ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2015, T. 44, Nr. 1

On the Designing of EPC C1 G2 Authentication Protocols using AKARI-1 and AKARI-2 PRNGs

Masoumeh Safkhani

Computer Engineering Department
Shahid Rajaee Teacher Training University
Lavizan, Tehran, Iran, Postal code: 1678815811
e-mail: Safkhani@srttu.edu

Fatemeh Baghernejad

Electrical Engineering Department
Shahid Rajaee Teacher Training University
Lavizan, Tehran, Iran, Postal code: 1678815811

Nasour Bagheri

Electrical Engineering Department
Shahid Rajaee Teacher Training University
Lavizan,Tehran, Iran, Postal code: 1678815811
e-mail: NBagheri@srttu.edu

crossref http://dx.doi.org/10.5755/j01.itc.44.1.5883

Abstract. Chen *et al.* have recently proposed a mutual authentication scheme for RFID compliant EPCglobal Class 1 Generation 2 standard (or in brief EPC C1 G2) and claimed that their protocol can provide immunity against usual attacks same as replay attack, traceability attack and secret disclosure attack. However, in this paper we prove that unfortunately these claims do not hold. For this purpose, we present a tag impersonation attack, a server impersonation attack and a traceability attack against Chen *et al.* protocol. The success probability of tag impersonation and server impersonation attacks is 1 while the complexity of them is only two runs of the protocol. The success probability of traceability attack is "$1-\frac{1}{2^n}$" where *n* is the bit length of parameters in the protocol and the complexity is only two runs of the protocol. In addition, we propose an improved protocol exploiting lightweight PRNGs same as AKARI-1 and AKARI-2. We also prove both using formal and informal methods that our scheme solves its predecessor weaknesses and is resistant against the attacks considered in this paper and the other known active and passive attacks. In this paper, we choose BAN logic as our formal proof method. Our formal and informal security analysis of the improved protocol shows that it has better security level than its predecessors.

Keywords: RFID; EPC C1 G2; authentication; AKARI-1 and 2; traceability; impersonation.

1. Introduction

A set of technologies in which radio waves are used for automatic identification, tracking and management of humans and objects is called radio frequency identification or in short RFID. The functionality of RFID depends on the tags, the readers and the back-end servers. An RFID tag is a simple chip which is located on the object to be identified; RFID reader can communicate with all kinds of tags which are active, passive and semi passive and read or modify the tags information. Back-end server helps to the reader by saving extra information about tags or sometimes with doing some complex operations for tag's identification and authentication.

In EPC C1 G2 based protocols, only pseudorandom number generator (PRNG) and cyclic redundancy code (CRC) operations are used to provide authentication in which all of these operations have lower security level than the other operations such as hash functions or encryption algorithms. So, the compiling of protocols messages exchange scenario has the main role to provide security goals for the protocol. It should be noted that many EPC C1 G2 compliant protocols have been proposed in the literature [5, 6, 11, 13, 15, 18, 20, 27] aim to meet authentication purposes but there are reports on vulnerability of most of these proposed schemes [5, 9, 18, 24].

The confidentiality, integrity and availability (or in brief CIA) triad is one of the basic principles of information security. Preventing the disclosure of secret information to illegal users is interpreted to the confidentiality principle. Maintaining and ensuring the correctness of data during their transmission over unsecure channel is interpreted to integrity principle. The system services must be available whenever they are needed which is interpreted to availability principle [16]. The basic security requirement of an RFID EPC C1 G2 conforming authentication scheme similar to other security systems is providing CIA principles. To provide confidentiality, a secure RFID protocol must resist against attacks such as tracking attacks and secret information disclosure attacks. In the traceability attack, the adversary traces the target tag by linking between the transmitted messages in wireless channel and identity of the target tag or directly by retrieving the identity of the tag. In the secret information disclosure attack, the adversary can find the secret information of the target tag and so can apply all of other attacks on it. To provide integrity principle, a secure RFID protocol must resist against attacks such as impersonation attacks. In the impersonation attack, the adversary forges a legitimate entity for other legitimate entities which tag impersonation, reader impersonation and back-end data base impersonation attacks are its examples. And finally, to provide availability principle, a secure RFID protocol must resist against desynchronization attacks. In the desynchronization attack, the adversary forces legitimate entities to update their critical information to different values. Therefore, they exit from synchronization state and cannot authenticate each other in further transactions.

In 2007, Chien *et al.* [14] proposed a mutual authentication protocol which is based on EPC C1 G2 standard. They used cyclic redundancy code (CRC) calculations and random nonces in their protocol. However, cryptanalysis of Chien *et al.*'s scheme by Peris-Lopez *et al.* [26], shows that it suffers from tag impersonation attack and back-end database impersonation attack and cannot provide forward secrecy. Qingling *et al.* [6] proposed a protocol for RFID systems based on EPC C1 G2 standard. They claimed that their protocol is secure against the spoofing attack, the replay attack and tracking attack. However, cryptanalysis of Qingling *et al.*'s protocol by Burmester *et al.* [18] shows that it suffers from replay attack and reader impersonation attack. Chen and Deng [5] proposed a mutual authentication protocol conforming to EPC C1 G2 based on CRC and PRNG functions and claimed that the proposed protocol is secure against all known attacks against RFID systems. However, attackers can perform their attacks by exploiting the linear property of the CRC function. Peris-Lopez *et al.* [24] proved that Chen and Deng's protocol is vulnerable to reader impersonation attack and tag impersonation attack and cannot provide untraceability. Peris-Lopez *et al.* [24] proposed an EPCbased protocol, named Azumi, and claimed that their protocol offers a better security level than Chen and Deng's protocol and it resists against replay attack and traceability attack. However, cryptanalysis of the scheme by Safkhani *et al.* [22] shows that Azumi protocol is vulnerable to secrets disclosure attack and tag impersonation attack. Recently, Chen *et al.* [4] have proposed a novel EPC C1 G2 compliant scheme for RFID systems and claimed that their protocol resists against replay attack, man-in-the-middle attack and traceability attack. However, in this paper we show that their protocol is vulnerable to tag impersonation attack, server impersonation attack and traceability attack. Finally, in this paper we propose some solutions to address Chen *et al.* protocol and formally and informally prove their resistance against the attacks presented in this paper and the other known active and passive attacks.

Throughout the paper, we use notations shown in Table 1 which have also been used in Chen *et al.* protocol.

Adversary Model. The model of adversary used in this paper is an active man in the middle adversary who can eavesdrop, modify and intercept the messages being exchanged between protocols parties.

Security Proof Method. We use BAN logic [19] to prove the security correctness of our proposed protocol. We also use informal method to show that the proposed protocol provides suitable security and privacy for RFID systems.

Paper Organization. In Section 2, Chen *et al.* protocol is described. Security analysis of Chen *et al.* protocol including tag impersonation attack, server impersonation attack and traceability attack is explained in Section 3. In Section 4, we present an improved version of the protocol. Its formal and informal proofs of immunity against the attacks presented in Section 3 and the other known active and passive attacks are presented in Section 5. In Section 6, we provide a performance analysis of the improved protocol compared to the known protocols. Finally, Section 7 concludes the paper.

