Security Analysis of Niu *et al.* Authentication and Ownership Management Protocol

Nasour Bagheri, Masoumeh Safkhani and Hoda Jannati

***Abstract*—Over the past decade, besides authentication, ownership management protocols have been suggested to transfer or delegate the ownership of RFID tagged items. Recently, Niu *et al.* have proposed an authentication and ownership management protocol based on 16-bit pseudo random number generators and exclusive-or operations which both can be easily implemented on low-cost RFID passive tags in EPC global Class-1 Generation-2 standard. They claim that their protocol offers location and data privacy and also resists against desynchronization attack. In this paper, we analyze the security of their proposed authentication and ownership management protocol and show that the protocol is vulnerable to secret disclosure and desynchronization attacks. The complexity of most of the attacks are only two runs of the protocol and the success probability of the attacks are almost 1.**

***Index Terms*—RFID, ownership transfer, ownership delegation, secret disclosure attack, desynchronization attack.**

I. Introduction

Radio Frequency IDentifiction (RFID) is a wireless identification technology which contains tags, readers and servers and works using radio waves. Tag is a microchip which connects to the objects and the reader can read or modify the information of tags. Complex operations can be done in the servers and also more information about tags and readers are stored in them. There are three kinds of tags which are passive, active and semi-passive. Passive tags have no batteries and receive their energies from the reader, while the active tags have internal batteries which lead to increase their cost. Semi-passive tags are between passive and active tags, i.e., they have a small battery for some of their functionalities and the required energy for other functionalities is obtained form the reader [15].

Electronic Product Code Class-1 Generation-2 (or in brief EPC-C1G2) [6], [12] is one of the important standards related to passive tags which supports only Cyclic Redundancy Check (CRC) functions, Pseudo Random Number Generator (PRNG) functions and lightweight operations such as AND, OR, XOR and etc. Due to the widespread use of passive tags, many EPC-C1G2 complaint protocols such as authentication [16], [29], [18], [5], [17], [19], ownership transfer [9], tag search [27], [28], distance bounding protocols [10], grouping proof [26], [21], [20] and etc [14] have been designed. There also are many reports on vulnerabilities of these protocols against different attacks [13], [23], [4], [3], [30], [1], [8], [22], [2], [7], [25]. All of these efforts show that designing a secure protocol in the framework of EPC-C1G2 is not a straighted forward task and we still need secure protocols in this area. Recently, in response to this need, in [11], Niu *et al.* presented a mutual authentication and ownership management protocol including ownership transfer and ownership delegation in order to provide location and data privacy in EPC-C1G2 passive tags. Their protocol relies only on pseudo random number generators and exclusive-or operations for execution. Both operations are easily implemented on low-cost RFID passive tags that comply with EPC-C1G2 standard. Niu *et al*. claim that their protocol provides location privacy, backward privacy, forward privacy and also suitable security against replay attack, desynchronization attack and windowing.

In this paper, we analyze the security of the authentication and ownership management protocol proposed by Niu *et al*. and show that unfortunately their security claims do not hold. In particular, their protocol is vulnerable against secret disclosure and desynchronization attacks. In fact, this paper shows that this need is still unmet.

The rest of this paper is arranged as follows: In Section II, we review authentication and ownership management protocol proposed by Niu *et al*. Secret disclosure and the desynchronization attacks against the protocol are presented in Sections III and IV respectively and finally Section V concludes the paper.

II. Review of Niu *et al.* Authentication and Ownership Management Protocol

There are four types of players in the protocol proposed by Niu et al. [11]:

1) A trusted third party TTP
2) An RFID tag T
3) An old owner (reader R_{ID1})
4) A new owner (reader R_{ID2})

It must be noted that Niu *et al.* have assumed all the protocol parameters are 96 bits to preserve compatibility to EPC standard and all 96-bit parameters are broken into six 16-bit words because of convenience of the protocol implementation [11]. To prevent desynchronization attack, they also have assumed that the reader and the tag both should maintain their old and current pseudonyms and keys.

In the following, we begin with an overview of the system notations shown in Table I, then their protocol is described in three phases.

Nasour Bagheri is with the Department of Electrical Engineering, Shahid Rajaee Teachers Training University, Tehran, Iran (email: NBagheri@srttu.edu).

Masoumeh Safkhani is with the Department of Computer Engineering, Shahid Rajaee Teachers Training University, Tehran, Iran (email: Safkhani@srttu.edu).

Hoda Jannati is with the School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran (email: hodajannati@ipm.ir).

TABLE I
NOTATIONS UTILIZED TO FORMULATE NIU *et al.* PROTOCOL

Symbol	Description
EPC	The unique and static electronic product code of the tag T
IDS	The pseudonym of the tag T
R_{ID_i}	The identifier of i^{th} reader
K	A secret key which is shared between the tag T and its owner
K_M	A master key which is shared between the tag T and its owner (the owner of the tag T with K_M is able to modify the key K)
K_{TTP}	A secret key which is shared between the tag T and TTP
$W(i)$	i^{th} 16-bit of W
$PRNG(.)$	A 16-bit pseudo random number generator
$Per(X,Y)$	The permutation of $X = x_1x_2 \dots x_n$ according to $Y = y_1y_2 \dots y_n$ ($x_i, y_i \in \{0,1\}$, for $0 \leq i \leq n$) as $Per(X,Y) = x_{k_1}x_{k_2} \dots x_{k_m}x_{k_n}x_{k_{n-1}} \dots x_{k_{m+2}}x_{k_{m+1}}$ where m ($0 \leq m \leq n$) is the hamming weight of Y, so that $y_{k_1} = y_{k_2} = \dots = y_{k_m} = 1$ and $y_{k_{m+1}} = y_{k_{m+2}} = \dots = y_{k_m} = 0$ for $1 \leq k_1 < k_2 < \dots < k_m \leq n$ and $1 \leq k_{m+1} < k_{m+2} < \dots < k_n \leq n$

