Weaknesses in another Gen2-Based RFID Authentication Protocol

Masoumeh Safkhani*, Nasour Bagheri†, Pedro Peris-Lopez‡, Aikaterini Mitrokotsa§ and Julio C. Hernandez-Castro ¶
*Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran
Email: M_Safkhani@iust.ac.ir
†Department of Electrical Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran
Email: Nbagheri@srttu.ac.ir
‡Computer Security Lab, Carlos III University of Madrid, Spain
Email: pperis@inf.uc3m.es
§EPFL, Lausanne, Switzerland,
Email:katerina.mitrokotsa@epfl.ch
¶School of Computing, University of Kent, UK
Email: J.C.Hernandez-Castro@kent.ac.uk

***Abstract*—There is a high need for secure authentication protocols conforming with the EPC Class-1 Generation 2 (Gen2 in short) standard. The security analyses of the new born authentication protocols provide some guidelines and lessons that should be considered in the design of new proposals. In this paper, we scrutinize the security of a Gen2 based RFID authentication protocol which has been recently proposed by Yi *et al.* [8]. Our security analysis highlights important security pitfalls in this proposal. More precisely, we show a simple approach to desynchronize the tag and the reader. Moreover, we present tag impersonation and reader impersonation attacks. Finally, we show how the use of random numbers does not prevent traceability attack. The success probability of all the proposed attacks is 1 and their complexity is minimal since at most one eavesdropped session of the protocol is required.**

I. Introduction

Radio Frequency Identification (RFID) is a technology that can be employed in order to automatically identify objects and people. If we compare this technology with traditional barcodes, it has several advantages, e.g. the long reading distance, the unique identifier and the speedy identification of multiple objects without physical contact.

EPC Class-1 Generation-2 (or in brief Gen2) [4], [7] is a standard for RFID tags with very constrained resources. It only supports a very basic security level, provided by a 16-bit pseudorandom number generator (PRNG) and a 16-bit cyclic redundancy code (CRC). As a result of the inherent weaknesses in the Gen2 specification [5], there are a number of proposals in the literature that try to design more secure but still Gen2-compliant RFID authentication protocols [1]–[3]. These schemes can only use very lightweight operations (i.e., bitwise operations, CRC functions or PRNGs) as specified in the standard.

Recently, Yi *et al.* in [8] have analyzed the security of Chien's authentication protocol [2] and proposed an improved design based on a set of clear security requirements, heavily inspired on previous proposals. The authors claim that their protocol is much more secure than other Gen2-based schemes. However, our security analysis highlights important security shortcomings in the protocol proposed by Yi *et al.* More precisely, we show a simple approach to de-synchronize a tag and a reader that use the Yi *et al.* protocol. Moreover, we present a tag impersonation attack and then a reader impersonation attack. The success probability of all attacks is 1 and their complexity is at most eavesdropping one session of the protocol per attack, with negligible associated computational requirements.

Paper Organization: The paper is organised as follows. In Section II we give a brief description of the Yi *et al.* authentication protocol. In Section III we describe a desynchronization attack that can be deployed against the Yi *et al.* protocol. We describe reader impersonation and tag impersonation attacks in Section IV and Section VI respectively. Finally, Section VII concludes the paper.

R_i	RFID reader i
T_i	RFID tag i
EPC_x	Electronic product code of T_i
$CRC(.)$	16-bit cyclic redundancy code
N_1	16 bit random number generated by the reader
N_2	16-bit random number generated by the tag
K_{x_i}	Authentication key after i^{th} successful authentication
P_{x_i}	Access key after i^{th} successful authentication
$\{0\}^{16}$	A string of zeros of length 16-bit
Δ	A non zero arbitrary 16-bit number
$A \rightarrow B$	Sending a message from A to B

TABLE I
Notations

II. Protocols Description

The Yi *et al.* authentication protocol [8] is an extension of the Chien protocol [2]. It was proposed in order to overcome the weaknesses of the Chien protocol and also to offer forward secrecy and resistance against de-synchronization and forgery

