Security Flaws in Two Recently Proposed RFID Authentication Protocols

Sonam Devgan Kaul
Department of Mathematics
Amity University
Noida, India.

Amit K. Awasthi
School of Applied Sciences
Gautam Buddha University
Greater Noida, India.

ABSTRACT
On the basis of Vaudenay's untraceability model, this paper describes cryptanalyses of recently proposed Zhuang et al.'s ultralightweight RFID authentication protocol for low cost tags R^2AP and Dehkordi and Farzaneh's improved hash based RFID mutual authentication protocol. This paper formally demonstrates that R^2AP is insecure and does not attain even *Narrow Forward* privacy level of security. Additionally, R^2AP protocol is traceable and suffers from impersonation attack. Also Dehkordi and Farzaneh's proposed protocol is impractical formally as it does not attain even *Narrow Forward* privacy level of security.

Keywords
RFID; privacy; authentication; cryptanalysis

1. INTRODUCTION
Radio frequency identification (RFID) is a technology that uses radio waves to automatically recognize an individual, an article or an object without any physical contact with the same and suffers from severe resource constraints. Automatic identification RFID system generally consists of a set of low cost RFID tags, a few RFID readers and a secure back-end server. The communication channel between server and reader is generally wired, which is assumed to be secure, while reader and tag communicates through a radio frequency wireless channel, which is insecure and can be intercepted and read by an eavesdropper or an adversary can modify the message in such a way that legitimate recipient does not detect the manipulation [1]. In RFID authentication protocol server verifies the identity of the tag and retrieves the detailed information of the corresponding object to which the tag is incorporated via radio signals.

In 2006, Peris-Lopez et al. proposed a family of ultralightweight RFID mutual authentication protocols, LMAP [2] (Lightweight mutual authentication protocol), M^2AP [3] (Minimalist mutual authentication protocol) and EMAP [4] (Efficient mutual authentication protocol), which uses only XOR ($\oplus$), OR ($\vee$), AND ($\wedge$) and addition modulo n ($+_n$) operations. But in 2007, Li and Wang [5] identify desynchronization and fully disclosure attack on these protocols. In 2007, Chien [6] introduced ultralightweight mutual authentication protocol to provide strong authentication and strong integrity *SASI*, which uses bitwise operations as well as rotation (*Rot*) operation. Unfortunately, in 2009, Phan [7] and Sun et al. [8] found that his protocol is also vulnerable to desynchronization attack, DOS attack, traceability attack, replay attack and fully disclosure attack. In 2008, Lopez et al. [9] proposed Gossamer protocol for low cost RFID tags using non triangular functions: RotBits and MixBits. But in 2009, Bilal et al. [10] showed memory and computational exhaustive attack, de-synchronization attack and replay attack against gossamer protocol [9].

In 2010, Kulseng et al. [11] proposed lightweight mutual authentication and ownership transfer protocol for RFID system using minimalistic cryptography functions such as physically unclonable functions (PUF) and linear feedback shift registers (LFSR) and reduce number of gates in tags. However in 2012, Yang et al. [12] presented tracking attack, disclosure attack and authentication attack against [11]. In 2012, Tian et al. [13] proposed a new ultralightweight RFID authentication protocol *RAPP*, which uses XOR, left rotation as well as new bitwise permutation operation. However in 2013, Avoine and Xavier [14] pointed out traceability attack on [13], by describing Hamming weight of output of permutation operation is same as the input parameter.

Recently, Zhuang et al. [15] proposed a new ultra-lightweight RFID authentication protocol for low cost tags R^2AP , which is based on reconstruction bitwise operation. By extending Juels and Weis [16] untraceability model, Zhuang et al. claimed that R^2AP is secure and effective protocol that can be implemented on low cost tags. Also Cho et al. proposed a hash-based radio-frequency identification tag mutual authentication protocol [17] on the basis of brute-force attack cost and retrieval cost. However in 2014, Dehkordi and Farzaneh [18] demonstrated traffic analysis and tag/reader impersonation attacks on [17] and proposed the countermeasures to thwart the security threats and to minimize the computation cost.

In this paper, the author cryptanalyze two recently proposed Zhuang et al.'s ultralightweight RFID authentication protocol for low cost tags R^2AP [15] and Dehkordi and Farzaneh's improved hash based RFID mutual authentication protocol [18] on the basis of Vaudenay's untraceability model [19]. The author formally demonstrate that R^2AP is insecure and does not attain even *Narrow Forward* privacy level of security. In addition, R^2AP protocol is traceable and suffers from impersonation attack. Also the author point out that Dehkordi and Farzaneh's protocol is impractical as it does not attain even *Narrow Forward* privacy level of security.