Mutual Authentication Phase:
In the mutual authentication phase of Niu *et al.* protocol the reader R_{ID1} (the old owner of the tag T) authenticates the tag T before the reader R_{ID1} delegates the ownership of the tag T to the reader R_{ID2}. This phase of the protocol is shown in Figure 1 and described below:

1) To start this phase of the protocol, the reader R_{ID1} (the old owner of the tag T) generates two random numbers rnd_1 and rnd_2. Then, it computes $A(i) = rnd_1(i) \oplus PRNG(K(i) \oplus R_{ID1}(i)) \oplus PRNG(K(i) \oplus R_{ID2}(i))$, $B(i) = rnd_2(i) \oplus PRNG(rnd_1(i) \oplus K(i))$ and $C(i) = PRNG(rnd_1(i) \oplus R_{ID1}(i)) \oplus PRNG(rnd_2(i) \oplus R_{ID2}(i))$ for $i = 1, \dots, 6$.

2) The reader R_{ID1} sends A, B and C to the tag T.

3) After receiving the messages A, B and C from the reader R_{ID1}, the tag T computes $rnd_1(i) = A(i) \oplus PRNG(K(i) \oplus R_{ID1}(i)) \oplus PRNG(K(i) \oplus R_{ID2}(i))$, $rnd_2(i) = B(i) \oplus PRNG(rnd_1(i) \oplus K(i))$ and $C'(i) = PRNG(rnd_1(i) \oplus R_{ID1}(i)) \oplus PRNG(rnd_2(i) \oplus R_{ID2}(i))$ for $i = 1, \dots, 6$. Then, the tag T verifies whether $C' \stackrel{?}{=} C$ is or not. In the case of equality, the tag T authenticates the reader R_{ID1}, updates $K(i)$ and $IDS(i)$ as $K^*(i) = Per(rnd_1(i), K(i)) \oplus K((i+1) \ mod \ 6)$ and $IDS^*(i) = Per(rnd_2(i), K(i)) \oplus K(i)$ respectively, and computes $D(i) = PRNG(K^*(i) \oplus IDS^*(i))$ for $i = 1, \dots, 6$.

4) The tag T sends D to the reader R_{ID1}.

5) After receiving the message D from the tag T, the reader R_{ID1} computes $K^*(i) = Per(rnd_1(i), K(i)) \oplus K((i+1) \ mod \ 6)$, $IDS^*(i) = Per(rnd_2(i), K(i)) \oplus K(i)$ and $D'(i) = PRNG(K^*(i) \oplus IDS^*(i))$ for $i = 1, \dots, 6$. Then, it verifies whether $D' \stackrel{?}{=} D$ is or not. In the case of equality, it authorizes the tag T and updates K and IDS as K^* and IDS^* respectively.

Ownership Delegation Phase:
In ownership delegation phase of the protocol, the reader R_{ID1} (which is the old owner of the tag T) wants to delegate all its rights over the tag T to the reader R_{ID2} by using the parameter called $ticket$. The old owner R_{ID1} and the tag T both compute $ticket = K_M \oplus EPC \oplus rnd_1 \oplus rnd_2$. Then, the reader R_{ID1} sends $ticket$, EPC, IDS and K through a secure channel to the reader R_{ID2} (the new owner R_{ID2} of the tag T). Ownership delegation steps shown in Figure 2 are as follows:

1) The reader R_{ID2} sends its identification R_{ID2} and a *Query* command to the tag T.

2) The tag T sends its IDS to the reader R_{ID2}.

3) The reader R_{ID2} generates one random number rnd_3, computes $E(i) = rnd_3(i) \oplus PRNG(K(i) \oplus R_{ID2}(i)) \oplus PRNG(K(i))$ and $F(i) = PRNG(ticket(i) \oplus rnd_3(i))$ for $i = 1, \dots, 6$.

4) The reader R_{ID2} sends E and F to the tag T.

5) After receiving the messages E and F from the reader R_{ID2}, the tag T computes $rnd_3(i) = E(i) \oplus PRNG(K(i) \oplus R_{ID2}(i)) \oplus PRNG(K(i))$ and $F'(i) = PRNG(ticket(i) \oplus rnd_3(i))$ for $i = 1, \dots, 6$. Then, the tag T verifies whether $F' \stackrel{?}{=} F$. In the case of equality, the tag T authenticates the new owner R_{ID2} and updates $K(i)$ and $IDS(i)$ as $K^*(i) = Per(rnd_3(i), K(i)) \oplus K((i+1) \ mod \ 6)$ and $IDS^*(i) = Per(rnd_3(i), K(i)) \oplus K(i)$ respectively as well as computing $G(i) = PRNG(K^*(i) \oplus IDS^*(i))$ for $i = 1, \dots, 6$.

6) The tag T sends G to the reader R_{ID2}.

7) After receiving the message G from the tag T, the reader R_{ID2} computes $K^*(i) = Per(rnd_3(i), K(i)) \oplus K((i+1) \ mod \ 6)$ and $IDS^*(i) = Per(rnd_3(i), K(i)) \oplus K(i)$ and $G'(i) = PRNG(K^*(i) \oplus IDS^*(i))$ for $i = 1, \dots, 6$.