2. Chen *et al.* Protocol

Chen *et al.* protocol, which is depicted in Fig. 1, runs as below:

1. The reader generates a random number N_1, computes $A = CRC(N_1)$ and then sends M_{req} and A to the tag.

2. Once the tag received M_{req} and A, it does as follows:

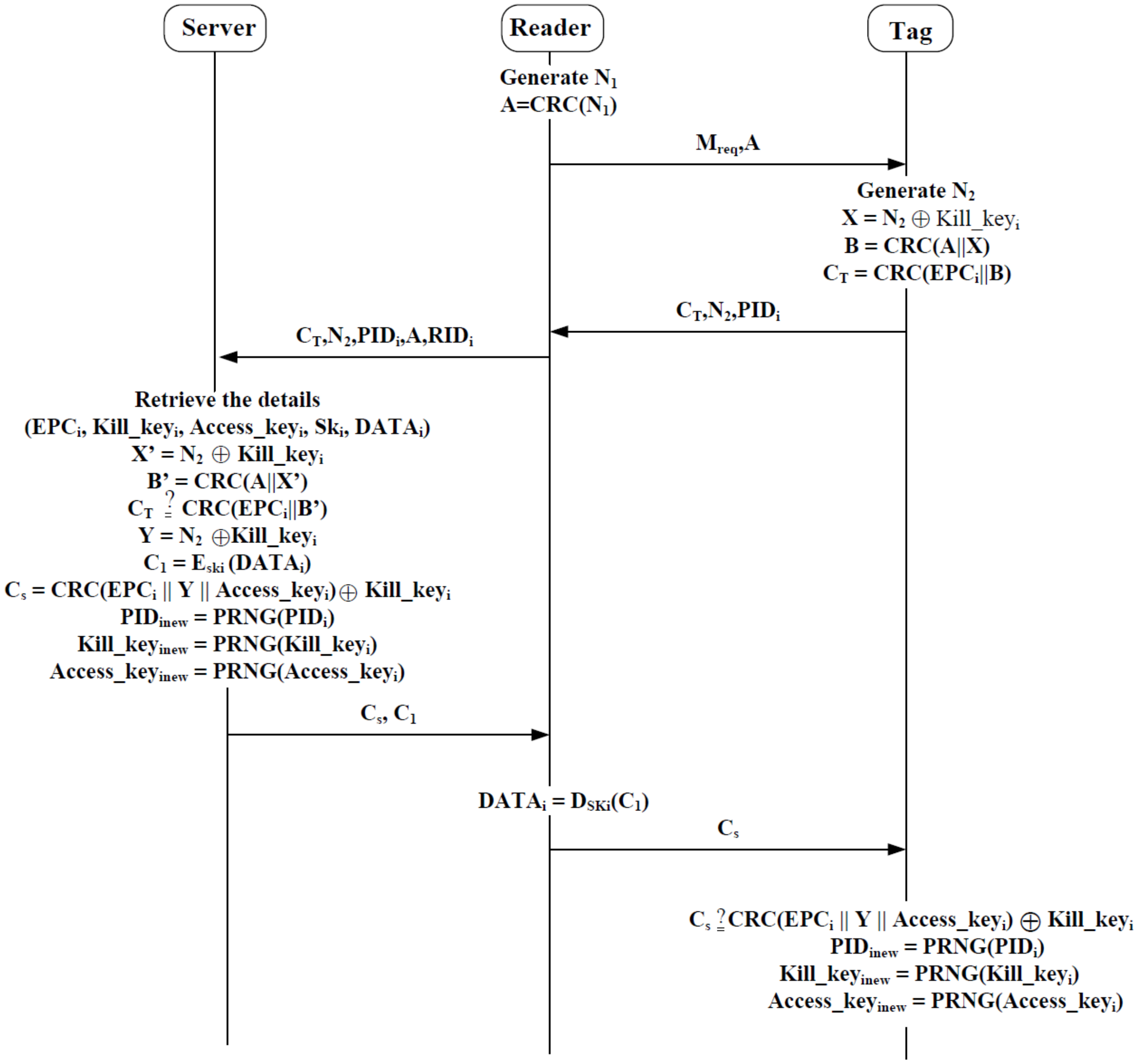

Figure 1. Chen *et al.* protocol [4]

- generates another random number N_2,
- computes:

  $X = N_2 \oplus Kill\_key_i$,

  $B = CRC(A||X)$,

  $C_T = CRC(EPC_i||B)$,

- and responds (C_T, N_2, PID_i) to the reader.

3. Once the reader received (C_T, N_2, PID_i), then it sends $(C_T, N_2, PID_i, A, RID_i)$ to the server.

4. Once the server received $(C_T, N_2, PID_i, A, RID_i)$, it does as follows:

- checks PID_i and RID_i; if they are correct, then the server retrieves (EPC_i, $Kill\_key_i$, $Access\_key_i$, Sk_i, $DATA_i$),
- computes:

  $X' = N_2 \oplus Kill\_key_i$,

  $B' = CRC(A||X')$,

- then the server verifies whether $C_T \overset{?}{=} CRC(EPC_i||B')$; if it is correct, it:

  – computes Y, C_1 and C_s as follows:

  $Y = N_2 \oplus Kill\_key_i$,

  $C_1 = E_{SKi}(DATA_i)$,

  $C_s = CRC(EPC_i \| Y \| Access\_key_i) \oplus Kill\_key_i$,

  – updates PID_i , $Kill\_key_i$ and $Access\_key_i$ as follows:

  $PID_{i_{new}} = PRNG(PID_i)$,

  $Kill\_key_{i_{new}} = PRNG(Kill\_key_i)$,

  $Access\_key_{i_{new}} = PRNG(Access\_key_i)$,

  – and sends (C_s, C_1) to the reader.

5. Once the reader received (C_s, C_1), it forwards C_s to the tag and extracts $DATA_i$ as $DATA_i = D_{SKi}(C_1)$.

6. Once the tag received (C_s), it does as follows:

- checks whether $C_s \overset{?}{=} CRC\left(EPC_i \| Y \| Access_{key_i}\right) \oplus Kill\_key_i$ and in the case of equality, computes the following values:

$$PID_{i_{new}} = PRNG(PID_i),$$
$$Kill\_key_{i_{new}} = PRNG(Kill\_key_i),$$
$$Access\_key_{i_{new}} = PRNG(Access\_key_i).$$

Table 1. Notation

Notation	Description
M_{req}	The request message
M_{resp}	The response message
N	Random number which is used once
PID_i	The pseudonym of the *i^th^* tag
RID_i	The identity of the *i^th^* reader
SK_i	The *i^th^* session key shared between the server and the reader
$E_{Ski}(m)$	Encryption of the message m with the key of SK_i
$D_{Ski}(m)$	Decryption of the message m with the key of SK_i
EPC_i	Electronic product code of the *i^th^* tag which has 96 bits length
$CRC(x)$	Cyclic redundancy check operation
$Kill\_key_i$	Kill key of the *i^th^* tag which is used in disabling tag according to EPC C1 G2 standard
$Access\_key_i$	Access key of the *i^th^* tag which is used in writing data on the tag's memory according to EPC C1 G2 standard
$PRNG$	Pseudo random number generator
$DATA_i$	The information of the *i^th^* tag
$P \lhd MSG1$	P receives $MSG1$
$P\|{\sim}MSG1$	P sends $MSG1$
$\#(X)$	X is fresh
$P\| \equiv \#(\ MSG1)$	P believes the freshness of $MSG1$
$\{X\}_K$	Message X is encrypted with the key of K
$P\| \equiv P \overset{K}{\longleftrightarrow} Q$	P believes that the secret K is shared between P and Q
$P2:\ \frac{P\|\equiv P\overset{K}{\longleftrightarrow}Q,P\lhd\{X\}_K}{P\|\equiv Q\|\sim X}$	The message meaning rule of BAN logic that means if P believes that it shares a secret key K with Q and if P receives a message X encrypted with K, then P is entitled to believe that Q once said X. In this paper, we called this rule $P1$
$P2:\ \frac{P\|\equiv Q\sim\{X,Y\}}{P\|\equiv Q\|\sim\{X\}}$	This is one rule of BAN logic that means if P believes that Q has sent $\{X,Y\}$, then P is entitled to believe that Q has sent X. In this paper, we called this rule $P2$
$\|X\|$	Bit length of string X

3. Security Analysis of Chen *et al.* Protocol

In this section, exploiting linear property of CRC function, we describe how Chen *et al.* protocol is vulnerable against tag impersonation attack, server impersonation attack and traceability attack. It must be noted that Chen *et al.* protocol is vulnerable against trivial desynchronization attack which occurs by stopping the last message of protocol from reader to tag. As a result of this attacker's operation, the server updates its PID_i, access and kill passwords while the tag does not update its corresponding values, so they exit from synchronism and cannot authenticate each other anymore. Success probability of this attack is 1 while its complexity is only one run of the protocol. To defend against such trivial desynchronization attack, the server in Chen et al. protocol must store the old and new pairs of PID_i , $Kill\_key_i$ and $Access\_key_i$.