1.1 Organization
The rest of the paper is organized as follows: Definitions are described in Section 2. Vaudenay Privacy Model is presented in Section 3 Review of Zhuang, Zhu and Chang's Protocol and its cryptanalysis is given in Section 4. Review of Dehkordi and Farzaneh's Protocol and its cryptanalysis is presented in Section 5. Finally, we conclude the paper in Section 6.

2. DEFINITION
Reconstruction Function
Reconstruction function [15] of two l-bit strings A and B is:

$$Rec(A,B)=c_{l-1}c_{l-2}...c_0=F(a_i,b_i)$$

where

$A=a_{l-1}a_{l-2}...a_0$, a_iÎ{0,1}
$B=b_{l-1}b_{l-2}...b_0$, b_jÎ{0,1}

and

$$F(a_i,b_i)=\begin{cases} a_{(i-1) \bmod l} & \text{if } a_i > b_i \\ b_{(i-1) \bmod l} & \text{if } a_i < b_i \\ a_i & \text{if } a_i = b_i \end{cases}$$

For example:

1. If *A* = 01010110 and *B* = 11001010, then *Rec*(*A, B*) = 11000110.
2. If *A* = 10010110 and *B* = 00111001, then *Rec*(*A, B*) = 00110100.

Bitwise operation reconstruction has following main features:

1. Hamming weight unpredictability: An adversary can't be able to predict hamming weight of output of reconstruction bitwise operation of *A* and *B*; *wt*(*Rec*(*A, B*)) as neither *wt*(*Rec*(*A, B*)) = *wt*(*A*) nor *wt*(*Rec*(*A, B*)) = *wt*(*B*).
2. Irreversibility: Reconstruction bitwise operation behaves like one way function. Given the value of *Rec*(*A, B*) and one of the parameter either *A* or *B*, then it is infeasible for an adversary to find the value of another parameter.
3. Effectiveness: Time complexity to compute reconstruction bitwise operation is same as in bitwise XOR operation. Implementation of reconstruction bitwise operation only requires one traversal and 2*l* comparison operation for each parameter.

3. VAUDENAY PRIVACY MODEL

To design privacy preserving, secure and efficient RFID authentication protocol, in 2007, Vaudenay [19] proposed simulation based comprehensive RFID security and privacy model in which adversary's capabilities are classified into *{Wide, Narrow}* × *{Strong, Destructive, Forward, Weak}* classes. In this section, we present the RFID system set up procedures, adversary oracle model and security as well as privacy experiment:

3.1 System Model

RFID system of Vaudenay model composed of following three algorithms:

1. $SetupReader(1^{\lambda}) \rightarrow (pk_r, sk_r)$: This probabilistic polynomial-time algorithm initialize the reader by generating public/private key pair of the reader (pk_r, sk_r) depending upon the security parameter λ.
2. $SetupTag(ID, pk_r) \rightarrow (K, IS_{ID})$:This probabilistic polynomial-time algorithm initialize the tag with unique identifier ID. Using its unique ID, algorithm generates its secret key K and its initial state IS_{ID}. Initial state IS_{ID} is saved inside the tag while the pair (ID, K) is stored in the server's database.
3. $IdentProtocol(\pi)$ *:* Execute a polynomial time interactive protocol π between reader and tag. If the tag is legitimate then reader accepts it and produces an output ID otherwise output is $\perp$.

3.2 Adversarial Capabilities

An adversary A of [19] is able to interact with the RFID system and play polynomial number of games with the set of tags by sending the following queries to an oracle o:

1. $CreateTag^b(ID)$: An adversary *A* is able to create legitimate as well as fake tag with unique identity *ID* corresponding to *b*=1 or *b*=0 respectively.
2. $DrawTag(distr) \rightarrow (vtag_1, b_1, ..., vtag_n, b_n)$: An adversary has access to polynomial number of tags and randomly draw a set of free tags between all the existing ones with given probability distribution *distr*. New pseudonym $vtag_i$ (virtual ID) is allotted to each drawn tag and for legitimate identity of tag, b_i=1 otherwise b_i=0.
3. *Free*(*vtag*): An adversary reverts the drawn tag *vtag* to the set of free tags and no longer be able to call *vtag* in its oracles.
4. *Launch*(π): This deterministic oracle authorized reader to initiate a new session of the protocol π between *R* and *T*.
5. *SendReader*(*m*,π)→*m*'
   An adversary *A* may send a message *m* of his choice to the reader in the protocol execution π which output *m*'.
6. *SendTag*(*m*,*vtag*)→*m*'
   An adversary *A* may send any message *m* to the drawn tag *vtag* which responds with *m*'.
7. *Result*(π): This oracle outputs 1 to indicate the session of the protocol π is successfully executed otherwise it outputs 0.
8. *Corrupt*(*vtag*): This oracle outputs the current internal state of the drawn tag *vtag*.