Reader R_{ID1} (old owner)		**Tag T**
(K,K_m,EPC,IDS,R_{ID1},R_{ID2})		(K,K_m,EPC,IDS,R_{ID1},R_{ID2})

1. Generate rnd_1 and rnd_2
   Compute for $i = 1$ to 6:
   $A(i) = rnd_1(i) \oplus PRNG(K(i) \oplus R_{ID1}(i))$
   $\quad \oplus PRNG(K(i) \oplus R_{ID2}(i))$
   $B(i) = rnd_2(i) \oplus PRNG(rnd_1(i) \oplus K(i))$
   $C(i) = PRNG(rnd_1(i) \oplus R_{ID1}(i))$
   $\quad \oplus PRNG(rnd_2(i) \oplus R_{ID2}(i))$

$$\xrightarrow{2.\ A,B,C}$$

3. Compute for $i = 1$ to 6:
   $rnd_1(i) = A(i) \oplus PRNG(K(i) \oplus R_{ID1}(i))$
   $\quad \oplus PRNG(K(i) \oplus R_{ID2}(i))$
   $rnd_2(i) = B(i) \oplus PRNG(rnd_1(i) \oplus K(i))$
   $C'(i) = PRNG(rnd_1(i) \oplus R_{ID1}(i))$
   $\quad \oplus PRNG(rnd_2(i) \oplus R_{ID2}(i))$
   If $C = C'$:
   Compute for $i = 1$ to 6:
   $K^*(i) = Per(rnd_1(i), K(i)) \oplus K((i+1)\ mod\ 6)$
   $IDS^*(i) = Per(rnd_2(i), K(i)) \oplus K(i)$
   $D(i) = PRNG(K^*(i) \oplus IDS^*(i))$
   Update K as K^* and IDS as IDS^*

$$\xleftarrow{4.\ D}$$

5. Compute for $i = 1$ to 6:
   $K^*(i) = Per(rnd_1(i), K(i)) \oplus K((i+1)\ mod\ 6)$
   $IDS^*(i) = Per(rnd_2(i), K(i)) \oplus K(i)$
   $D'(i) = PRNG(K^*(i) \oplus IDS^*(i))$
   If $D = D'$:
   authenticate the tag;
   update K as K^* and IDS as IDS^*.

Fig. 1. Mutual authentication phase of Niu *et al.* authentication and ownership management protocol [11]

Then, it verifies whether $G' \stackrel{?}{=} G$ is or not. In the case of equality, it authorizes the tag T and updates K and IDS as K^* and IDS^* respectively.

Complete Ownership Transfer Phase:
The above mentioned ownership delegation transfer has not the property of the backward privacy since the old owner R_{ID1} holds the same values shared between the new owner R_{ID2} and the tag T. In order to address this pitfall, Niu *et al.* have proposed the complete ownership phase by using TTP in which all its rights over the tag T are transferred to the reader R_{ID2} as a new owner. This phase is shown in Figure 3 and described below:

1) TTP generates a random number rnd_4, calculates $H(i) = rnd_4(i) \oplus PRNG(K_{TTP}(i))$, $L(i) = PRNG(K_M(i) \oplus rnd_4(i))$ and $K^*_M(i) = PRNG(Per(K_M, rnd_4(i))$ for $i = 1$ to 6. Then, TTP updates K_M as K^*_M.

2) TTP sends K^*_M to the reader R_{ID2} (the new owner). It also sends H and L to the tag T.

3) Once the tag T received the messages H and L, it retrieves $rnd_4(i)$ as $H(i) \oplus PRNG(K_{TTP}(i))$ and computes $L'(i) = PRNG(K_M(i) \oplus rnd_4(i))$ for $i = 1, \ldots, 6$. Then, the tag T verifies whether $L' \stackrel{?}{=} L$. In the case of equality, it updates $K_M(i)$ as $K^*_M(i) = PRNG(Per(K_M(i), rnd_4(i)))$ for $i = 1$ to 6.

4) New owner R_{ID2} and the tag T go to mutual authentication phase. If mutual authentication succeeds, the ownership transfer has successfully been done.

III. Secret disclosure attack on Niu *et al.* authentication and management protocol

In this section, we show that it is possible to disclose secret parameters in Niu *et al.* authentication and management protocol efficiently. The main observation is that in this protocol the 96-bit parameters are divided into 16-bit strings and messages are generated using a 16-bit PRNG. On the other hand, several related works have shown that it is hard to achieve high security using small components [24]. Based on this observation, we present an attack to disclose secret parameters in this protocol.

In an off-line phase of the attack, the adversary creates a table TB and for $0 \le x < 2^{16}$ stores $(x, PRNG(x))$ in TB. Hence, given TB and $PRNG(x)$, it is possible to determine possible values of x. We use $(A, B, C)^{K^j}_{r^j_1, r^j_2}$ to show the messages based on the secret key $K = K^j$ and random values $rnd_1 = r^j_1$ and $rnd_2 = r^j_2$. The secret disclosure attack works as follows:

Reader R_{ID2} **(new owner)** $(K,IDS,ticket,EPC,R_{ID2})$		**Tag** T $(K,K_m,EPC,ticket,IDS,R_{ID2})$
	$\xrightarrow{1.\ Query,R_{ID2}}$	
	$\xleftarrow{2.\ IDS}$	
3. Generate rnd_3 Compute for $i=1$ to 6: $E(i)=rnd_3(i)\oplus PRNG(K(i))$ $\oplus PRNG(K(i)\oplus R_{ID2}(i))$ $F(i)=PRNG(ticket(i)\oplus rnd_3(i))$		
	$\xrightarrow{4.\ E,F}$	
		5. Compute for $i=1$ to 6: $rnd_3(i)=E(i)\oplus PRNG(K(i))$ $\oplus PRNG(K(i)\oplus R_{ID2}(i))$ $F'(i)=PRNG(ticket(i)\oplus rnd_3(i))$ If $F=F'$: Compute for $i=1$ to 6: $K^*(i)=Per(rnd_3(i),K(i))\oplus K((i+1)\ mod\ 6)$ $IDS^*(i)=Per(rnd_3(i),K(i))\oplus K(i)$ $G(i)=PRNG(K^*(i)\oplus IDS^*(i))$ Update K as K^* and IDS as IDS^*
	$\xleftarrow{6.\ G}$	
7. Compute for $i=1$ to 6: $K^*(i)=Per(rnd_3(i),K(i))\oplus K((i+1)\ mod\ 6)$ $IDS^*(i)=Per(rnd_3(i),K(i))\oplus K(i)$ $G'(i)=PRNG(K^*(i)\oplus IDS^*(i))$ If $G=G'$: authenticate the tag; update K as K^* and IDS as IDS^*.		

Fig. 2. Ownership delegation phase of Niu *et al.* authentication and ownership management protocol [11]

Reader R_{ID2} **(new owner)** (K,IDS,EPC,R_{ID2})		TTP $(K_M,K_{TTP},EPC,R_{ID2})$		**Tag** T (K_M,K_{TTP},EPC,IDS)
		1. Generate rnd_4 Compute for $i=1$ to 6: $H(i)=rnd_4(i)\oplus PRNG(K_{TTP}(i))$ $L(i)=PRNG(K_M(i)\oplus rnd_4(i))$ $K_M^*(i)=PRNG(Per(K_M,rnd_4(i)))$ Update K_M as K_M^*		
	$\xleftarrow{2.\ K_M^*}$		$\xrightarrow{2. H,L}$	
3. Store K_M^*				3. Compute for $i=1$ to 6: $rnd_4(i)=H(i)\oplus PRNG(K_{TTP}(i))$ $L'(i)=PRNG(K_M(i)\oplus rnd_4(i))$ If $L=L'$: Compute for $i=1$ to 6: $K_M^*(i)=PRNG(Per(K_M,rnd_4(i))$ Update K_M as K_M^*
4. Go to mutual authentication phase to complete ownership transfer phase.				4. Go to mutual authentication phase to complete ownership transfer phase.

Fig. 3. Complete ownership transfer phase of Niu *et al.* authentication and ownership management protocol [11]

1) Assume that the current state of the tag T and the reader R_{ID1} is (K^0, IDS^0, K^1, IDS^1) where K^0 and IDS^0 are the old key and pseudonym as well as K^1 and IDS^1 are the current key and pseudonym of the tag T. The adversary also has a table TB include $(x, PRNG(x))$ for $0 \leq x < 2^{16}$.
2) In the mutual authentication phase, the reader R_{ID1} (old owner of the tag T) sends $(A, B, C)^{K^1}_{r_1^1, r_2^1}$ to the tag T.
3) The tag T updates its state to (K^1, IDS^1, K^2, IDS^2) and sends $D(i) = PRNG(K^2(i) \oplus IDS^2(i))$ to the reader R_{ID1} for $i = 1, \ldots, 6$.
4) The adversary eavesdrops $(A, B, C)^{K^1}_{r_1^1, r_2^1}$ and D. The adversary also determines possible values of $K^2(i) \oplus IDS^2(i)$, for $i = 1, \ldots, 6$, using TB and D.
5) In the ownership delegation phase, the reader R_{ID2} (new owner of the tag T) sends its identification R_{ID2} and a *Query* command to the tag T.
6) The tag T sends its IDS^2 to the reader R_{ID2}.
7) The adversary eavesdrops IDS^2 and determines possible values of K^2, given the result of step 4.
8) The reader R_{ID2} generates a random number r_3^2 and computes $E(i) = r_3^2(i) \oplus PRNG(K^2(i) \oplus R_{ID2}(i)) \oplus PRNG(K^2(i))$ and $F(i) = PRNG(ticket(i) \oplus r_3^2(i))$ for $i = 1, \ldots, 6$ and sends E and F to the tag T, where $ticket = K_M \oplus EPC \oplus r_1^1 \oplus r_2^1$.
9) The adversary eavesdrops E and F and determines possible values of r_3^2 and $ticket$ using TB.
10) Once the tag T received the messages E and F, it authenticates the new owner R_{ID2} and updates $K(i)$ and $IDS(i)$ as below:

$$K^3(i) = Per(r_3^2(i), K^2(i)) \oplus K^2((i+1) \ mod \ 6);$$
$$IDS^3(i) = Per(r_3^2(i), K^2(i)) \oplus K^2(i).$$

11) The tag T computes $G_i = PRNG(K^3(i) \oplus IDS^3(i))$ for $i = 1, \ldots, 6$ and sends it to the reader R_{ID2}.
12) The adversary eavesdrops G and uses TB to determine possible values of $K^3(i) \oplus IDS^3(i)$ for $i = 1$ to 6.