3.1. Tag Impersonation Attack

In tag impersonation attack, the adversary convinces the legitimate reader to be a legitimate tag. Chen *et al.* have claimed that their protocol provides resistance against tag impersonation attack. However, we show that unfortunately their claim does not hold.

The main idea of this attack is based on the following observation:

Observation 1. According to the protocol description, one can state:

$$X = N_2 \oplus Kill\_key_i,$$
$$B = CRC(A||X) = CRC(A.x^n \oplus X) = CRC(A.x^n) \oplus CRC(X) = CRC(A.x^n \oplus N_2 \oplus$$

$Kill\_key_i) = CRC(A.x^n) \oplus CRC(N_2) \oplus CRC(Kill\_key_i)$

$C_T = CRC(EPC_i||B)$

$= CRC(EPC_i.x^n) \oplus CRC(B)$

$= CRC(EPC_i.x^n) \oplus CRC(CRC(A.x^n \oplus N_2 \oplus Kill\_key_i))$

$= CRC(EPC_i.x^n) \oplus CRC^2(A.x^n) \oplus CRC^2(N_2) \oplus CRC^2(Kill\_key_i)$

$= \chi \oplus CRC^2(A.x^n)$

$\chi = C_T \oplus CRC^2(A.x^n)$

In our tag impersonation attack, the adversary is an active man-in-the-middle adversary who works as below:

First Session:

In this session, the adversary supplants the reader.

1. The adversary supplants the legitimate reader and sends M_{req} and A to the tag.

2. Once receipt, the tag sends (C_T, N_2, PID_i) to the reader which is supplanted by the adversary.

Second Session:

In this session, the adversary supplants the tag.

1. The reader sends M'_{req} and A' to the tag which is supplanted by the adversary.

2. According to the Observation 1, the adversary can extract χ from C_T, therefore she calculates C'_T as below:

$C'_T = CRC(EPC_i||B')$

$C'_T = \chi \oplus CRC^2(A'.x^n)$

3. The adversary sends (C'_T, N_2, PID_i) to the reader.

Hence, the adversary is authenticated as a legitimate tag with probability of 1 and the complexity of attack is only two runs of the protocol.

3.2. Server Impersonation Attack

In server impersonation attack, the tag and the reader authenticate the adversary as a legitimate server. Chen *et al.* have claimed that their protocol provides resistance against server impersonation attack. However, we prove that a man-in-the-middle adversary can perform her server impersonation attack with only two runs of the protocol as below:

First session:

1. The adversary waits until the reader starts a session and sends M_{req} and A to the tag.

2. Upon reception, the tag responds (C_T, N_2, PID_i) to the reader.

3. The reader sends $(C_T, N_2, PID_i, A, RID_i)$ to the server.

4. The server sends (C_s, C_1) to the reader.

5. The reader forwards C_s to the tag.

6. The adversary blocks C_s and sends a random stream to the tag.

7. The tag terminates the current session and does not update its records of secret information.

Second session:

1. The adversary supplants the server and sends M_{req} and A to the tag.

2. The tag responds (C'_T, N'_2, PID_i) to the adversary.

3. According to what was mentioned in Section 2, the adversary calculates C'_s as $C'_s = CRC(N_2 \oplus N'_2) \oplus C_s$ because: $C_s \oplus C'_s = CRC((Y \oplus Y')||(Access\_key_i \oplus Access\_key'_i)) \oplus Kill\_key_i \oplus Kill\_key'_i$.

We have $Access\_key_i = Access\_key'_i$ and $Kill\_key_i = Kill\_key'_i$, so we can deduce: $C_s \oplus C'_s = CRC(Y \oplus Y')$

$= CRC(N_2 \oplus Kill\_key_i \oplus N'_2 \oplus Kill\_key'_i)$

$= CRC(N_2 \oplus N'_2)$

$C'_s = CRC(N_2 \oplus N'_2) \oplus C_s$

Hence, the tag authenticates the adversary as a legitimate server with probability of 1 and the complexity of attack is only two runs of the protocol.

3.3. Traceability Attack

In traceability attack, the adversary uses linkage of the tag's responses in different runs of protocol in order to distinguish a target tag among other tags. Chen *et al.* have claimed that their protocol resists against traceability attack. However, in this subsection we describe a traceability attack on Chen *et al.* protocol. In this attack, the adversary is a man-in-the-middle adversary. The attack consists of three phases, learning phase, execution phase and decision phase as follows:

Learning phase: In this phase of the attack, the adversary gathers the necessary information about the target tag, *i.e.* T_1:

1. The reader sends M_{req} and A to T_1,

2. T1 responds (C_T, N_2, PID_i) to the reader.

3. The adversary listens to the channel and eavesdrops (C_T, N_2, PID_i).

Execution phase: In this phase of the attack, given a tag T', the adversary does as below:

1. The adversary sends message (M_{req}, A) to the tag T' (before the target tag updates its information).

2. The tag T' responds message (C'_T, N'_2, PID_i) to the adversary.

Decision phase: In this phase of the attack, the adversary determines whether the tag T' is the target tag T_1:

If C_T and C'_T come from the same tag, we have:

$C_T \oplus C'_T = CRC(B \oplus B')$

$= CRC(CRC(A||X) \oplus CRC(A||X'))$

$= CRC^2(A.x^n \oplus X \oplus A.x^n \oplus X')$

$= CRC^2(X \oplus X')$

$= CRC^2(N_2 \oplus Kill\_key_i \oplus N_2' \oplus Kill\_key_i)$

$C_T \oplus C_T' = CRC^2(N_2 \oplus N_2')$

The adversary calculates $(C_T \oplus C_T')$ and compares it with the above result. If the tag T_1 and the tag T' are the same, the probability of satisfying the above equation is 1 and if the tag T_1 and the tag T' are different, the probability of satisfying the above equation is "$\frac{1}{2^n}$" where n is the bit length of C_T and C_T'. Therefore, the success probability of traceability attack is calculated as follows:

$\Pr[Avr|_{traceability-attack}] = \Pr[(T_1 = T') \wedge$ (satisfying the above equation)] $- \Pr[(T_1 \neq T') \wedge$ (satisfying the above equation)]

$\Pr\left[Avr|_{traceability-attack}\right] = 1 - \frac{1}{2^n}$.

Therefore, the adversary has been succeeded in tracking T_1 with the probability of "$1 - \frac{1}{2^n}$" while the complexity of attack is two runs of the protocol.

4. Improved Protocol

In this section, we present an improved version of Chen *et al.* protocol. In the improved protocol, to avoid the linear property of CRC function, we use a lightweight PRNG function instead of CRC function in calculation of C_T and C_s messages of the protocol. In practice, the user can employ AKARI-1 or AKARI-2 [12] as the target PRNG. On the other hand, a PRNG with 32 bit or less than 32 bit input/output length is vulnerable against PRNG input discovery attacks. In PRNG input discovery attacks, the attacker can retrieve the input of PRNG by exhaustive search [21]. Therefore, in the improved protocol we use AKARI-1 or AKARI-2 with 64 bit input/ output length.