978-1-4673-0328-6/12/$31.00 ©2012 IEEE

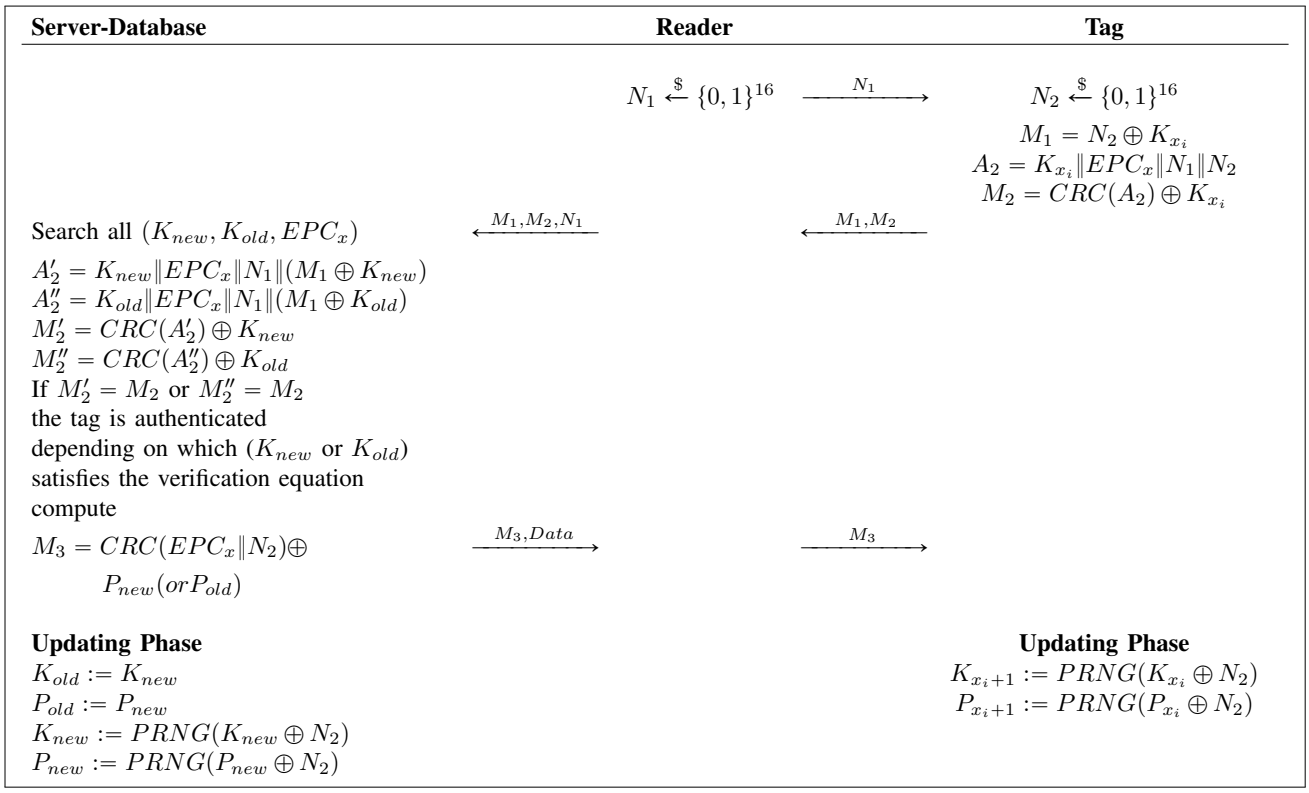

Fig. 1. The Gen2 Based Authentication Protocol proposed by Yi *et al.* [8].

attacks. A description of the Yi *et al.* protocol is provided in Figure 1. We give a brief description of this scheme, but we urge the reader to consult the original paper for further details [8]. Throughout this paper we use the notations indicated in Table I.

In Yi *et al.* protocol the server randomly chooses an initial authentication key K_{x_0} and access key P_{x_0}. At first, we have $K_{old} = K_{new} = K_{x_0}$ and $P_{old} = P_{new} = P_{x_0}$. The tag T_i shares its secret parameters, i.e. $(EPC_x, K_{x_0}, P_{x_0})$, with the server. After the first successful authentication, K_{x_0} and P_{x_0} will be updated to K_{x_1} and P_{x_1} respectively – similarly for the forthcoming sessions after the i^{th} successful authentication, $K_{x_{i-1}}$ and $P_{x_{i-1}}$ will be updated to K_{x_i} and P_{x_i} respectively. To avoid de-synchronization attacks, the server stores old and new values $(K_{new}, P_{new}, K_{old}, P_{old}, EPC_x, Data)$. The steps of the protocol are sketched below:

1) The reader generates a random number N_1 and sends it to the tag.
2) Once N_1 is received, the tag generates another random number N_2 and computes:

$$M_1 = N_2 \oplus K_{x_i}$$
$$M_2 = CRC(K_{x_i} \| EPC_x \| N_1 \| N_2) \oplus K_{x_i}$$

and sends M_1 and M_2 to the reader.
3) Upon receiving the tuple (M_1, M_2), the reader sends M_1, M_2 and N_1 to the server.
4) A searching process is started in the server. For each entry $(K_{new}, K_{old}, EPC_x)$ in the database, the server computes:

$$M'_2 = CRC(K_{new} \| EPC_x \| N_1 \| (M_1 \oplus K_{new})) \oplus K_{new}$$
$$M''_2 = CRC(K_{old} \| EPC_x \| N_1 \| (M_1 \oplus K_{old})) \oplus K_{old}$$

If M'_2 or M''_2 equals to M_2, the server authenticates the tag and depending on whether K_{new} or K_{old} satisfies the verification equation, it computes:

$$M_3 = CRC(EPC_x \| N_2) \oplus P_{new}(or P_{old})$$

and sends M_3 and $Data$ to the reader. Finally, it updates the tag's secret values as follows:

$$K_{old} := K_{new}$$
$$P_{old} := P_{new}$$
$$K_{new} := PRNG(K_{new} \oplus N_2)$$
$$P_{new} := PRNG(P_{new} \oplus N_2)$$

5) The reader forwards M_3 to the tag.
6) Once M_3 is received, the tag computes $M_3 \oplus P_{x_i}$ and verifies whether it is equal to $CRC(EPC_x \| N_2)$ or not. If so the tag authenticates the server and updates its secret values as follows:

$$K_{x_i+1} := PRNG(K_{x_i} \oplus N_2)$$
$$P_{x_i+1} := PRNG(P_{x_i} \oplus N_2)$$

Yi *et al.* have claimed that their protocol mainly meets the common security requirements (i.e. privacy, anonymity, resistance to replay attacks, resistance to DoS attacks, forward security, backward security, data confidentiality, mutual authentication, tag forgery resistance, server forgery resistance and data recovery). However, in the next sections we prove that unfortunately their claims do not hold and the proposed protocol is as insecure as its predecessors.

III. Desynchronization Attack

In this section we present a de-synchronization attack against Yi *et al.* Gen2 based RFID authentication protocol. Our de-synchronization attack exploits the linear behavior of CRC functions [6]. In fact, it is based on the observation that:

$$CRC(A||B||C||D) \oplus CRC(E||F||G||H) = CRC((A \oplus E)\|(B \oplus F)\|(C \oplus G)\|(D \oplus H))$$

In the proposed attack, the adversary behaves as a man-in-the-middle attacker who proceeds as follows:

1) The reader generates a random number N_1 and sends it to the tag.
2) Upon reception of N_1, the tag generates another random number N_2, computes:
$$M_1 = N_2 \oplus K_{x_i}$$
$$M_2 = CRC(K_{x_i}\|EPC_x\|N_1\|N_2) \oplus K_{x_i}$$
and sends M_1 and M_2 to the reader.
3) The adversary intercepts M_1 and M_2 and sends $M_1 \oplus \Delta$ and $M_2 \oplus CRC(\{0\}^{16}\|\{0\}^{16}\|\{0\}^{16}\|\Delta)$ to the reader, where based on the above mentioned property of CRCs the second message is $CRC(K_{x_i}\|EPC_x\|N_1\|N_2 \oplus \Delta) \oplus K_{x_i}$, where Δ is a non zero arbitrary value.
4) Once the above two messages are received, the reader forwards $M_1 \oplus \Delta$, $CRC(K_{x_i}\|EPC_x\|N_1\|N_2 \oplus \Delta) \oplus K_{x_i}$ and N_1 to the server.
5) Upon its reception, for each entry $(K_{new}, K_{old}, EPC_x)$ in the back-end database, the server computes:
$$M_2' = CRC(K_{new}\|EPC_x\|N_1\|(M_1 \oplus \Delta \oplus K_{new})) \oplus K_{new}$$
$$M_2'' = CRC(K_{old}\|EPC_x\|N_1\|(M_1 \oplus \Delta \oplus K_{old})) \oplus K_{old}$$
The server verifies whether M_2' or M_2'' matches with $CRC(K_{x_i}\|EPC_x\|N_1\|(N_2 \oplus \Delta)) \oplus K_{x_i}$ in order to authenticate the tag. After that and depending on whether K_{new} or K_{old} satisfies the verification equation, the server computes:
$$M_3 = CRC(EPC_x\|(N_2 \oplus \Delta)) \oplus P_{new}(\text{ or } P_{old})$$
and sends M_3 and $Data$ to the reader and finally updates its values as follows:
$$K_{old} := K_{new}$$
$$P_{old} := P_{new},$$
$$K_{new} := PRNG(K_{new} \oplus N_2 \oplus \Delta)$$
$$P_{new} := PRNG(P_{new} \oplus N_2 \oplus \Delta)$$
6) When M_3 is received, the reader forwards it to the tag.
7) The adversary intercepts M_3 and sends to the tag:
$$\begin{aligned} M_3' &= M_3 \oplus CRC(\{0\}^{16}\|\Delta) \\ &= CRC(EPC_x\|(N_2 \oplus \Delta)) \oplus CRC(\{0\}^{16}\|\Delta) \\ &= CRC(EPC_x\|(N_2 \oplus \Delta \oplus \Delta) \\ &= CRC(EPC_x\|N_2) \end{aligned}$$
8) The tag after receiving M_3', it computes $M_3' \oplus P_{x_i}$ which is $CRC(EPC_x\|N_2)$. So the tag authenticates the server and uses N_2 during the updating phase of its secret values, but the server updated its internal parameters using $N_2 \oplus \Delta$. Thus the reader in the updating phase computes:
$$K_{x_i+1} := PRNG(K_{x_i} \oplus N_2)$$
$$P_{x_i+1} := PRNG(P_{x_i} \oplus N_2).$$

Hence, after conducting this attack, the reader and the tag, each of them update their shared secret values to different values. So they lose synchronization and will not authenticate each other in any future transaction (see Table II for details). It should be noted that the success probability of the given attack is 1 (100%) and the attack complexity is one run of the protocol, with negligible computational requirements.

IV. Reader Impersonation Attack

In this section we show how an adversary can deceive the tag to accept her as a legitimate reader. Similarly to the above attack, we use the following property of the CRC functions [6]:

$$CRC(A||B||C||D) \oplus CRC(E||F||G||H) = CRC((A \oplus E)\|(B \oplus F)\|(C \oplus G)\|(D \oplus H))$$

The steps for impersonating a reader are described below:

1) **Learning Phase:** In this phase of the attack, the adversary eavesdrops one session of the protocol and stores N_1, M_1, M_2 and M_3, that are easily accessible through the insecure radio channel. In addition, to prevent the tag from updating its secret values, the adversary blocks the last message, i.e., M_3, that is sent from the reader to the tag.
2) **Reader Impersonation Phase:** In this phase of the attack, the adversary supplants the reader as follows:
   - The adversary sends $N_1' = N_1$ which was eavesdropped in the previous phase from the tag.
   - Upon receiving $N_1' = N_1$, the tag generates another random number N_2' and computes:
   $$M_1' = N_2' \oplus K_{x_i}$$
   $$M_2' = CRC(K_{x_i}\|EPC_x\|N_1\|N_2') \oplus K_{x_i}$$
   and sends M_1' and M_2' to the reader (supplanted by the adversary). Note that the tag has not updated its secret values in the previous phase.
   - The adversary computes:
     - $M_1 \oplus M_1' = N_2 \oplus K_{x_i} \oplus N_2' \oplus K_{x_i} = N_2 \oplus N_2'$,

TABLE II
DESYNCHRONIZATION ATTACK: INTERNAL SECRET VALUES AT THE END OF THE ATTACK

Reader (database)	Tag
$K_{old} = K$ $P_{old} = P$	$K_{x_i+1} = PRNG(K \oplus N_2)$ $P_{x_i+1} = PRNG(P \oplus N_2)$
$K_{new} = PRNG(K \oplus N_2 \oplus \Delta)$ $P_{new} = PRNG(P \oplus N_2 \oplus \Delta)$	