Given information extracted in steps 7 and 9, the adversary have some possible values of $K^2(i)$ and $r_3^2(i)$ for $i = 1, \ldots, 6$. On the other hand, given $K^2(i)$, $K^2((i+1) \ mod \ 6)$ and $r_3^2(i)$, it is possible to determine $K^3(i)$ and $IDS^3(i)$. Hence, given the extracted information from step 12 of the attack, it is possible for the adversary to filter the wrong guesses for the extracted $K^2(i)$ and $r_3^2(i)$. Following the given attack, the adversary can extract the tag's secret parameters, i.e., K^2, IDS^2, K^3 and IDS^3. The major complexity of the attack is eavesdropping one run of the protocol and 2^{16} calls to a $PRNG$ function in an off-line mode. For simulation purposes, the proposed attack is performed 1000 times over Niu *et al.* protocol. Simulation results show that the success probability of the attacker is almost 1.

Moreover, each tag has a secrete parameter K_{TTP} which is shared between the tag and the TTP. This parameter is expected to be known only by the tag and the TTP, even not a legitimate owner. Now, we present an attack to retrieve this parameter by a legitimate old owner. In this attack, the old owner at the first generates the table TB and for $0 \leq x < 2^{16}$ stores $(x, PRNG(x))$ in TB. Next, assume that the current secret shared between the tag T and the owner is K_M^1 and the TTP wants that all rights over the tag T are transferred to the reader R_{ID2} from the reader R_{ID1}. The attack procedure which is performed by the reader R_{ID1} is as follows:

1) In the complete ownership transfer phase of the protocol, the TTP generates a random number r_4^1 and updates $K_M^1(i)$ as $K_M^2(i) = PRNG(Per(K_M^1(i), r_4^1(i))$ for $i = 1, \ldots, 6$.
2) The TTP calculates $H^1(i) = r_4^1(i) \oplus PRNG(K_{TTP}(i))$ and $L^1(i) = PRNG(K_M^1(i) \oplus r_4^1(i))$ for $i = 1, \ldots, 6$ and sends K_M^2 to the reader R_{ID2} (the new owner of the tag T) and H^1 and L^1 to the tag T.
3) The old owner R_{ID1}, as the adversary, eavesdrops H^1 and L^1.
4) Given the eavesdropped H^1 and L^1 and the table TB, the old owner R_{ID1} does as follows for all $i = 1$ to 6:
   - Computes $r_4^1(i) = PRNG^{-1}(L^1(i)) \oplus K_M^1(i)$;
   - Assigns $H^1(i) \oplus r_4^1(i)$ to $PRNG(K_{TTP}(i))$ and calculates $K_{TTP}(i)$ by looking up at TB.

Following the above passive attack, the old owner retrieves $r_4^1(i)$ and $K_{TTP}(i)$ for $i = 1, \ldots, 6$. Since K_{TTP} is the value that is needed as the permanent parameter to access the tag T, the old owner R_{ID1} finds a permanent control like TTP on the tag T in this attack. On the other hand, the old owner R_{ID1} knows K_M^1 and it can calculate $K_M^2(i) = PRNG(Per(K_M^1(i), r_4^1(i)))$ which is the secret parameter shared between the tag T and the new owner R_{ID2}. This information compromises the new owner privacy.

The complexity of the given attack is eavesdropping a sessions between the target tag and the new owner. The proposed attack is simulated experimentally. The success probability of the attacker is almost 1 for 1000 runs of the attack.

Now we present another attack that an adversary can follow to extract K_M and $PRNG(K_{TTP})$. Similarly, as the off-line phase of the attack, the adversary generates the table TB and for $0 \leq x < 2^{16}$ stores $(x, PRNG(x))$ in TB. Then, the adversary does as follows:

1) In complete ownership transfer phase of the protocol, the TTP generates a random number r_4^1 and updates $K_M^1(i)$ as $K_M^2(i) = PRNG(Per(K_M(i), r_4^1(i))$ for $i = 1, \ldots, 6$.
2) The TTP calculates $H^1(i) = r_4^1(i) \oplus PRNG(K_{TTP}(i))$ and $L^1(i) = PRNG(K_M^1(i) \oplus r_4^1(i))$ for $i = 1, \ldots, 6$ and sends K_M^2 to the reader R_{ID2} (the new owner of the tag T) and H^1 and L^1 to the tag.
3) The adversary eavesdrops and blocks H^1 and L^1.
4) The tag T will not authenticate the new owner R_{ID2} and the TTP generates another random number r_4^2 and updates $K_M^1(i)$ as $K_M^3(i) = PRNG(Per(K_M^1(i), r_4^2(i))$.
5) The TTP calculates $H^2(i) = r_4^2(i) \oplus PRNG(K_{TTP}(i))$ and $L^2(i) = PRNG(K_M^1(i) \oplus r_4^2(i))$ for $i = 1, \ldots, 6$ and sends K_M^3 to the new owner R_{ID2} and H^2 and L^2 to the tag T.

6) The adversary eavesdrops H^2 and L^2 and does as follows for all $i = 1$ to 6:
   a) For $j = 1, \ldots, 2^{16}$ does as follows:
      - $r_4^1(i) \longleftarrow j$;
      - $PRNG(K_{TTP}(i)) \longleftarrow H^1(i) \oplus r_4^1(i)$;
      - $K_M^1(i) \longleftarrow PRNG^{-1}(L^1(i)) \oplus r_4^1(i)$;
      - $r_4^2(i) \longleftarrow PRNG^{-1}(L^2(i)) \oplus K_M^1(i)$;
      - If $H^2(i) = r_4^2(i) \oplus PRNG(K_{TTP}(i))$, return $K_M^1(i)$ and $PRNG^{-1}(PRNG(K_{TTP}(i)))$

Following the above attack, the adversary retrieves $r_4^1(i)$, $r_4^2(i)$, $K_M^1(i)$ and $K_{TTP}(i)$ for $i = 1, \ldots, 6$. Since K_{TTP} is the value that is needed as the permanent parameter to access the tag T, the adversary R_{ID1} finds a permanent control like TTP on the tag T in this attack. On the other hand, the adversary extracted K_M^1 and r_4^2 and she can calculate $K_M^3(i) = PRNG(Per(K_M^1(i), r_4^2(i))$ which is the secret parameter shared between the tag T and the new owner R_{ID2}. This information compromises the new owner privacy.