Also, in the improved protocol, we assume that the length of $Kill\_key_i$ and $Access\_key_i$ passwords, SK_i, PID_i, RID_i and $DATA_i$ are 64 bits while the length of these values in Chen *et al.* protocol are 32 bits [4]. In addition, in the improved protocol, we assume that the length of random numbers are 64 bits. We use PRESENT [1] block cipher for encrypting messages in the server and decrypting messages in the reader which is an ultra lightweight block cipher. In the improved protocol has been assumed that the length of electronic product code (i.e. *EPC*) is 96 bits same as the most other EPC C1 G2 compliant protocols.

AKARI-1 and AKARI-2 lightweight PRNGs were introduced by Martin *et al.* in [12]. These PRNGs are using filter functions to provide nonlinearity properties [12]. AKARI-1's filter function has an iterative structure and iterated 64 times. In contrast to AKARI-1, to decrease the number of iteration to 24, in AKARI-2 two mixed filter functions are used. Pseudo-code of AKARI-1 and AKARI-2 PRNGs are depicted in Fig. 2, where (≪) and (≫) denote left and right circular shift, respectively.

Martin *et al.* have proposed several implementation schemes for AKARI PRNGs which are AKARI-1A, AKARI-1B, AKARI-2A, AKARI-2B and AKARI-2C. AKARI-1A and AKARI-2A were designed to minimize the number of required clock cycles while AKARI-1B and AKARI-2B were designed to reduce the chip area. AKARI-2C attempts to decrease the area more, at the expense of more clock cycles.

As depicted in Table 2 and Table 3, the required gate equivalents (GE), power and clock cycles for AKARI PRNGS are in the range of EPC C1 G2 RFID tags specification (an EPC C1 G2 standard tag supports at most 4000 GE, at most 600 clock cycles and a power in the range of micro-watt can devote to security purposes).

Table 2. AKARI-1 PRNG [12]

128 bits	Gate Equivalents	Power (nW)	Clock cycles
AKARI-1A	3898	343	66
AKARI-1B	3402	350	450

Table 3. AKARI-2 PRNG [12]

128 bits	Gate Equivalents	Power (nW)	Clock cycles
AKARI-2A	3743	216	51
AKARI-2B	3193	255	290
AKARI-2C	3040	231	530

```
AKARI-1
Initialize x0 and x1 of m-bits
x0=x0+((x0*x0)∨5)
x1=x1+((x1*x1)∨13)
z=x0
for r from 0 to 63
z=(z»1)+(z«1)+z+x1
%Output m/2 bits
Lower half of z
```

```
AKARI-2
Initialize x0 and x1 of m-bits
x0=x0+((x0*x0)∨5)
x1=x1+((x1*x1)∨13)
z=x0^x1
for r from 0 to 24
z=(z«1)+((z+(0x56AB0A))>1)
y=x1^z
for r from 0 to 24
y=(y»1)+(y«1)+y+(0x72A4FB))
z=z^y
%Output m/2 bits
Lower half of z
```

Figure 2. Pseudo-code of AKARI-1 and AKARI-2 PRNGs [12]

Desynchronization attack was another vulnerability of Chen *et al.* protocol. To overcome this flaw, in the improved protocol, we assume that the server stores RID_i , PID_{iold} , $Access\_key_{iold}$, $Kill\_key_{iold}$, PID_{inew} , $Access\_key_{inew}$ and $Kill\_key_{inew}$. Hence, stopping the last message of protocol between reader and tag would not be lead to desynchronize them.

The improved protocol, which is shown in (Fig. 3), runs as below. It must be noted that in the improved protocol, we use 64 bit output length CRCs and 64 bit output length PRNGs. In the other EPC C1 G2 compliant protocols, CRCs and PRNGs with the same output length usually equal to 16 or 32 bits were used.

1. The reader generates a random number N_1, computes $A = CRC(N_1)$ and sends M_{req} and A to the tag.

2. On reception the messages, the tag does as follows:

- generates a random number N_2,
- computes:

  $X = N_2 \oplus Kill\_key_i$,

  $B = CRC(A||X)$,

  $C_T = PRNG(EPC_i||B)$,
- responds (C_T, N_2, PID_i) to the reader.

3. On reception (C_T, N_2, PID_i), the reader sends $(C_T, N_2, PID_i, A, RID_i)$ to the server.

4. Once the server received $(C_T, N_2, PID_i, A, RID_i)$, it does as follows:

- verifies correctness of PID_i and RID_i; if they are correct, then the server based on PID_i retrieves $(EPC_i, Kill\_key_i, Access\_key_i, SK_i, DATA_i)$,
- computes:

  $X' = N_2 \oplus Kill\_key_i$,

  $B = CRC(A||X')$,
- verifies whether $C_T \overset{?}{=} PRNG(EPC_i|| B')$, if so, then it :

  – computes Y, C_1 and C_s as follows:

  $Y = N_2 \oplus Kill\_key_i$,

  $C_1 = E_{SKi}(DATA_i)$,

  $C_s = PRNG(EPC_i||Y||Access\_key_i) \oplus Kill\_key_i$,

  – updates PID_i , $Kill\_key_i$ and $Access\_key_i$ as follows:

  $PID_{i_{old}} = PID_i$,

  $Kill\_key_{i_{old}} = Kill\_key_i$,

  $Access\_key_{i_{old}} = Access\_key_i$,

  $PID_{i_{new}} = PRNG(PID_i)$,

  $Kill\_key_{i_{new}} = PRNG(Kill\_key_i)$,

  $Access\_key_{i_{new}} = PRNG(Access\_key_i)$, sends (C_s, C_1) to the reader.

5. On reception (C_s, C_1), the reader forwards C_s to the tag and extracts $DATA_i$ as $DATA_i = D_{SKi}(C_1)$.

6. Once the tag received (C_s), it does as follows:

- checks whether $C_s \overset{?}{=} PRNG(EPC_i||Y|| Access\_key_i) \oplus Kill\_key_i$, in the case of equality, computes the following values:

  $PID_i = PRNG(PID_i)$,

  $Kill\_key_i = PRNG(Kill\_key_i)$,

  $Access\_key_i = PRNG(Access\_key_i)$.

5. Security Analysis of the Improved Protocol

Generally, two approaches are used to prove the security of a cryptographic authentication protocol: informal methods and formal methods. Informal methods, in order to prove the security correctness of cryptographic protocols, rely on the heuristic opinions of security experts to draw a conclusion. On the other hand, the formal methods rely on the mathematical rules and frameworks.

In this section, we show that the improved protocol is immune against the attacks that considered in this paper and the other known active and passive attacks. Hence, we can claim that the improved protocol provides better security level compared to its predecessors. Our security proof is carried out based on informal method and also based on formal method.

5.1. Informal Security Analysis

As previously mentioned, the informal security analysis consists of a series of trial and error methods to find security holes in the protocols. Objections of this type of analysis are summarized as below:

- Its reliance on intelligence and ingenuity of analyst;
- Since it is always possible that the analyst forget or ignore some points during the analysis, so there is no way to understand the analysis is complete or not.

However, due to its simplicity and lack of need for sophisticated tools, the informal methods are widely used in the protocol security analysis. In this subsection, we argue the soundness and security of the improved protocol against the known attacks in the context.

