it;
Sends N_1 to the tag; //Start new session
Eavesdrops the values of stared session, N_1, A', B';
Attack Phase //offline operations
$\Delta A \leftarrow A \oplus A' \leftarrow PRNG(SSK_{T_i} \oplus\ N_2) \oplus PRNG(SSK_{T_i} \oplus\ N_2')$;
$\Delta N_2 \leftarrow B \oplus B' \leftarrow N_2 \oplus N_2'$;
For $i \leftarrow 0$ *to* $2^{16} - 1$ **do**
    $SSK_{T_i} \oplus\ N_2 \leftarrow i$;
    $SSK_{T_i} \oplus\ N_2' \leftarrow i \oplus \Delta N_2$;
    IF $\Delta A = PRNG(SSK_{T_i} \oplus\ N_2) \oplus PRNG(SSK_{T_i} \oplus\ N_2')$ **then**
    End for
Return i; //Return i as $SSK_{T_i} \oplus N_2$
$L \leftarrow s_1 \oplus N_2 \oplus\ PRNG\,(N_2) \leftarrow i \oplus E$; //Named L combination of $i \oplus E$;
For $j \leftarrow 0$ *to* $2^{16} - 1$ **do**
    For $k \leftarrow 0$ *to* $2^{16} - 1$ **do**
        $s_1 \leftarrow j$;
        $N_2 \leftarrow k$;
        IF $L = s_1 \oplus N_2 \oplus\ PRNG\,(N_2)$ **then**
        End for
    End for
Return j and k; //Return j as s_1 and k as N_2
$SSK_{T_i} \leftarrow E \oplus s_1 \oplus\ PRNG\,(N_2)$; //Extract the SSK_{T_i} value
For $l \leftarrow 0$ *to* $2^{16} - 1$ **do**
    $s_2 \leftarrow l$;
    IF $N_1 = PRNG\,(s_1) \oplus PRNG\,(s_2)$ **then**
    End for
Return l; //Return l as the s_2 value
$ID_{T_i} \leftarrow F \oplus s_2 \oplus PRNG\,(N_2)$; //Extract the $\mathrm{ID_{T_i}}$ value
$EPC_{s_{T_i}} \leftarrow N_2 \oplus B \oplus ID_{T_i}$; //Extract the $EPC_{s_{T_i}}$ value

Algorithm 2. The Possible Attack Against the Revised Scheme

between a legitimate tag and the server. Thus, the target tag cannot be identified in subsequent sessions and will be unavailable. Our revised scheme provides resistance against de-synchronization attack. Since the server keeps the value of the old and new pair $\{ID_{T_i},\ SSK_{T_i}\}$ in its storage, de-synchronization of the tag and the server does not happen. Even if the adversary interferes communication to cause synchronization problem, the two-state update procedure for each tag keeps, keeps the tags synchronized with the server. Hence, to de-synchronize a specific tag, the adversary should either impersonate the reader or change the last message from the reader to the tag such that the tag authenticates the reader.

Traceability Attack Prevention. The tag's tracking attack consists of tracking of the behavior of the owner of the tag. The adversary can track the tag whose response information remains invariant in all transmissions. In the revised protocol the response of the tags will be random in each session because a fresh

random number is used each time. Thus the adversary does not know which tag the response belongs to. Therefore, it is not possible to trace the tag.

5.3 Performance Comparison

We compare the revised scheme with its predecessors based on the performance. Note that computational cost of these protocols includes $PRNG$ function and XOR operation. Let n stands for the total number of tags stored in backend database of server, l denotes the bit length of parameters, and m indicates a given count of random numbers. Thus, the overhead of a successful updating and mutual identification procedure of these protocols are illustrated in Table III.

TABLE III. Performance comparison between the revised protocol and its predecessors

		Num. of PRNG	Num. of XOR	Num. of Stored bits	Num. of Transferred. bits
CG [7]	Server	$n+2$	n	$2l$	l
	Reader	1	1	l	$4l$
	Tag	4	2	$2l$	l
CG+ [8]	Server	$2n+3$	$3n+1$	$4l$	l
	Reader	1	1	l	$5l$
	Tag	6	6	$2l$	$2l$
CG++ [9]	Server	$m+2$	$m+1$	$4l$	$2l$
	Reader	1	1	l	xl
	Tag	m	m	$2l$	$ml+1$
This Paper	Server	$2n+3$	$5n+2$	$5l$	$2l$
	Reader	2	3	$2l$	$5l$
	Tag	6	10	$3l$	$3l$

Consider computational and storage restrictions lay on the tag side, while the back-end server and the reader do not face these restrictions due to the presence of powerful processors and amount of physical storage. Table III implies that on the server side of CG and CG+ protocols, tag secret key can be found at most through calling $n+2$ $PRNG$ and $2n+3$ XOR operations. Our proposed modification has same $PRNG$ calculation process as CG+ scheme but, CG++ protocol performs $m+2$ calculation where m is raised up if a specific random value is found. On the tag side of CG++ protocol, the authors approach is generating m number of random values which effects authentication process time. Whereas, this paper scheme increases calling number of XOR and $PRNG$ functions a few times on the tag side. Comparing to CG and CG+ protocols, these amount of calling is affordable by low-cost tags. In addition, the modification does not extensively affect enhancement of the stored and transferred bits but provides more security in return. Thus, the result shows the revised scheme is suitable for RFID applications.

6 Conclusions

In this paper, we considered the security of an improved EPC-C1G2 compatible authentication protocol, which has recently been presented as CG+ [8]. We proved that this scheme like its predecessor, Caballero-Gil et al. authentication protocol [7], suffers from information leakage vulnerability. We presented a secret disclosure attack that can retrieve all secret parameters related to a given tag with the complexity of $O(2^{16})$. The complexity of this attack is one session eavesdropping and only 2×2^{16} offline $PRNG$ evaluations. After the successful cryptanalysis of CG+, we revised it to rectify its security flaws. Our detailed security analysis demonstrated significant security improvement for the new scheme, which may be considered as a step toward the security of PRNG-based EPC-C1G2 compliant authentication protocols.