The complexity of the given attack is eavesdropping two sessions between the target tag and the new owner and blocking one session. The proposed attack is simulated experimentally. The success probability of the attacker is almost 1 for 1000 runs of the attack.

IV. Desynchronization attacks on Niu *et al.* authentication and management protocol

In this section, we explain two different desynchronization attacks against Niu *et al.* authentication and management protocol on mutual authentication phase and ownership delegation phase.

A. Desynchronization attack on mutual authentication phase

As explained in Section II, in the mutual authentication phase of the Niu *et al.* protocol, the reader R_{ID1} generates two random numbers rnd_1 and rnd_2 and sends A, B and C to the tag T. Once the tag T received the messages A, B and C, it verifies the received values, updates K and IDS to K^* and IDS^* respectively and sends D to the reader. In addition, the designers stated that [11, p. 4, Sec. II. B] "*both the reader and the tag should maintain a copy of the old key and* IDS *to avoid desynchronization problems*". Now we present a desynchronization attack which works even with this assumption.

We use $(A, B, C)_{r_1^j, r_2^j}^{K^l}$ and $(D)_{r_1^j, r_2^j}^{K^l}$ to show the messages based on the secret key $K = K^l$ and random values $rnd_l = r_l^j$ and $rnd_2 = r_2^j$. The procedure of the proposed attack is as follows:

1) Assume that the current state of the tag T and the reader R_{ID1} is (K^0, IDS^0, K^1, IDS^1) where K^0 and IDS^0 are the old key and pseudonym as well as K^1 and IDS^1 are the current key and pseudonym of the tag T.
2) In the next mutual authentication phase, the reader R_{ID1} sends $(A, B, C)_{r_1^1, r_2^1}^{K^1}$ to the tag T.
3) The tag T updates its state to (K^1, IDS^1, K^2, IDS^2) and sends $(D)_{r_1^1, r_2^1}^{K^2}$ to the reader R_{ID1}.
4) The adversary blocks $(D)_{r_1^1, r_2^1}^{K^2}$.
5) Since the reader R_{ID1} does not receive the tag's feedback, it will assume that the tag T does not recognize K^1 and sends $(A, B, C)_{r_1^2, r_2^2}^{K^0}$ to the tag T.
6) However, the tag T has no record of K^0 and will not authenticate the reader R_{ID1} any more and the tag T and the reader R_{ID1} has been desynchronized.

One may argue that the reader R_{ID1} will try with K^1 once again. In this case, the adversary does as follows:

1) Assume that the current secrets of the tag T and the reader R_{ID1} is (K^0, IDS^0, K^1, IDS^1).
2) In the next mutual authentication phase, the reader R_{ID1} sends $(A, B, C)_{r_1^1, r_2^1}^{K^1}$ to the tag T.
3) The tag T updates its states to (K^1, IDS^1, K^2, IDS^2) and sends $(D)_{r_1^1, r_2^1}^{K^2}$ to the reader R_{ID1}.
4) The adversary stores $(A, B, C)_{r_1^1, r_2^1}^{K^1}$.
5) The reader R_{ID1} also updates its state to (K^1, IDS^1, K^2, IDS^2).
6) Since the designers stated that [11, p. 3, Sec. II. A] *"before either delegation or complete ownership transfer take place, mutual authentication is needed to verify the authority of all parties involved"*. So, the adversary blocks all the messages of the phase after authentication which can be delegation phase or complete ownership transfer phase. So once again the mutual authentication phase starts.
7) In the next mutual authentication phase, the reader R_{ID1} sends $(A, B, C)_{r_1^2, r_2^2}^{K^2}$ to the tag T.
8) The adversary stores $(A, B, C)_{r_1^2, r_2^2}^{K^2}$ and prevents the tag T to receive them.
9) Since the reader R_{ID1} does not receive the tag's feedback, it will assume that the tag T does not recognize K^2. So, The reader R_{ID1} sends $(A, B, C)_{r_1^3, r_2^3}^{K^1}$ to the tag T.
10) The tag T updates its state to (K^1, IDS^1, K^3, IDS^3) and sends $(D)_{r_1^3, r_2^3}^{K^3}$ to the reader R_{ID1}.
11) The reader R_{ID1} also updates its state to (K^1, IDS^1, K^3, IDS^3).
12) The adversary sends $(A, B, C)_{r_1^1, r_2^1}^{K^1}$ to the tag T.
13) The tag T sends $(D)_{r_1^1, r_2^1}^{K^2}$ to the expected reader and also updates its state to (K^1, IDS^1, K^2, IDS^2).
14) The adversary prevents the reader R_{ID1} to receive $(D)_{r_1^1, r_2^1}^{K^2}$.
15) The adversary sends $(A, B, C)_{r_1^2, r_2^2}^{K^2}$ to the tag T.
16) The tag T sends $(D)_{r_1^1, r_2^1}^{K^4}$ to the expected reader and also updates its state to (K^2, IDS^2, K^4, IDS^4).
17) The adversary prevents the reader R_{ID1} to receive $(D)_{r_1^1, r_2^1}^{K^4}$.