Resistance against Desynchronization Attack

Desynchronization attacks may occur by stopping the last messages of the protocol between the reader and the tag. In the improved protocol, in order to defend against a desynchronization attack, we assume that the server maintains RID_i , PID_{iold}, $Access\_key_{iold}$, $Kill\_key_{iold}$, PID_{inew} , $Access\ \_key_{inew}$ and $Kill\_key_{inew}$. This assumption

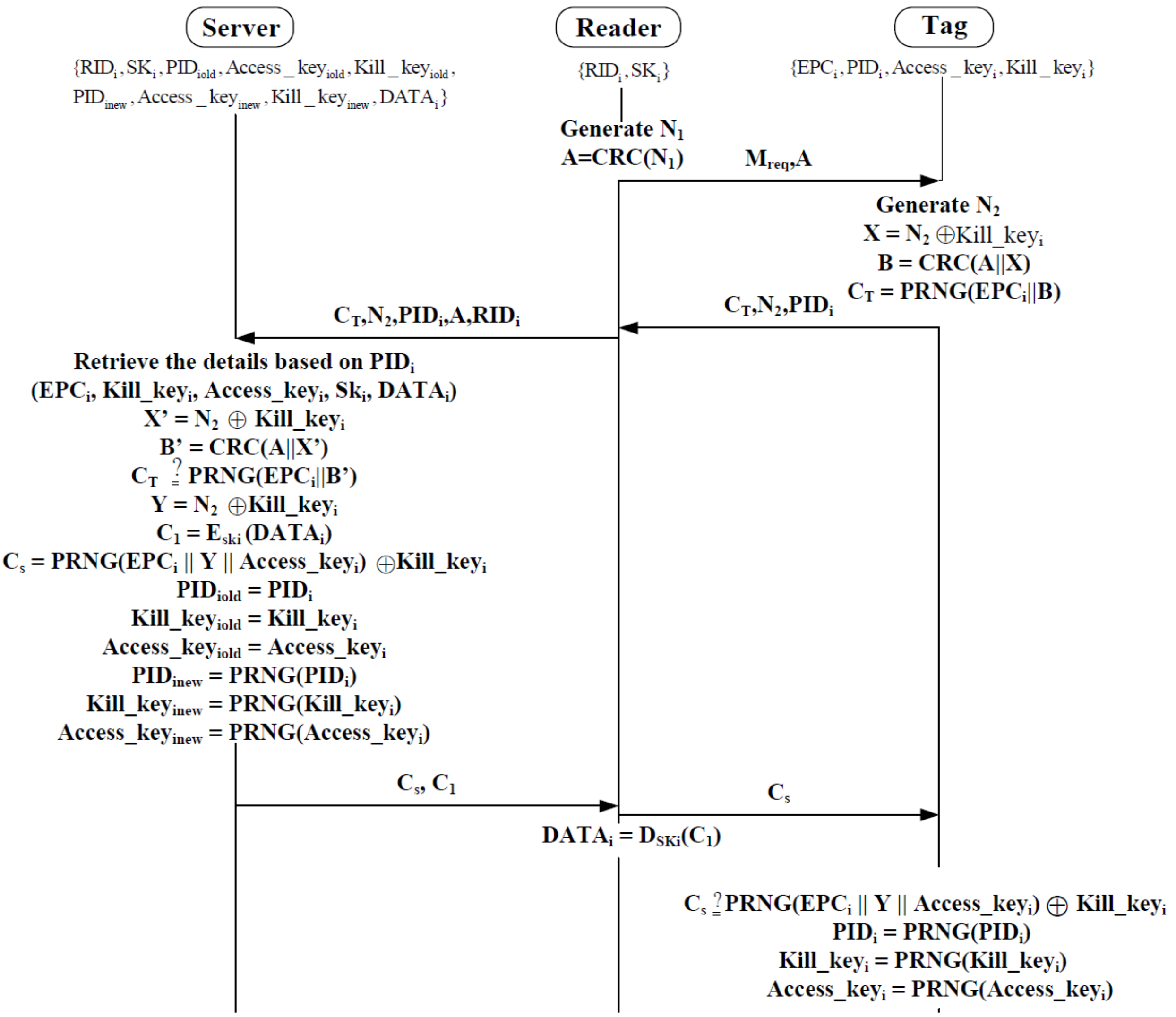
Server
{RID_i, SK_i, PID_iold, Access_key_iold, Kill_key_iold, PID_inew, Access_key_inew, Kill_key_inew, DATA_i}
Reader
{RID_i, SK_i}
Tag
{EPC_i, PID_i, Access_key_i, Kill_key_i}
Generate N1
A=CRC(N1)
M_req,A
Generate N2
X = N2 ⊕Kill_key_i
B = CRC(A||X)
C_T = PRNG(EPC_i||B)
C_T,N2,PID_i
C_T,N2,PID_i,A,RID_i
Retrieve the details based on PID_i
(EPC_i, Kill_key_i, Access_key_i, Sk_i, DATA_i)
X' = N2 ⊕ Kill_key_i
B' = CRC(A||X')
C_T ≟ PRNG(EPC_i||B')
Y = N2 ⊕Kill_key_i
C1 = E_ski (DATA_i)
C_s = PRNG(EPC_i || Y || Access_key_i) ⊕Kill_key_i
PID_iold = PID_i
Kill_key_iold = Kill_key_i
Access_key_iold = Access_key_i
PID_inew = PRNG(PID_i)
Kill_key_inew = PRNG(Kill_key_i)
Access_key_inew = PRNG(Access_key_i)
C_s, C1
C_s
DATA_i = D_SKi(C1)
C_s ≟PRNG(EPC_i || Y || Access_key_i) ⊕ Kill_key_i
PID_i = PRNG(PID_i)
Kill_key_i = PRNG(Kill_key_i)
Access_key_i = PRNG(Access_key_i)

Figure 3. The improved protocol

allows the server to authenticate tags and re-synchronize these each time they suffer a desynchronization attack.

Resistance against Tag Impersonation Attack

In the improved protocol, lightweight PRNGs same as AKARI-1 and AKARI-2 have been used instead of CRC function in calculation of C_T and C_s messages. Therefore, the adversary cannot calculate correct C_T and consequently cannot send it to the reader. Hence, immunity of the improved protocol against tag impersonation attack is proved.

Resistance against Server Impersonation Attack

Using $PRNG$ function instead of CRC function in calculation of C_T and C_s messages and also sharing of $Access\_key_i$ and $Kill\_key_i$ between the legitimate server and the legitimate tag lead the adversary can not compute and send correct C_s to the tag. Hence, the improved protocol resists against server impersonation attack.

Resistance against Traceability Attack

Due to the use of PRNG function in calculation of C_T, the adversary cannot deduce any special information related to the target tag's information from C_T and therefore cannot distinguish it among other tags and consequently cannot trace the target tag. Hence, the improved protocol resists against traceability attack.

Resistance against Replay Attack

Updating all shared secret data, B and N_2 in each round leads the adversary cannot use the previously obtained data to pass the authentication tests. Thus, the improved protocol provides resistance against replay attack.

All in all, the improved protocol exploiting lightweight PRNGs with larger output length same as AKARI-1 and AKARI-2 instead of CRC function can meet all security requirements of an RFID EPC C1 G2 compliant authentication protocol.

5.2. Formal Security Analysis

As mentioned before, the formal methods are techniques for cryptographic protocols' analysis which describe properties of protocols based on mathematics and logic. In these methods, protocols and also their features are modeled based on algebra and logic. There are a number of logic tools to prove the security correctness of cryptographic authentication protocols such as BAN logic [19], GNY logic [17], AVISPA tool [2] and Proverif tool [3]. In this paper, we use BAN logic to prove the security correctness of the proposed protocol for the following reasons:

- It is proposed for reasoning about cryptographic authentication protocols.
- It is easy to formalize and apply to a cryptographic authentication protocol.