– $CRC(\{0\}^{16} \| (N_2 \oplus N_2'))$,

$$\begin{aligned} M_3' &= M_3 \oplus CRC(\{0\}^{16} \| (N_2 \oplus N_2')) \\ &= CRC(EPC_x \| N_2) \oplus CRC(\{0\}^{16} \| (N_2 \oplus N_2')) \\ &= CRC(EPC_x \| (N_2 \oplus N_2 \oplus N_2')) \\ &= CRC(EPC_x \| N_2') \end{aligned}$$

and finally sends M_3' to the tag.

- Once M_3' is received, the tag computes $M_3' \oplus P_{x_i}$ which is identical to $CRC(EPC_x \| N_2')$. Hence, the tag authenticates the adversary as a legitimate reader.

After following the described steps, the adversary succeeds with probability equal to 1 while the attack requires only one eavesdropped successful run of the protocol.

V. TAG IMPERSONATION ATTACK

In this section, we show how the Yi *et al.* protocol does not offer protection against tag impersonation. Our proposed attack is described below:

1) **Learning Phase:** In this phase the adversary starts a session with the tag by sending a random value N_1, stores the tag's reply:

$$\begin{aligned} M_1 &= N_2 \oplus K_{x_i} \\ M_2 &= CRC(K_{x_i} \| EPC_x \| N_1 \| N_2) \oplus K_{x_i} \end{aligned}$$

and ends the protocol at this stage. Hence, the tag does not update its secret parameters.

2) **Tag Impersonation Phase:** In this phase the adversary supplants the legitimate tag. She waits until a legitimate reader starts a new session where:
   - The reader generates a random number N_1' and sends it to the tag (supplanted by the adversary).
   - The adversary replies with M_1 and $M_2 \oplus CRC(\{0\}^{16} \| \{0\}^{16} \| (N_1 \oplus N_1') \| \{0\}^{16})$ to the reader.
   - Upon receiving the above two messages, the reader sends M_1, $M_2 \oplus CRC(\{0\}^{16} \| \{0\}^{16} \| (N_1 \oplus N_1') \| \{0\}^{16})$ and N_1' to the server.
   - After its reception, for each entry $(K_{new}, K_{old}, EPC_x)$ in the database the server computes:

$$\begin{aligned} M_2' &= CRC(K_{new} \| EPC_x \| N_1' \| (M_1 \oplus K_{new})) \oplus K_{new} \\ M_2'' &= CRC(K_{old} \| EPC_x \| N_1' \| (M_1 \oplus K_{old})) \oplus K_{old} \end{aligned}$$

where for the target tag's record, either M_2' or M_2'' matches with the received:

$$\begin{aligned} &M_2 \oplus CRC(\{0\}^{16} \| \{0\}^{16} \| (N_1 \oplus N_1') \| \{0\}^{16}) = \\ &CRC(K_{old} \| EPC_x \| (N_1 \oplus N_1 \oplus N_1') \| (M_1 \oplus K_{old})) \\ &\oplus K_{old} \end{aligned}$$

The server then authenticates the adversary as a legitimate tag and depending on which K_{new} or K_{old} satisfies the verification equation it computes:

$$M_3 = CRC(EPC_x \| N_2) \oplus P_{new} (or P_{old})$$

Finally, the server sends M_3 and $Data$ to the reader and updates its internal secret values.

- The reader forwards M_3 to the tag (supplanted by the adversary).

Hence, following the given attack, the server authenticates the adversary as a legitimate tag. It should be noted that as a collateral effect of the proposed attack, this leads also to a desynchronization attack because through this attack the server updates its secret values while the tag does not update its secret parameters. The success probability of the given attack is 1 while the attack's complexity is just the eavesdropping of one protocol session.