After the above attack, the reader has K^1, IDS^1, K^3, IDS^3 as its records of secret parameters while the tag T has K^2, IDS^2, K^4 and IDS^4. It is clear that, after the given attack neither of the tag's records for secret parameters matches the reader's records for secret parameters. Hence, the tag and the reader have been desynchronized. Although it may be possible to contact the trusted third party(TTP) to re-synchronize the tag, but this attack shows that the given protocol does not satisfy the designers expectation. The complexity of the attack is a few runs of the protocol while the success probability is almost 1.

B. Desynchronization attack on ownership delegation phase

In the ownership delegation phase of the protocol, The old owner R_{ID1} sends $ticket$, EPC, IDS and K through a secure channel to the new owner (the reader R_{ID2}). The reader R_{ID2} generates a random number rnd_3 and computes E and F using K and rnd_3. Then, the reader R_{ID2} sends E and F to the tag T. Once the tag T received the messages E and F, it verifies the received values and updates K and IDS to K^* and IDS^* using rnd_3. Then, the tag computes G using K^* and IDS^* and sends it to the reader R_{ID2}.

In the protocol, the parameter $G(i)$ for $i = 1, \cdots, 6$ is not dependent on $K^*(i)$, $IDS^*(i)$ and the random number $rnd_3(i)$ selected by the new owner R_{ID2}. It is computed only using the secret key $K(i)$ as shown as follows:

$$\begin{aligned} G(i) &= PRNG(\ K^*(i) \oplus IDS^*(i)\ ) \\ &= PRNG(\ \ Per(rnd_3(i), K(i)) \ \oplus\ K((i+1)\ mod\ 6) \\ &\oplus\ \ Per(rnd_3(i), K(i)) \ \oplus\ K(i)\ \ ) \\ &= PRNG(\ K((i+1)\ mod\ 6) \ \oplus\ K(i)\ ); \end{aligned}$$

In other word, the response of E and F which are computed by rnd_3 and K, i.e, G, is computed by K not K^* and rnd_3. Now, we show that this property can be used by the attacker to perform desynchronization attack in the ownership delegation phase of the protocol. We use $(E, F)^{K^j}_{r_3^j}$ and $(G)^{K^j}$ to show the messages based on the secret key K^j and the random value $rnd_3 = r_3^j$. We assume that both parties are synchronized in state (K^1, IDS^1). The procedure of the proposed attack is as follows:

1) To update the tag T by the new owner (the reader R_{ID2}) for the first time, the new owner sends $(E, F)^{K^1}_{r_3^1}$ to the tag T.
2) The tag T updates its states to (K^2, IDS^2) where $K^2(i) = Per(r_3^1(i), K^1(i)) \oplus K^1((i+1)\ mod\ 6)$ and $IDS^2(i) = Per(r_3^1(i), K^1(i)) \oplus K^1(i)$. Then, the tag T sends $(G)^{K^1}$ to the reader R_{ID2}.
3) The adversary prevents the reader R_{ID2} to receive $(G)^{K^1}$ and stores it.
4) Since the reader R_{ID2} does not receive the tag's feedback, it will assume that the tag T does not receive $(E, F)^{K^1}_{r_3^1}$. Therefore, the reader R_{ID2} chooses another random number r_3^2 and sends $(E, F)^{K^1}_{r_3^2}$ to the tag T again.
5) The adversary prevents the tag T to receive $(E, F)^{K^1}_{r_3^2}$ and sends $(G)^{K^1}$ (which is stored by the adversary in step 3) to the reader R_{ID2}.
6) Note that according to the property G which is not dependent on the random number selected by the reader, the reader R_{ID2} detects the validity of $(G)^{K^1}$. So, the reader R_{ID2} updates its state to (K^3, IDS^3) where $K^3(i) = Per(r_3^2(i), K^1(i)) \oplus K^1((i+1)\ mod\ 6)$ and $IDS^3(i) = Per(r_3^2(i), K^1(i)) \oplus K^1(i)$.

After the above attack, the reader R_{ID2} has K^2 and IDS^2 as its records of secret parameters while the tag T has K^3 and IDS^3. It is clear that, these two states are not the same and the adversary succeeds in performing desynchronization attack between the tag T and the new owner R_{ID2}. Although it may be possible to contact the old owner R_{ID1} to re-synchronize the tag, but this attack shows that the given protocol does not satisfy the designers expectation. The complexity of the attack is only three runs of the protocol and the probability of a successful de-synchronization attack is equal to 1.

V. Conclusion

In this paper, we scrutinized the security of the mutual authentication and ownership transfer management protocol proposed by Niu *et al*. Precisely, we present secret disclosure and desynchronization attacks against the protocol with the complexity of a few runs of the protocol and the success probability of almost 1. This paper shows that the need to secure EPC-C1G2 complaint protocols is still unmet and the new secure protocols must be designed.

References

[1] G. Avoine and X. Carpent. Yet another ultralightweight authentication protocol that is broken. In J. Hoepman and I. Verbauwhede, editors, *RFIDSec 2012*, volume 7739 of *Lecture Notes in Computer Science*, pages 20–30. Springer, 2013.

[2] G. Avoine, X. Carpent, and B. Martin. Privacy-friendly synchronized ultralightweight authentication protocols in the storm. *J. Network and Computer Applications*, 35(2):826–843, 2012.

[3] N. Bagheri and M. Safkhani. Secret disclosure attack on kazahaya, a yoking-proof for low-cost RFID tags. *IACR Cryptology ePrint Archive*, 2013:453, 2013.

[4] N. Bagheri, M. Safkhani, P. Peris-Lopez, and J. E. Tapiador. Weaknesses in a new ultralightweight RFID authentication protocol with permutation - RAPP. *Security and Communication Networks*, 7(6):945–949, 2014.