In this subsection, we show that after one run of the proposed protocol, the tag, the reader and the server believe that the received messages are from each other and these messages are fresh and in this manner they can be authenticated to each other. Formal analysis of the improved protocol with the BAN logic includes the following four steps [19]:

- Stating the messages and the actions of the protocol parties in the mathematical relations;
- Converting the messages and the actions of the protocol parties into BAN logic formulas and dropping the plain text messages from protocol messages. Outputs of this step, are called idealized messages;
- Expressing the protocol initial assumptions and security goals as BAN logic formulas;
- Deducing the protocol security goals. In this step, using BAN logic rules, it is considered whether protocol security goals are satisfied or not.

In the rest of the paper, R_i, T_i and S denote the reader, the tag and the server respectively. BAN logic notations and rules which are used in the proof are shown in Table 1.

Stating the messages of the protocol in the mathematical relations

First, we express messages of the proposed protocol as the mathematical relations as below:

$M1 : R_i \rightarrow T_i : M_{req}, CRC(N_1)$

$M2 : T_i \rightarrow R_i : PID_i, N2, PRNG(EPC_i||CRC((CRC(N_1)||(N_2 \oplus Kill\_key_i))))$

$M3 : R_i \rightarrow S : RID_i, PID_i, N_2, CRC(N_1), PRNG(EPC_i||CRC((CRC(N_1)||(N_2 \oplus Kill\_key_i))))$

$M4 : S \rightarrow R_i : E_{SK_i}(DATA_i), PRNG(EPC_i|| (N_2 \oplus Kill\_key_i)||Access\_key_i) \oplus Kill\_key_i$

$M5 : R_i \rightarrow T_i : PRNG(EPC_i||(N_2 \oplus Kill\_key_i)||Access\_key_i) \oplus Kill\_key_i.$

Converting the protocol messages into idealized form based on BAN logic formulas

In this step, we transform each message of the proposed protocol into an idealized message, *i.e.* plaintexts are omitted from protocol messages and only encrypted message contents are relevant to this step. We also use BAN logic notations for expressing these idealized messages as follows:

$IM1 : R_i \lhd \{N_1, N_2, Kill\_key_i\}_{EPC_i}$

$IM2 : S \lhd \{N_1, N_2, Kill\_key_i\}_{EPC_i}$

$IM3 : R_i \lhd \{N_2, Kill\_key_i, Access\_key_i\}_{EPC_i}$

$IM4 : R_i \lhd \{DATA_i\}_{SK_i}$

$IM5 : R_i \lhd \{N_2, Kill\_key_i, Access\_key_i\}_{EPC_i}$

Expressing the initial assumptions and security goals as BAN logic formulas

The explicit assumptions of the proposed protocol are as follows:

$A1 : S| \equiv T_i \overset{EPC_i}{\longleftrightarrow} S$

$A2 : T_i| \equiv S \overset{EPC_i}{\longleftrightarrow} T_i$

$A3 : S| \equiv T_i \overset{Kill\_key_i}{\longleftrightarrow} S$

$A4 : T_i| \equiv S \overset{Kill\_key_i}{\longleftrightarrow} T_i$

$A5 : S| \equiv T_i \overset{Access\_key_i}{\longleftrightarrow} S$

$A6 : T_i| \equiv S \overset{Access\_key_i}{\longleftrightarrow} T_i$

$A7 : S| \equiv R_i \overset{SK_i}{\leftrightarrow} S$

$A8 : R_i| \equiv S \overset{SK_i}{\leftrightarrow} R_i$

$A9 : R_i| \equiv \#(N_1)$

$A10 : T_i| \equiv \#(N_2)$

The assumptions A1 to A8 are related to secrets which are shared between the protocol parties and the assumptions A9 and A10 are related to freshness of random numbers which are generated by the reader and the tag respectively.

The goals of the proposed protocol are as below:

$G_1 : S| \equiv T_i| \sim N_2$

$G_2 : R_i| \equiv S| \sim DATA_i$

In the above, G_1 means that the server believes that the tag T_i has sent the random number N_2. This goal indicates that the adversary does not have any control on this random number. Since N_2 has been generated by the tag and transmitted to the server through of the reader, the adversary has not made any undetected modification on this random number, to apply an attack on the protocol.

G_2 means that the reader believes that the server has sent the information contained in T_i, *i.e.* $DATA_i$. This goal indicates that the adversary has not made any undetected modification on this random number, to apply an attack on the protocol.

Deducing the protocol security goals BAN logic rules are expressed as fractional forms [19] which if their numerator expressions are correct then it can be concluded that their denominator expressions are also correct. In this step, we combine idealized messages and the assumptions to construct numerator expressions of BAN logic rules. If such relations are corresponded to the numerator expressions of BAN logic rules it can be concluded that the denominator expressions of BAN logic rules are correct. We show these deductions as below:

We consider $IM2$ idealized message which is previously expressed as $S \lhd \{N_1, N_2, Kill\_key_i\}_{EPC_i}$ with $A1$ assumption which is expressed as $S| \equiv T_i \overset{EPC_i}{\longleftrightarrow} S$. It can easily seen that the numerator of $P1$ rule of BAN logic which is expressed as $\frac{P|\equiv P\overset{K}{\leftrightarrow}Q, P \lhd \{X\}_K}{P|\equiv Q|\sim X}$ is constructed. So based on BAN logic, we can deduce that its denominator is also correct which can be expressed as follows:

$D1 : IM2, A1, P1 \Longrightarrow S|\equiv T_i|\sim\{N_1, N_2, Kill\_key_i\}$

Similarly, if we consider the previous result which is expressed as *i.e.* $D1 : S \mid \equiv T_i| \sim \{N_1, N_2,$ $Kill\_key_i\}$, it can easily seen that the numerator of $P2$ rule of BAN logic, *i.e.* $P2 : \frac{P|\equiv Q\sim\{X,Y\}}{P|\equiv Q|\sim\{X\}}$ is constructed. So, we can deduce that its denominator is also correct which can be shown as follows:

$D2 : D1, P2 \Longrightarrow S|\equiv T_i|\sim N_2$

Finally, we consider $IM4$ idealized message which is expressed as $R_i \lhd \{DATA_i\}_{SK_i}$ with $A8$ assumption which is expressed as $R_i| \equiv S \overset{SK_i}{\leftrightarrow} R_i$. It can easily seen that the numerator of $P1$ rule of BAN logic, *i.e.* $\frac{P|\equiv P\overset{K}{\leftrightarrow}Q, P \lhd \{X\}_K}{P|\equiv Q|\sim X}$ is constructed. So, we can deduce that its denominator is also correct which can be shown as follows:

$D3 : IM4, A8, P1 \Longrightarrow R|\equiv S|\sim DATA_i$

It can be deduced that D_2 equals to G_1 and D_3 equals to G_2 goal. Hence, the security goals of the proposed protocol, *i.e.* G_1 and G_2, are satisfied.

6. Performance Analysis of the Improved Protocol

In this section, we evaluate the performance of the proposed protocol in several aspects which are computation cost, communication cost, storage cost and security analysis. The performance comparisons between the proposed protocol and the existing work are shown in Tables 4 to 8.

6.1. Computation Cost

Tags computation restriction is the main problem of designing secure protocols for RFID EPC C1 G2 complaint systems. Therefore, we use several lightweight operations on the tag which are bitwise XOR, 64 bit output length CRC and 64 bit output length AKARI-1 and AKARI-2. AKARI-1 and AKARI-2 are lightweight pseudo random number generators.