3.3 Adversary Classes

An adversary A's capability is categorized in to following privacy classes based on A's access to Corrupt or Result oracles:

1. **Strong class adversary** has full access to all the above oracles at any time without any restriction.
2. **Destructive class adversary** has no ability to access any other oracle query on vtag after querying *Corrupt(vtag)* oracle.
3. **Forward class adversary** can access no oracle except *Corrupt()* oracle only once.
4. **Weak class adversary** is allowed to access all the oracles except *Corrupt()* oracle.

5. **Narrow class adversary** has no access to *Result* oracle query while **Wide adversary** can access *Result* oracle.

Thus obviously we have:

$$Weak \subseteq Forward \subseteq Destructive \subseteq Strong$$

3.4 Security and Privacy Notions

Here, we discuss security notions in which non legitimate tags are rejected by the server as well as the privacy notions which present the untraceablity of tags.

3.4.1 Definition: Tag Authentication

An RFID system attains tag authentication if the success probability of strong adversary *A* for identifying a non-legitimate tag is at most negligible. Privacy is explained by means of the blinder *B* and trivial adversary. *B* simulates *Launch*, *SendReader*, *SendTag* and *Result* oracles without having any knowledge of real secret keys. Also *B* sees input/output of any oracle query made by *A*. RFID system is said to be secure if the success probability of an adversary to differentiate real RFID system from the blinder *B* is at most negligible.

3.4.2 Definition: Trivial Adversary

An adversary *A* is said to be trivial if there exist a blinded adversary A^B (who response via the blinder) such that

$$|P\ r(A\ succeeds) - P\ r(A^B\ succeeds)| < \epsilon(\lambda)$$

3.5 Privacy Experiment EXP_A^{Priv}

Let *P* be the adversary class such that $P \in \{Wide, Narrow\} \cup \{Strong, Destructive, Forward, Weak\}$. Privacy game is defined between the adversary *A* and the challenger *C* and composed of following three phases:

1. **Learning Phase**: Foremost *C* setup the RFID system. An adversary *A* interacts with the system and inquiries oracle queries according to her class *P*. Real oracle queries may be analyzed by the adversary *A* or the blinder *B* may simulate the *Launch*, *SendReader*, *SendT ag* and *Result* oracles.

2. **Challenge Phase:** An adversary *A* obtains the hidden table, which maps *vtag* to identity of the tag. An adversary *A* gets access to two uncorrupted challenge tags and then randomly select any one from them.

3. **Guess Phase**: Eventually, an adversary *A*'s privacy game simulation comes to an end and *A* is expected to produce 1 if he succeeds otherwise 0.
Privacy experiment EXP_A^{Priv} wins if *A* returns 1.

3.5.1 Definition: Privacy

An RFID system is said to be P-private if $\forall A \in P$, if

$$|EXP_A^{P\ riv} - EXP_{A\ B}^{P\ riv}| < \epsilon(\lambda)$$

Table 1. Notations

S	Server
R	Reader
T	Tag
A	Adversary
ID	Unique identity of i^{th} tag
K_1,K_2,K_3	Secret keys of i^{th} tag
IDS	Index pseudonym mechanism
r_1,r_2,r_3	Random string of *l* bits generated by pseudo random generator
h(.)	Secure one way hash function $h(.):\{0,1\}^* \rightarrow \{0,1\}^l$
wt(x)	Hamming weight of string *x*
Rot(x,y)	Circular left rotate string *x* by *wt(y)* bits
Rec (x,y)	Reconstruction bitwise operation of *x* with *y*
$\oplus$	Bitwise XOR operation
//	Concatenation operation

4. ZHUANG, ZHU AND CHANG PROTOCOL

In 2014, Zhuang et al. proposed a new ultralightweight RFID authentication protocol for low cost tags R^2AP [15], which is based on reconstruction bitwise operation and index pseudonym mechanism *IDS*. By extending Juels and Weis untraceability model [16], Zhuang et al. showed that R^2AP is secure and effective protocol that can be implemented on low cost tags. The notations used throughout the paper are sum up in Table 1.

4.1 Protocol Description

As described in Table 2, to initialize tags, reader assigns identity $ID \in \{0, 1\}^l$, index pseudonym $IDS \in \{0, 1\}^l$, and three secret keys $K_1, K_2, K_3 \in \{0, 1\}^l$ to each tag stores (ID, IDS, K_1, K_2, K_3) in tag's memory as well as in reader's database. Database system also stores last session index pseudonym IDS^O, secret keys K_1^O, K_2^O, K_3^O and random numbers r_1^O and r_2^O to keep the synchronization state. In R^2AP protocol reader and tag follow the following steps to mutually authenticate each other and to update the secret parameters:

1. Reader sends 'Hello' message to tag to start the communication and tag responds with its *IDS*.

2. Corresponding to received *IDS*, reader finds secret parameters of the requested tag with complexity level *O*(1). Then reader generates a pseudo random number $r_1 \in \{0, 1\}^l$ and transmits the message *{A, B}* to the tag in an insecure communication channel, where

$$A = Rec(K_1, K_2) \oplus r_1$$

$$B = Rot(Rec(K_2, r_1), Rec(K_3, r_1)) \oplus Rot(r_1, r_1)$$

3. After receiving the message *{A, B}*, tag foremost extracts $r_1^* = A^* \oplus Rec(K_1, K_2)$ and then confirm the exactness of r_1 by verifying computed

   $B^* = Rot(Rec(K_2, r_1^*), Rec(K_3, r_1^*)) \oplus Rot(r_1^*, r_1^*)$

   with the received *B*. If it finds incorrect, the authentication request is rejected else the tag response with the message *C*, where

   $C = Rec(Rec(K_2, K_3), Rec(r_1, K_1)) \oplus ID.$

4. Now reader verifies the legality of the tag by $ID^* = C^* \oplus Rec(Rec(K_2, K_3), Rec(r_1, K_1))$ which reader extracts from the received request message *C* and match it with the *ID*, that is stored in the database. Only after confirmation, reader generates $r_2 \in \{0, 1\}^l$ and sends the message *{D, E}* to tag, where

   $D = Rec(r_1, K_3) \oplus Rec(K_1, K_3) \oplus r_2$

   $E = Rot(Rec(K_2, r_2), Rec(K_2, r_1)) \oplus Rot(r_2, r_2)$

5. Eventually upon receiving the message *{D, E}*, tag extracts $r_2^* = D^* \oplus Rec(r_1, K_3) \oplus Rec(K_1, K_3)$ to verify the validity of the requested message *E* by computing

   $E^* = Rot(Rec(K_2, r_2^*), Rec(K_2, r_1)) \oplus Rot(r_2^*, r_2^*)$

   After successful mutual authentication, tag and reader update its secret parameters by $(IDS^N, K_1^N, K_2^N, K_3^N)$,

   $IDS^N = Rec(IDS \oplus r_2, K_3) \oplus K_1$

   $K_1^N = Rec(r_2, r_1) \text{Å} K_2$

   $K_2^N = Rec(K_2, r_1 \text{Å} r_2) \text{Å} K_3$

   $K_3^N = Rec(K_2, K_3) \text{Å} r_1$

Table 2: Zhuang, Zhu and Chang Protocol

Reader $(ID,\ IDS,\ K_1,\ K_2,\ K_3, IDS^O,\ K_1^O,\ K_2^O, K_3^O,\ r_1^O,\ r_2^O)$	**Tag** $(ID,\ IDS,\ K_1,\ K_2,\ K_3)$
$\xrightarrow{Hello}$	
	$\xleftarrow{IDS}$
Generate r_1	
Compute $A = Rec(K_1, K_2) \oplus r_1$	
$B = Rot(Rec(K_2, r_1), Rec(K_3, r_1)) \oplus Rot(r_1, r_1)$	
$\xrightarrow{A,B}$	
	Compute $r_1^* = A^* \oplus Rec(K_1, K_2)$
	$B^* = Rot(Rec(K_2, r_1^*), Rec(K_3, r_1^*)) \oplus Rot(r_1^*, r_1^*)$
	Verify $B^{*?}_{=} B$
	Compute $C = Rec(Rec(K_2, K_3), Rec(r_1, K_1)) \oplus ID$
	$\xleftarrow{C}$
$ID^* = C^* \oplus Rec(Rec(K_2, K_3), Rec(r_1, K_1))$	
Verify $ID^{*?}_{=} ID$	
Generate r_2	
Compute $D = Rec(r_1, K_3) \oplus Rec(K_1, K_3) \oplus r_2$	
$E = Rot(Rec(K_2, r_2), Rec(K_2, r_1)) \oplus Rot(r_2, r_2)$	
$\xrightarrow{D,E}$	
	Compute $r_2^* = D^* \oplus Rec(r_1, K_3) \oplus Rec(K_1, K_3)$
	$E^* = Rot(Rec(K_2, r_2^*), Rec(K_2, r_1)) \oplus Rot(r_2^*, r_2^*)$
	Verify $E^{*?}_{=} E$
Update secret parameters	
$IDS^N = Rec(IDS \oplus r_2, K_3) \oplus k_1,\ K_1^N = Rec(r_2, r_1) \oplus K_2$	
$K_2^N = Rec(K_2, r_1 \oplus r_2) \oplus K_3$ and $K_3^N = Rec(K_2, K_3) \oplus r_1$	

4.2 Cryptanalysis

We cryptanalyze Zhuang et al. ultralightweight RFID authentication protocol for low cost tags R^2AP [15] on the basis of Vaudenay untraceability model [19]. We demonstrate by formal security analysis that their RFID authentication scheme is insecure and does not attain even *Narrow Forward* privacy level of security. Thus author's claims of strong privacy level are proven to be wrong. In addition R^2AP protocol is traceable and suffers from impersonation attack. Security flaws of Zhuang et al.'s protocol are described as follows:

Theorem 4.1 R^2AP protocol does not provide even *Narrow Forward* privacy.