VI. TRACEABILITY ATTACK

Protection against traceability is one of the main objectives of all RFID authentication protocols. In the Yi *et al.* protocol, the authors argue that the use of nonces and the fact of never sending N_2 in plaintext over the insecure radio channel prevents any traceability attacks. That is, it is not possible to link a constant value to a particular tag. Nevertheless, we show how an attacker can bypass this by combining the messages passed over the insecure radio channel. The attack follows the steps described below:

1) **Step 1 - Learning Phase:** The adversary eavesdrops the messages $\{N_1, M_1, M_2, M_3\}$ and interrupts or alters the message M_3 in order to prevent the updating phase in the tag.
2) **Step 2 - Discovering Phase:** The adversary obtains the Z value (described below) that identifies unequivocally

the tag:

$$
\begin{aligned}
Y &= CRC(\{0\}^{16}\|\{0\}^{16}\|\{0\}^{16}\|M_1)\oplus \\
&\quad CRC(\{0\}^{16}\|\{0\}^{16}\|N_1\|\{0\}^{16}) \\
&= CRC(\{0\}^{16}\|\{0\}^{16}\|N_1\|N_2 \oplus K_{x_i}) \\
Z &= Y \oplus M_2 \\
&= CRC(K_{x_i}\|EPC_x\|N_1 \oplus N_1\|N_2 \oplus N_2 \oplus K_{x_i}) \oplus K_{x_i} \\
&= CRC(K_{x_i}\|EPC_x\|\{0\}^{16}\| \oplus K_{x_i}) \oplus K_{x_i}
\end{aligned}
$$

The adversary can repeat steps 1 and 2 indefinitely and if the same Z value is obtained, she is 100% sure that the same tag was interrogated again. The attack only requires the eavesdropping of one session (and blockage of one message) and some simple computations. So, the usage of random numbers clearly does not prevent traceability attacks at all.

VII. Conclusions

Designing secure Gen2-compliant RFID authentication protocols is particularly challenging because the only operations that are available in this sort of low-cost tags are a 16-bit pseudorandom number generator (PRNG) and a 16-bit cyclic redundancy code (CRC). CRCs should only be used for detection of errors in the transmitted messages. These functions do not provide the one-wayness required, and offered, for example, by cryptographic hash functions. In fact, they are linear codes whose one-wayness is comparable to that of a modular reduction, hence none. In this paper we have shown that a recently proposed Gen2 compliant RFID protocol by Yi *et al.* [8] fails to provide adequate security and is vulnerable against tag (reader) impersonation attacks, de-synchronization and traceability attacks (privacy location). The success probability of all the presented attacks is 1 and the eavesdropping of one session and some negligible computation are the only requirements in order to successfully perform each of the described attacks.

VIII. Acknowledgements

This work was partially supported by the Marie Curie IEF project "PPIDR: Privacy-Preserving Intrusion Detection and Response in Wireless Communications", grant number: 252323.

References

[1] C. Chen and Y. Deng. Conforming of EPC Class 1 Generation 2 Standards RFID System with Mutual Authentication and Privacy Protection. *Journal of Engineering Applications of Artificial Intelligence*, 22(8):1284–1291, 2009.

[2] H. Chien and C. Chen. Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation -2 Standard. *Computer Standards & Interfaces*, 29(2):254–259, 2007.

[3] E. Choi, D. Lee, and J. Lim. Anti-cloning Protocol Suitable to EPCglobal Class 1 Generation 2 RFID Systems. *Computer Standards and Interfaces*, 31(6):1124–1130, 2009.

[4] Class-1 generation 2 UHF air interface protocol standard version 1.2.0, Gen2, 2008. http://www.epcglobalinc.org/standards/. In *Gen-2 Standard*. EPCGlobal, 2008.

[5] D. N. Duc, J. Prk, H. Lee, and K. Kim. Enhancing Security of EPCglobal EPC Class 1 Generation 2 RFID Tag against Traceability and Cloning. *The 2006 Symposium on Cryptography and Information Security*, Berlin:Springer,2006.

[6] D. Han and D. Kwon. Vulnerability of an rfid authentication protocol conforming to epc class 1 generation 2 standards. *Computer Standards & Interfaces*, 31(4):648 – 652, 2009.

[7] Information technology - Radio frequency identification for item management. Part 6: Parameters for air interface communications at 860 MHz to 960MHz. http://www.iso.org. 2005.

[8] X. YI, L. W. D. MAO, and Y. ZHAN. An Gen2 Based Security Authentication Protocol for RFID System. *Physics Procedia*, 24:1385–1391, 2012.