Tags in the proposed protocol only need to execute six PRNG functions, two bitwise XOR operations and one CRC operation to finish both the authentication and secret updating phases while reader computes one PRNG operation for producing random number, one CRC operation and one decryption function and server

Table 4. Computation Cost

	Chien ***et al.* [14]**	**Qingling** ***et al.* [6]**	**Chen and** **Deng [5]**	**AZUMI** **[24]**	**Chen** ***et al.* [4]**	**Improved** **Protocol**
# of $\boldsymbol{CRC}(.)$ in tag	2	4	2	-	3	1
# of $\boldsymbol{CRC}(.)$ in reader	-	-	2	-	1	1
# of $\boldsymbol{CRC}(.)$ in server	2	4	-	-	3	1
Total # of $\boldsymbol{CRC}(.)$	4	8	4	-	7	3
# of $\boldsymbol{PRNG}(.)$ in tag	3	1	6	7	4	6
# of $\boldsymbol{PRNG}(.)$ in reader	1	1	6	6	1	1
# of $\boldsymbol{PRNG}(.)$ in server	2	-	-	-	3	5
Total # of $\boldsymbol{PRNG}(.)$	6	2	12	13	8	12
# of ⊕ bits in tag	32	160	480	118	64	128
# of ⊕ bits in reader	-	-	384	96	-	-
# of ⊕ bits in server	96	160	-	-	96	192
Total # of ⊕ bits	128	320	864	214	160	320
# of Encryption/ Decryption in tag	-	-	-	-	-	-
# of Encryption/ Decryption in reader	-	-	-	-	1	1
# of Encryption/ Decryption in server	-	-	-	-	1	1
Total # of Encryption/ Decryption	-	-	-	-	2	2

Table 5. The output length of CRC, $PRNG$ and random numbers in different protocols

	Chien *et al.* **[14]**	**Qingling** *et al.* **[6]**	**Chen and Deng [5]**	**AZUMI [24]**	**Chen** *et al.* **[4]**	**Improved Protocol**
Output length of # of $\boldsymbol{CRC}(.)$	32 [14]	16 [6]	16 [5]	- [24]	32 [4]	64
Output length of # of $\boldsymbol{PRNG}(.)$	32 [14]	16 [6]	16 [5]	16 [24]	32 [4]	64
Length of random numbers	32 [14]	16 [6]	16 [5]	16 [24]	32 [4]	64

Table 6. Communication Cost

	Chien *et al.* **[14]**	**Qingling** *et al.* **[6]**	**Chen and Deng [5]**	**AZUMI [24]**	**Chen** *et al.* **[4]**	**Improved Protocol**
# of transferred bits for tag	64	48	288	48	96	192
# of transferred bits for reader	160	112+$\boldsymbol{M_{req}}$	112+$\boldsymbol{M_{req}}$+$\boldsymbol{M_{resp}}$	32+$\boldsymbol{M_{req}}$	224+$\boldsymbol{M_{req}}$	508+$\boldsymbol{M_{req}}$
# of transferred bits for server	32	32	-	-	32+$\boldsymbol{\lvert E\rvert}$	64+$\boldsymbol{\lvert E\rvert}$
Total # of transferred bits	256	$\boldsymbol{192 + M_{req}}$	400+$\boldsymbol{M_{req}}$+$\boldsymbol{M_{resp}}$	80+$\boldsymbol{M_{req}}$	352+$\boldsymbol{M_{req}}$+$\boldsymbol{\lvert E\rvert}$	764+$\boldsymbol{M_{req}}$+$\boldsymbol{\lvert E\rvert}$

Table 7. Storage Cost

	Chien *et al.* **[14]**	**Qingling** *et al.* **[6]**	**Chen and Deng [5]**	**AZUMI [24]**	**Chen** *et al.* **[4]**	**Improved Protocol**
# of stored bits for tag	160	64	288	48	192	288
# of stored bits for reader	-	-	288	80	64	128
# of stored bits for server	256	64	-	-	288	672
Total # of stored bits	416	128	576	128	544	1088

CRC operation and one decryption function and server executes one CRC operation, two bitwise XOR operations, one encryption operation and six PRNG functions. In Table 4, the number of PRNG function calls to produce random numbers are also counted. If the bit length of random numbers is more than the output bits of the protocol's PRNG, random number generation needs to make more calls to PRNG function which also are counted in our comparison. For example, in Chen and Deng protocol [5] the length of random numbers are 96 bits while the output length of used PRNGs is 16 bits. So, to generate theses random numbers it is required to make six calls to PRNG function. The output length of CRC, PRNG and the length of random numbers in the proposed protocol and the existing work are shown in Table 5.

6.2. Communication Cost

As mentioned before, in the improved protocol, we use 64 bit output length CRCs and 64 bit output length PRNGs while in the other EPC C1 G2 complaint protocols CRCs and PRNGs with the same length usually equal to 16 bits or 32 bits were used.

In Table 6, it can easily seen that the number of all transferred bits in the proposed protocol is $764 + |M_{req}| + |E|$. Meanwhile, the number of all transferred bits in Chien *et al.* protocol [14], Qingling *et al.* [6], Chen and Deng protocol [5], AZUMI protocol [24] and Chen *et al.* [4] protocol are 256, $192 + |M_{req}|$, $400 + |M_{req}| + |M_{resp}|$, $80 + |M_{req}|$, $352 + |M_{req}| + |E|$, respectively, where $|M_{req}|$, $|M_{res}|$ and $|E|$ denote the bit length of request message, response message and encryption function output, respectively.

6.3. Storage Cost

In Chen *et al.* protocol, each tag stores its 96 bits length electronic product code (EPC_i), its pseudonym (PID_i), its secret kill password, *i.e.* $Kill\_key_i$ and its secret access password, *i.e.* $Access\_key_i$ which have the bit length of 32 bits. The server also stores all tags and reader information such as RID_i, SK_i, EPC_i, PID_{inew}, $Access\_key_{inew}$, $Kill\_key_{inew}$ and $DATA_i$. Reader also stores RID_i and SK_i which have the bit length of 32 bits. The storage cost comparisons between the proposed protocol and the existing work are shown in Table 7.

In the proposed protocol, each tag stores its 96 bits length electronic product code (EPC_i), its pseudonym (PID_i), its secret kill password, *i.e.* $Kill\_key_i$ and its secret access password, *i.e.* $Access\_key_i$ which have the bit length of 64 bits. The EPC is a static value, thus stored in ROM. The remaining values are stored in a rewritable memory because they need to be updated. Actually, we store a little more information on the server side, which are all tags and reader information such as (RID_i, Sk_i, EPC_i, PID_{iold}, $Kill\_key_{iold}$, $Access\_key_{iold}$, PID_{inew}, $Kill\_key_{inew}$, $Access\_key_{inew}$, $DATA_i$) which is much cheaper on the hardware and easy to implement. Reader also stores 64 bits length RID_i and 64 bits length SK_i.

Table 8. Security Analysis: SP depicts the success probability of the attack and C. shows the number of protocol's run which is required in the attack

	Chien *et al.* **[14]**		**Qingling** *et al.* **[6]**		**Chen and Deng [5]**		**AZUMI [24]**		**Chen** *et al.* **[4]**		**Improved Protocol**	
	SP	C.	**SP**	C.	**SP**	C.	**SP**	C.	**SP**	C.	**SP**	C.
Tag Impersonation Attack	1 [26]	2	1 [18]	2	1 [24]	2	1 [22]	2	1	2	secure	-
Server-Reader Impersonation Attack	1 [26]	2	1 [18]	1	1 [24]	2	secure	-	1	2	secure	-
Traceability Attack	1 [26]	2	0.49 [25]	2	0.49 [24]	2	secure	-	$1-\frac{1}{2^n}$	2	secure	-
Replay Attack	secure	-	secure [18]	-	1 [18]	1	secure	-	secure	-	secure	-
Desynchronization attack	secure	-	secure	-	secure	-	1	1	1	1	secure	-

7. Conclusion

In this paper, we have analyzed the security of a mutual authentication scheme conforming to EPC C1 G2 standard which has been proposed by Chen *et al.*. Precisely, we indicated this protocol's vulnerability against desynchronization attack, tag impersonation attack, server impersonation attack and traceability attack. The success probability of tag impersonation and server impersonation attacks is 1 and the success probability of the traceability attack is "$1-\frac{1}{2^n}$" where n is the bit length of parameters in the protocol. Meanwhile, the complexity of all presented attacks in this paper is only two runs of the protocol.