Proof: R^2AP protocol only achieves at max Weak privacy level. An adversary A get access to two uncorrupted tags $vtag_0$ and $vtag_1$ as its challenge tags and then randomly chooses $vtag_b$, $b \in \{0,1\}$ among them. A analyzes the protocol run between and $vtag_b$ and evaluates all oracles on $vtag_b$. A calls *Free* oracle query to free the chosen tag. Finally an adversary calls $Corrupt(vtag_x)$ on any one of the challenge tags to get K_{1x}, K_{2x}, K_{3x}. Now A is able to compute

$r_{1x} = A \oplus Rec(K_{1x}, K_{2x})$
$B_x = Rot(Rec(K_{2x}, r_{1x}), Rec(K_{3x}, r_{1x})) \oplus Rot(r_{1x}, r_{1x})$

by means of corrupted keys. If $B_x = B$, then $x = b$ otherwise $x = |1-b|$, i.e. corrupted tag by an adversary is having identity ID_b or ID_{1-b} respectively. Thus an adversary is able to trace the tag. Hence R^2AP protocol does not provide *Narrow Forward* privacy.

$CreateTag(ID_0)$ *and* $Create\ Tag(ID_1)$
Choose $b \in \{0,1\}$
$vtag_b \leftarrow DrawTag(ID_b)$
$\pi \leftarrow Launch$
$IDS \leftarrow SendTag(Init, vtag_b)$
$A,B \leftarrow SendReader(IDS,\pi)$
$C \leftarrow SendTag(A,B, vtag_b)$
$Free(vtag_b)$
The session is incomplete.
$vtag_x \leftarrow DrawTag(ID_x)$
$K_{1_x}, K_{2_x}, K_{3_x} \leftarrow Corrupt(vtag_x)$
Oracle query comes to an end.
$r_{1_x} = A \oplus Rec(K_{1_x}, K_{2_x})$
$B_x = Rot(Rec(K_{2_x}, r_{1_x}), Rec(K_{3_x}, r_{1_x})) \oplus Rot(r_{1_x}, r_{1_x})$
If $B_x = B$ then $x = b$ otherwise $x = |1-b|$

Theorem 4.2: R^2AP protocol is traceable.

Proof: Based on the traceability definition of [19], an adversary A has not given any permission to call *Corrupt* and *Result* oracles. Still R^2AP protocol is traceable just by means of active and passive attacks. An adversary A get access to two uncorrupted tags $vtag_0$ and $vtag_1$ as its challenge tags and then randomly chooses $vtag_b$, $b \in \{0, 1\}$ among them. A analyzes the protocol run between R and $vtag_b$ and evaluates all oracles on $vtag_b$. An adversary A chooses two random numbers r_{A1} and r_{A2} and sends them to tag to stop the tag for key updation. A calls *Free* oracle query to free the chosen tag. Finally an adversary sends previously recorded A, B to any one of the challenge tags by quering *SendTag* oracle and gets C_x. As we know last session is incomplete and tag is unable to update its secret parameters, thus C_x is computed by the secret parameters K_1, K_2, K_3. If $C_x = C$, then $x =$ otherwise $x = |1 - b|$, i.e. challenge tag $vtag_x$ is having identity ID_b or ID_{1-b} respectively. Thus an adversary is able to trace the tag in R^2AP protocol.

$CreateTag(ID_0)$ *and* $Create\ Tag(ID_1)$
Choose $b \in \{0,1\}$
$vtag_b \leftarrow DrawTag(ID_b)$
$\pi \leftarrow Launch$
$IDS \leftarrow SendTag(Init, vtag_b)$
$A,B \leftarrow SendReader(IDS,\pi)$
$C \leftarrow SendTag(A,B, vtag_b)$
$D,E \leftarrow SendReader(C,\pi)$
An adversary chooses two random no's r_{A_1}, r_{A_2}.
$Null \leftarrow SendTag(r_{A_1}, r_{A_2}, vtag_b)$
$Free(vtag_b)$
The session is incomplete.
$vtag_x \leftarrow DrawTag(ID_x)$
$C_x \leftarrow SendTag(A,B,vtag_x)$
Oracle query comes to an end.
If $C_x = C$ then $x = b$ otherwise $x = |1-b|$

Theorem 4.3: R^2AP protocol is vulnerable to impersonation attack.