[5] H.-Y. Chien. Sasi: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Trans. Dependable Sec. Comput.*, 4(4):337–340, 2007.

[6] Class-1 generation 2 UHF air interface protocol standard version 1.2.0, Gen2, 2008. http://www.epcglobalinc.org/standards/.

[7] P. D'Arco and A. D. Santis. Weaknesses in a recent ultra-lightweight rfid authentication protocol. In S. Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pages 27–39. Springer, 2008.

[8] P. D'Arco and A. D. Santis. On ultralightweight RFID authentication protocols. *IEEE Trans. Dependable Sec. Comput.*, 8(4):548–563, 2011.

[9] R. Doss, W. Zhou, and S. Yu. Secure RFID tag ownership transfer based on quadratic residues. *IEEE Transactions on Information Forensics and Security*, 8(2):390–401, 2013.

[10] A. Falahati and H. Jannati. All-or-nothing approach to protect a distance bounding protocol against terrorist fraud attack for low-cost devices. *Electronic Commerce Research*, 15(1):75–95, 2015.

[11] S. J. Haifeng Niu, Eyad Taqieddin. EPC Gen2v2 RFID Standard Authentication and Ownership Management Protocol . *IEEE Transactions on Mobile Computing*, ?(?):?, 2014.

[12] Information technologyRadio frequency identification for item management. Part 6:parameters for air interface communications at 860 MHz to 960MHz .http://www.iso.org. 2005.

[13] H. Jannati and A. Falahati. Cryptanalysis and enhancement of a secure group ownership transfer protocol for RFID tags. In C. K. Georgiadis, H. Jahankhani, E. Pimenidis, R. Bashroush, and A. Al-Nemrat, editors, *ICGS3/e-Democracy 2011*, volume 99 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 186–193. Springer, 2011.

[14] H. Liu, H. Ning, Y. Zhang, D. He, Q. Xiong, and L. T. Yang. Grouping-proofs-based authentication protocol for distributed RFID systems. *IEEE Trans. Parallel Distrib. Syst.*, 24(7):1321–1330, 2013.

[15] S. B. Miles, S. E. Sarma, and J. R. Williams. *RFID technology and applications*. Cambridge University Press, 2011.

[16] B. Niu, X. Zhu, H. Chi, and H. Li. Privacy and authentication protocol for mobile RFID systems. *Wireless Personal Communications*, 77(3):1713–1731, 2014.

[17] P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda. Emap: An efficient mutual-authentication protocol for low-cost RFID tags. In R. Meersman, Z. Tari, and P. Herrero, editors, *OTM Workshops (1)*, volume 4277 of *Lecture Notes in Computer Science*, pages 352–361. Springer, 2006.

[18] P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda. Advances in ultralightweight cryptography for low-cost rfid tags: Gossamer protocol. In *WISA*, pages 56–68, 2008.

[19] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. Lmap: A real lightwight mutual authentication protocol for low-cost rfid tags. In *Proceedings of RFIDSec06 Workshop on RFID Security*, Graz,Austria, 12-14 July 2006.

[20] P. Peris-Lopez, A. Orfila, J. C. Hernandez-Castro, and J. C. A. van der Lubbe. Flaws on RFID grouping-proofs. guidelines for future sound protocols. *J. Network and Computer Applications*, 34(3):833–845, 2011.

[21] P. Peris-Lopez, A. Orfila, A. Mitrokotsa, and J. C. A. van der Lubbe. A comprehensive RFID solution to enhance inpatient medication safety. *I. J. Medical Informatics*, 80(1):13–24, 2011.

[22] R. C.-W. Phan. Cryptanalysis of a new ultralightweight RFID authentication protocol - sasi. *IEEE Transactions on Dependable and Secure Computing*, 6(4):316–320, 2009.

[23] M. Safkhani, N. Bagheri, and M. Naderi. Strengthening the security of EPC C-1 G-2 RFID standard. *Wireless Personal Communications*, 72(2):1295–1308, 2013.

[24] M. Safkhani, N. Bagheri, and M. Naderi. A note on the security of is-rfid, an inpatient medication safety. *I. J. Medical Informatics*, 83(1):82–85, 2014.

[25] M. Safkhani, P. Peris-Lopez, N. Bagheri, M. Naderi, and J. C. H. Castro. On the security of tan et al. serverless RFID authentication and search protocols. In J. Hoepman and I. Verbauwhede, editors, *RFIDSec 2012*, volume 7739 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2013.

[26] S. Sundaresan, R. Doss, S. Piramuthu, and W. Zhou. A robust grouping proof protocol for RFID EPC C1G2 tags. *IEEE Transactions on Information Forensics and Security*, 9(6):961–975, 2014.

[27] S. Sundaresan, R. Doss, S. Piramuthu, and W. Zhou. Secure tag search in RFID systems using mobile readers. *IEEE Trans. Dependable Sec. Comput.*, 12(2):230–242, 2015.

[28] C. C. Tan, B. Sheng, and Q. Li. Secure and serverless RFID authentication and search protocols. *IEEE Transactions on Wireless Communications*, 7(4):1400–1407, 2008.

[29] Y. Tian, G. Chen, and J. Li. A new ultralightweight RFID authentication protocol with permutation. *IEEE Communications Letters*, 16(5):702–705, 2012.

[30] A. Wickboldt and S. Piramuthu. Patient safety through RFID: vulnerabilities in recently proposed grouping protocols. *J. Medical Systems*, 36(2):431–435, 2012.