This paper shows that Chen *et al.* protocol is not anymore secure. So, we proposed an improved protocol and proved that it is resistant to the attacks considered in this paper and the other known active and passive attacks. We use BAN logic as a formal method to prove the security correctness of the proposed protocol.

This paper showed that the EPC C1 G2 standard's recommendations need to be revised by security experts in RFID field. This paper also indicated that the proposing the new framework free of all known security faults for lightweight RFID tags is inevitable.

Acknowledgments

We would like to thank anonymous reviewers for useful comments.

References

[1] **A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, C. Vikkelsoe.** Present: An Ultralightweight Block Cipher. *Workshop on Cryptographic Hardware and Embedded Systems (CHES'07)*, 2007, Vol. 4727, pp. 450-466.

[2] **AVISPA home page.** *Available online at http://www.avispa-project.org/* (Last access 02/02/2014).

[3] **B. Blanchet.** Automatic Verification of Cryptographic Protocols in the Formal Model Automatic Verifier ProVerif. Available online at *www.mpi-inf.mpg.de/.../VTSA11proverif.pdf.* (Last access 02/02/2014).

[4] **C. L. Chen, Y. C. Huang, T. F. Shih.** A Novel Mutual Authentication Scheme for RFID Conforming EPCglobal Class 1 Generation 2 Standards. *Information Technology and Control*, 2012, Vol. 41, No. 3, 220-228.

[5] **C. L. Chen, Y. Y. Deng.** Conformation of EPC Class 1 Generation 2 standards RFID system with mutual authentication and privacy protection. *Engineering Applications of Artificial Intelligence*, 2009, Vol. 22, 1284-1291.

[6] **C. Qingling, Z. Yiju, W. Yonghua.** A Minimalist Mutual Authentication Protocol for RFID System & BAN Logic Analysis. In: *Proceedings of 2008 ISECS International Colloquium on Computing, Communication, Control and Management (CCCM '08)*, 2008, pp. 449-453.

[7] **D. M. Konidala, K. Kim.** RFID tag-reader mutual authentication scheme utilizing tag's access password. *Auto-ID Labs White Paper WP-HARDWARE-033*, 2007.

[8] **D. V. Bailey, A. Juels.** Shoehorning Security into the EPC Tag Standard. *Security and Communication Networks, Lecture Notes in Computer Science*, 2006, Vol. 4116, pp. 303-320.

[9] **E. J. Yoon.** Improvement of the securing RFID systems conforming to EPC Class 1 Generation 2 standard. *Expert Systems with Applications*, 2012, Vol. 39, No. 1, 1589-1594.

[10] **EPCglobal Inc.** Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.09. Available online at *http://www.epcglobalinc.org/ standards technology/ specifications.html.* (Last access 02/02/2014).

[11] **E. Y. Choi, D. H. Lee, J. I. Lim.** Anti-cloning protocol suitable to EPCglobal Class-1 Generation-2 RFID systems. *Computer Standards & Interfaces*, 2009, Vol. 31, No. 6, 1124-1130.

[12] **H. Martin, E. S. Millán, L. Entrena, J. C. Hernández-Castro, P. Peris-Lopez.** AKARIX: A pseudorandom number generator for secure lightweight systems. In: *Proceedings of the 17th IEEE International On-Line Testing Symposium (IOLTS 2011)*, 2011, pp. 228-233.

[13] **H. M. Sun, W. C. Ting.** A Gen2-Based RFID Authentication Protocol for Security and Privacy. *IEEE Transactions on Mobile Computing*, 2009, Vol. 8, No. 8, 1052-1062.

[14] **H. Y. Chien, C. H. Chen.** Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards. *Computer Standards & Interfaces*, 2007, Vol. 29, No. 2, 254-259.

[15] **I. C. Lin, R. K. Luo, S. C. Tsao.** An Efficient Mutual Authentication Protocol for RFID Systems. In: *Proceedings of International Conference on Hybrid Intelligent Systems (HIS)*, 2009, pp. 41-45.

[16] **J. Andress.** The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice. *Elsevier Science & Technology*, 2011.

[17] **L. Gong, R. Needham, R. Yahalom.** Reasoning about Belief in Cryptographic Protocols. In: *IEEE Computer Society Symposium on Research in Security and Privacy*, 1990, pp. 234-248.

[18] **M. Burmester, B. de Medeiros, J. Munilla, A. Peinado.** Secure EPC Gen2 Compliant Radio Frequency Identification. *International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW), Lecture Notes in Computer Science*, 2009, Vol. 5793, pp. 227-240.

[19] **M. Burrows, M. Abadi, R. Needham.** A Logic of Authentication. *ACM Transactions on Computer Systems*, 1990, Vol. 8, No. 1, 18-36.

[20] **M. H. Habibi, M. R. Alaghband, M. R. Aref.** Attacks on a Lightweight Mutual Authentication Protocol under EPC C-1 G-2 Standard. *Workshop in Information Security Theory and Practice (WISTP 2014), Lecture Notes in Computer Science*, 2011, Vol. 6633, pp. 254-263.

[21] **M. Safkhani, N. Bagheri, M. Naderi.** A note on the security of IS-RFID, an inpatient medication safety. *International Journal of Medical Informatics*, 2014, Vol. 83, 82-85.

[22] **M. Safkhani, N. Bagheri, M. Naderi.** Cryptanalysis of AZUMI: an EPC Class-1 Generation-2 Standard Compliant RFID Authentication Protocol. *IACR Cryptology ePrint Archive*, 2011, pp. 424.

[23] **P. Peris-Lopez.** Hot Topics in RFID Security. Available online at *//www.cosic.esat.kuleuven.be/IEEE_Embed_Sec/registration/Presentations/peris_ieee.pdf.* (Last access 12/5/2013).

[24] **P. Peris-Lopez, J. C. Hernández-Castro, J. E. Tapiador, J. C. A. van der Lubbe.** Cryptanalysis of an EPC Class-1 Generation-2 standard compliant authentication protocol. *Engineering Applications of Artificial Intelligence*, 2011, Vol. 24, No. 6, 1061-1069.

[25] **P. Peris-Lopez, J. C. Hernández-Castro, J. E. Tapiador, T. Li, J. C. A. van der Lubbe.** Weaknesses in Two Recent Lightweight RFID Authentication Protocols. *Information Security and Cryptology (Inscrypt), Lecture Notes in Computer Science*, 2009, Vol. 6151, pp. 383-392.

[26] **P. Peris-Lopez, J. C. Hernández-Castro, J. M. Estévez-Tapiador, A. Ribagorda.** Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard. *Computer Standards & Interfaces*, 2009, Vol. 31, No. 2, 372-380.

[27] **T. C. Yeh, Y. J. Wang, T. C. Kuo, S. S. Wang.** Securing RFID systems conforming to EPC Class 1 Generation 2 standard. *Expert Systems with Applications*, 2010, Vol. 37, No. 12, 7678-7683.

Received December 2013.