Proof: In R^2AP protocol, an adversary can successfully impersonate the legitimate user of the server just by means of active and passive attacks. An adversary A get access to two uncorrupted tags $vtag_0$ and $vtag_1$ as its challenge tags and then randomly chooses $vtag_b$, $b \in \{0, 1\}$ among them. A analyzes the protocol run between R and $vtag_b$ and evaluates all oracles on $vtag_b$. An adversary chooses two random numbers r_{A1} and r_{A2} and sends them to tag to stop the tag for key updation. A calls *Free* oracle query to free the chosen tag. Eventually an adversary generates fake tag $vtag_x$ and sends previously recorded C to reader by quering *SendTag* oracle. As we know last session is incomplete and C is computed by the secret parameters K_1, K_2, K_3. Now reader extracts $ID_x = C_x \oplus Rec(Rec(K_2, K_3), Rec(r_1, K_1))$ from the received request message C_x and match it with the ID, that is stored in the database. If $ID_x = ID$, then an adversary is successful to impersonate the reader. Thus R^2AP protocol is vulnerable to impersonation attack as even any fake tag is authenticated by the reader.

$CreateTag(ID_0)$ *and* $Create\ Tag(ID_1)$
Choose $b \in \{0,1\}$
$vtag_b \leftarrow DrawTag(ID_b)$
$\pi \leftarrow Launch$
$IDS \leftarrow SendTag(Init, vtag_b)$
$A,B \leftarrow SendReader(IDS,\pi)$
$C \leftarrow SendTag(A,B, vtag_b)$
$D,E \leftarrow SendReader(C,\pi)$
An adversary chooses two random no's r_{A_1}, r_{A_2}.
$Null \leftarrow SendTag(r_{A_1}, r_{A_2}, vtag_b)$
$Free(vtag_b)$
The session is incomplete.
An adversary generate fake tag $Create(ID_x)$
$vtag_x \leftarrow DrawTag(ID_x)$
$IDS \leftarrow SendTag(Init, vtag_x)$
$A,B \leftarrow SendReader(IDS,\pi)$
$C_x \leftarrow SendTag(A,B, vtag_x)$
Oracle query comes to an end.
If requested C_x is accepted by the reader then an adversary wins.
Reader successfully authenticate fake tag $vtag_x$.

5. DEHKORDI AND FARZANEH SCHEME

In 2012, Cho et al. proposed a hash-based radio-frequency identification tag mutual authentication protocol [17] on the basis of brute-force attack cost and retrieval cost. However in 2014, Dehkordi and Farzaneh [18] demonstrated traffic analysis and tag/reader impersonation attacks on [17] and proposed the countermeasures to thwart the security threats and to minimize the computation cost.

5.1 Protocol Description

As described in Table 3, in Dehkordi and Farzaneh hash based RFID mutual authentication protocol [18], server assigns tag identity $ID \in \{0, 1\}^l$ and two secret keys $K_1, K_2 \in \{0, 1\}^l$ for all tags and stores the triplet (ID, K_1, K_2) in tag's memory as well as in the database system. Database system also stores the previous session secret keys K_1^O, K_2^O in its database corresponding to each tag's identity to keep the synchronized case. Thus server stores the parameters $(ID, K_1, K_2, K_1^O, K_2^O)$ for all tags. Back-end server, reader and tag follow the following steps to mutually authenticate each other and to update new dynamic secret parameters:

1. Foremost reader generates a pseudo random number $r_1 \in \{0, 1\}^l$ and sends it to the tag.
2. Upon receiving the random number r_1, tag itself generates a pseudo random number $r_2 \in \{0, 1\}^l$ and sends the response message $\{M_1, M_2\}$ to the reader, where

   $M_1 = r_2 \oplus K_1$ and $M_2 = h(ID \oplus r_2 \| r_1 \oplus K_1)$
3. Now reader forward the same response message along with its random number, i.e. the message $\{M_1, M_2, r_1\}$ to the backend server in a secure communication channel.
4. Then server firstly extracts the triplet (ID, K_1, K_2) corresponding to each tag in its database and finds $r_2^* = M_1 \oplus K_1$ for each tag. Server confirms the authenticity of r_2^* by verifying $M_2^* = h(ID \oplus r_2^* \| r_1 \oplus K_1)$ with the received M_2. If it is not verified for any of the triplet (ID, K_1, K_2), then server tries to verify it with the previous session keys (ID, K_1^O, K_2^0). If still it is not verified, then server dismiss the session otherwise authenticates the requested tag.
5. Furthermore, after successful authentication, server generates a pseudo random number $r_3 \in \{0, 1\}^l$ and transmits mutual authentication message $\{Data, M_3, M_4\}$ to the reader, where

   $M_3 = r_3 \oplus K_2$ and $M_4 = h(r_2 \oplus K_2 \| r_3)$
6. Later on from the authentication message $\{DATA, M_3, M_4\}$, reader extracts $DATA$, i.e. information regarding the requested tag and sends the remaining message $\{M_3, M_4\}$ to the tag for further communication.
7. Eventually tag extracts $r_3 = M_3 \oplus K_2$ and authenticates the server by confirming that computed $M_4^* = h(r_2 \oplus K_2 \| r_3^*)$ is identical to the received M_4. If it is so, then mutual authentication can be done.
8. After achieving the mutual authentication, server and the tag computes their newly updated secret parameters K_1^N and K_2^N ,where

   $K_1^N = r_3 \oplus K_1 \oplus (r_2 \& K_2)$

   $K_2^N = r_2 \oplus K_2 \oplus (r_3 \& K_1)$.