Acknowledgment

This research is supported by Swedish Foundation for Strategic Research (SSF) via the Serendipity project.

References

1. D. Carluccio, K. Lemke-Rust, C. Paar, and A.-R. Sadeghi, "E-passport: the global traceability or how to feel like a ups package," in *International Workshop on Information Security Applications*, pp. 391–404, Springer, 2006.
2. S. Konomi and G. Roussos, "Ubiquitous computing in the real world: lessons learnt from large scale rfid deployments," *Personal and Ubiquitous Computing*, vol. 11, no. 7, pp. 507–521, 2007.
3. V. D. Hunt, A. Puglia, and M. Puglia, *RFID: a guide to radio frequency identification*. John Wiley & Sons, 2007.
4. E. Global, "Epc radio-frequency identity protocols class-1 generation-2 uhf rfid protocol for communications at 860 mhz–960 mhz," *Version*, vol. 1, no. 0, p. 23, 2008.
5. D. N. Duc, J. Park, H. Lee, and K. Kim, "Enhancing security of epcglobal gen-2 rfid tag against traceability and cloning," in *SCIS 2006*, pp. 97–97, Institute of Electronics, Information and Communication Engineers, 2006.
6. J. Shen, H. Tan, Y. Ren, Q. Liu, and B. Wang, "A practical rfid grouping authentication protocol in multiple-tag arrangement with adequate security assurance," in *2016 18th international conference on advanced communication technology (ICACT)*, pp. 693–699, IEEE, 2016.
7. C. Caballero-Gil, P. Caballero-Gil, A. Peinado-Domínguez, and J. Molina-Gil, "Lightweight authentication for rfid used in vanets," in *International Conference on Computer Aided Systems Theory*, pp. 493–500, Springer, 2011.
8. F. Moradi, H. Mala, and B. T. Ladani, "Security analysis and strengthening of an rfid lightweight authentication protocol suitable for vanets," *Wireless Personal Communications*, vol. 83, no. 4, pp. 2607–2621, 2015.
9. L. Gao, "Security analysis and improvement on cg+ protocol," *Wireless Personal Communications*, vol. 107, no. 1, pp. 695–705, 2019.

10. L. Pang, L. He, Q. Pei, and Y. Wang, "Secure and efficient mutual authentication protocol for rfid conforming to the epc c-1 g-2 standard," in *2013 IEEE Wireless communications and networking conference (WCNC)*, pp. 1870–1875, IEEE, 2013.
11. S. Wang, S. Liu, and D. Chen, "Security analysis and improvement on two rfid authentication protocols," *Wireless Personal Communications*, vol. 82, no. 1, pp. 21–33, 2015.
12. M. Safkhani, M. Hosseinzadeh, M. E. Namin, S. Rostampour, and N. Bagheri, "On the (im) possibility of receiving security beyond 2 l using an l-bit prng," *Wireless Personal Communications*, vol. 92, no. 4, pp. 1591–1597, 2017.
13. F. Moradi, H. Mala, and B. T. Ladani, "Cryptanalysis and strengthening of srp+ protocol," in *2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, pp. 91–97, IEEE, 2015.
14. M. H. Habibi, M. Gardeshi, and M. R. Alaghband, "Practical attacks on a rfid authentication protocol conforming to epc c-1 g-2 standard," *arXiv preprint arXiv:1102.0763*, 2011.
15. H.-M. Sun, W.-C. Ting, and K.-H. Wang, "On the security of chien's ultralightweight rfid authentication protocol," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 2, pp. 315–317, 2009.
16. U. Mujahid, M. Najam-ul Islam, and S. Sarwar, "A new ultralightweight rfid authentication protocol for passive low cost tags: Kmap," *Wireless Personal Communications*, vol. 94, no. 3, pp. 725–744, 2017.
17. M. Safkhani and N. Bagheri, "Generalized desynchronization attack on umap: Application to rcia, kmap, slap and sasi+ protocols.," *IACR Cryptology ePrint Archive*, vol. 2016, p. 905, 2016.
18. M. Safkhani and M. Shariat, "Implementation of secret disclosure attack against two iot lightweight authentication protocols," *The Journal of Supercomputing*, vol. 74, no. 11, pp. 6220–6235, 2018.
19. F. Guo, Y. Mu, W. Susilo, and V. Varadharajan, "Privacy-preserving mutual authentication in rfid with designated readers," *Wireless Personal Communications*, vol. 96, no. 3, pp. 4819–4845, 2017.
20. P. Gope, R. Amin, S. H. Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving rfid authentication scheme for distributed iot infrastructure with secure localization services for smart city environment," *Future Generation Computer Systems*, vol. 83, pp. 629–637, 2018.
21. K. Fan, S. Zhu, K. Zhang, H. Li, and Y. Yang, "A lightweight authentication scheme for cloud-based rfid healthcare systems," *IEEE Network*, vol. 33, no. 2, pp. 44–49, 2019.
22. R. Doss, S. Sundaresan, and W. Zhou, "A practical quadratic residues based scheme for authentication and privacy in mobile rfid systems," *Ad Hoc Networks*, vol. 11, no. 1, pp. 383–396, 2013.
23. F. Moradi, "Passive disclosure attack code repository," in *https://github.com/fereidoun-moradi/DisclosureAttack*, IEEE, 2019.
24. R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, "The simon and speck lightweight block ciphers," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, IEEE, 2015.