   Tag replace the stored triplet (ID, K_1, K_2) with the new triplet (ID, K_1^N, K_2^N). To save the protocol from desynchronization attack server will not replace the new triplet (ID, K_1^N, K_2^N) with the existing one (ID, K_1, K_2) at that time and maintain the pair (ID, K_1, K_2) till synchronized authentication can be done.

Table 2: Dehkordi and Farzaneh Protocol

Server $(ID, K_1, K_2, K_1^O, K_2^O)$	**Reader**	**Tag** (ID, K_1, K_2)
	Generate r_1	
	$\xrightarrow{r_1}$	
		Generate r_2
		Compute $M_1 = r_2 \oplus K_1$
		$M_2 = h(ID \oplus r_2 \| r_1 \oplus K_1)$
		$\xleftarrow{M_1, M_2}$
	$\xleftarrow{M_1, M_2, r_1}$	
Compute $r_2^* = M_1 \oplus K_1$		
$M_2^* = h(ID \oplus r_2^* \| r_1 \oplus K_1)$		
Verify $M_{2=}^{*?} M_2$		
Generate r_3		

Compute $M_3 = r_3 \oplus K_2$		
$M_4 = h(r_2 \oplus K_2 \| r_3)$		
$\xrightarrow{DATA, M_3, M_4}$		
	Extract $DATA$	
	$\xrightarrow{M_3, M_4}$	
		Compute $r_3 = M_3 \oplus K_2$
		$M_4^* = h(r_2 \oplus K_2 \| r_3^*)$
		Verify $M_{4=}^{*?} M_4$
Update secret parameters		
$K_1^N = r_3 \oplus K_1 \oplus (r_2 \& K_2)$ and $K_2^N = r_2 \oplus K_2 \oplus (r_3 \& K_1)$		

5.2 Cryptanalysis

To overcome the security threats of Cho et al. protocol [24], Dehkordi and Farzaneh presented an improved hash based RFID mutual authentication protocol [25]. But still on the basis of Vaudenay untraceability model [26] and non tamper resistance property of RFID tags, we demonstrate by formal security analysis that their RFID authentication protocol is insecure for real life applications as it does not attain even *Narrow Forward* privacy level of security. Thus, author's claims are proven to be wrong.

Theorem 5.1: Dehkordi and Farzaneh hash based RFID mutual authentication protocol does not provide even *Narrow Forward* privacy.

Proof: Dekhordi and Farzaneh hash based RFID mutual authentication protocol only achieves at max Weak privacy level. An adversary *A* get access to two un-corrupted tags $vtag_0$ and $vtag_1$ as its challenge tags and then randomly chooses $vtag_b$, $b \in \{0, 1\}$ among them. *A* analyzes the protocol run between *R* and $vtag_b$ and evaluates all oracles on $vtag_b$. *A* calls *Free* oracle query to free the chosen tag. Finally an adversary calls *Corrupt*($vtag_x$) on any one of the challenge tags to get K_{1x} , K_{2x} . Now *A* is able to compute $r_{2x} = M_1 \oplus K_{1x}$, $M_{2x} = h(ID \oplus r_{2x} \| r_1 \oplus K_{1x})$ by means of corrupted keys. If $M_{2x} = M_2$, then $x = b$ otherwise $x = |1 - b|$, i.e. corrupted tag by an adversary is having identity ID_b or ID_{1-b} respectively. Thus an adversary is able to trace the tag.

CreateTag(ID_0) and Create Tag(ID_1)
Choose $b \in \{0,1\}$
$vtag_b \leftarrow$ *DrawTag(ID_b)*
$\pi \leftarrow$ *Launch*
$r_1 \leftarrow$ *SendReader(Init, π)*
M_1 , $M_2 \leftarrow$ *SendTag(r_1, $vtag_b$)*
Free($vtag_b$)
The session is incomplete.
$vtag_x \leftarrow$ *DrawTag(ID_x)*
K_{1_x} , $K_{2_x} \leftarrow$ *Corrupt ($vtag_x$)*
Oracle query comes to an end.
$r_{2_x} = M_1 \oplus K_{1_x}$
$M_{2_x} = h(ID \oplus r_{2_x} \| r_1 \oplus K_{1_x})$
If $M_{2_x} = M_2$ then $x = b$ otherwise $x = |1-b|$

6. CONCLUSION

In this paper, the author cryptanalyzed recently proposed Zhuang et al.'s ultralightweight RFID authentication protocol for low cost tags [15] and pointed out that their protocol can't provide even *Narrow Forward* privacy level of security. In addition *R²AP* protocol is traceable and suffers from impersonation attack. After that, the author examined Dehkordi and Farzaneh's improved hash based RFID mutual authentication protocol [18] on the basis of Vaudenay's untraceability model and non tamper resistance property of RFID tag and demonstrated that their authentication protocol is insecure for real life applications as it does not attain even *Narrow Forward* privacy level of security. In RFID authentication system, messages are communicated through radio frequency waves, which are highly insecure, as an adversary can intercept and do modification in the messages, so in future RFID authentication protocol will be designed in such a manner, that even strong class adversary which has access to all oracles, can't even edit, delete or modify any communication.

7. REFERENCES

[1] S. D. Kaul and A. K. Awasthi, "Rfid authentication protocol to enhance patient medication safety," Journal of medical systems, vol. 37, no. 6, pp. 1–6, 2013.

[2] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "LMAP : A real lightweight authentication protocol for low cost rfid tags," In Hand of Workshop on RFID and Lightweight Crypto, 2006.

[3] Peris Lopez, Pedro, et al., "M²AP : A minimalist mutual authentication protocol for low cost rfid tags," In Proc. of UIC'06, Springer Verlag, vol. 4159, pp. 912–923, 2006.

[4] Peris Lopez, Pedro, et al.,, "EMAP : An efficient mutual authentication protocol for low cost rfid tags," In Proc. of IS'06, Springer Verlag, vol. 4277, pp. 352– 361, 2006.

[5] T. Li and G. Wang, "Security analysis of two ultra-lightweight rfid authentication protocols," in New Approaches for Security, Privacy and Trust in Complex Environments. Springer, 2007, pp. 109–120.

[6] H. Y. Chien, "SAS1: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity," IEEE Transactions on Dependable and Secure Computing, vol. 4, no. 4, pp. 337–340, Oct-Dec 2007.

[7] R.-W. Phan, "Cryptanalysis of a new ultralightweight rfid authentication protocolsasi," Dependable and Secure Computing, IEEE Transactions on, vol. 6, no. 4, pp. 316–320, 2009.

[8] H.-M. Sun, W.-C. Ting, and K.-H. Wang, "On the security of chien's ultralightweight rfid authentication protocol," IEEE Transactions on Dependable and Secure Computing, no. 2, pp. 315–317, 2009.

[9] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "Advances in ultralightweight cryptography for low cost rfid tags : Gossamer protocol," In Proc. of WISA'08, Springer Verlag, vol. 5379, pp. 56–68, 2008.

[10] Z. Bilal, A. Masood, and F. Kausar, "Security analysis of ultra-lightweight cryptographic protocol for low-cost rfid tags: Gossamer protocol," in Network-Based Information Systems, 2009. NBIS'09. In-ternational Conference on. IEEE, 2009, pp. 260–267.

[11] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan, "Lightweight mutual authen-tication and ownership transfer for rfid systems," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–5.

[12] Y.Yang, J. Gu, C. Lv, Q. Jiang, W. Ma, "Security analysis of Kulseng et al.'s mutual authentication protocol for RFID systems" Information Security, IET, vol. 6, no. 4, pp. 239–248, 2012.

[13] Y. Tian, G. Chen, and J. Li, "A new ultralightweight rfid authentication protocol with permutation," IEEE Communications Letters, vol. 16, no. 5, pp. 702–705, May 2012.

[14] G. Avoine and X. Carpent, "Yet another ultralightweight authentication protocol that is broken," in Radio Frequency Identification. Security and Privacy Issues. Springer, 2013, pp. 20–30.

[15] X. Zhuang, Y. Zhu, and C.-C. Chang, "A new ultralightweight rfid protocol for low-cost tags: R^2AP ," Wireless Personal Communications, vol. 79, no. 3, pp. 1787–1802, 2014.

[16] A. Juels and S. A. Weis, "Defining strong privacy for rfid," ACM Transactions on Information and System Security (TISSEC), vol. 13, no. 1, p. 7, 2009.

[17] J.-S. Cho, Y.-S. Jeong, and S. O. Park, "Consideration on the brute-force attack cost and retrieval cost: A hash-based radio-frequency identification (rfid) tag mutual authentication protocol," Computers & Mathematics with Applications, 2012.

[18] M. H. Dehkordi and Y. Farzaneh, "Improvement of the hash-based rfid mutual authentication protocol," Wireless personal communications, vol. 75, no. 1, pp. 219–232, 2014.

[19] S. Vaudenay, "On privacy models for rfid," in Advances in Cryptology– ASIACRYPT 2007. Springer, 2007, pp. 68–87.