Wireless Pers Commun (2017) 95:2803–2828
DOI 10.1007/s11277-017-3965-1

Privacy Model for Threshold RFID System Based on PUF

Sonam Devgan Kaul[1] · Amit K. Awasthi[1]

Published online: 31 January 2017

Abstract In order to enhance shared control of the secret among multiple RFID tags and to ensure secure communication through an insecure channel, we present in this paper a new idea of threshold RFID system. We extend well-known Vaudenay's RFID privacy model (Vaudenay in Adv Cryptolo 68–87, 2007) to make the RFID system acceptable for threshold secret sharing system among n tags. To show its implementation and to resist tag compromising attack, we design an efficient threshold RFID authentication protocol based on physical unclonable functions. It is a method of distributing a secret s among a set of n RFID tags in such a way that any set of $t(t<n)$ or more tags will recover the shared secret s only after successful mutual authentication while the secret will remain uncertain if any of them will be unauthorized tag or a group of $t-1$ or less tags have given their information. In order to enhance tag anonymity, we use dynamic security parameters which are updated after each successful run of mutual authentication session. Furthermore, via analyzing security and privacy formally and informally, we demonstrate that our scheme achieves destructive privacy and withstand against various known attacks.

Keywords Mutual authentication · Threshold secret sharing · RFID · Physically unclonable function · One way hash function

1 Introduction

RFID is an automatic identification technology that uses radio waves to identify and authenticate objects over an insecure communication channel, the channel in which an adversary can intercept and modify the transactions in such a way that legitimate recipient of the transaction does not detect the manipulation [2]. In today's web enabled world,

✉ Sonam Devgan Kaul
sonamdevgan11@gmail.com

[1] School of Applied Sciences, Gautam Buddha University, Greater Noida 201306, India

RFID technology can play a vital role in multidimensional domains like transportation, access control, logistics, manufacturing, inventory control, asset management, e-health, etc. RFID system consists of a secure back-end server, a few readers and a set of low cost tags.

Physically Unclonable Function(PUF) is an emerging security technology that maps a set of challenges to a set of responses $P(z, .) : Challenge \rightarrow Response$ and based upon tag's untraceable complex physical characteristics, say z like supply voltage, temperature, electromagnetic interference, etc. To resist tag compromising or cloning attacks, we employ PUF function. PUF function can be easily implemented in tags as implementation of a PUF circuit in such a small area requires less than 1000 gates [3]. PUF behaves like one way function; it is infeasible to find the challenge corresponding to the given output. As PUF function itself produces somewhat different outputs for the same challenges due to environment noise, so fuzzy extractors are used with the PUF to produce same output for the same challenge [4]. Execution of any physical attack on the device for the purpose of exploring the structure of PUF function will cause the destruction of its respective physical characteristics.

In RFID system we use the concept of threshold secret sharing to enhance the shared control of the secret among multiple tags. The (t, n) threshold scheme [5] is a method of distributing the secret among a group of n participants in such a way that any group of atleast t $(t < n)$ participants can recover the secret by pooling their shares but the secret remains uncertain even with the knowledge of atmost $t - 1$ participant shares. Also we extend well-known Vaudenay's RFID privacy model [1] to make the RFID system acceptable for threshold secret sharing system among n tags and to show its implementation, we design an efficient threshold RFID authentication protocol based on physical unclonable functions and formally analyze its security and privacy.

Our proposed RFID threshold secret sharing authentication scheme is the mechanism of distributing the secret s among n tags by generating n shares and then initializing n tags with these shares in such a way that any set of t or more tags will enter into the system and then only after successful mutual authentication server will recover the shared secret key s using Lagrange interpolation but the secret will remain uncertain if any of them will be unauthorized tag or a group of $t - 1$ or less tags have given their information. Due to low storage capacity and limited computation and communication cost of tags, in our threshold RFID system we used only low cost cryptographic primitives such as bitwise Xor operation, pseudo random number generator function, one way hash function and physically unclonable function. In order to enhance anonymity and untraceability of tags, we use dynamic security parameters which are updated after each successful run of mutual authentication session.

1.1 Motivation

In a real world scenario, like to open bank vault, to authenticate an electronic fund transactions, to do shared asset management, to control shared public transport in desolated areas, etc. shared control of the secret among multiple RFID tags will be nowadays an emerging technology which motivate us to use threshold secret sharing in RFID system as the consequence of any adversary interpretation will be expensive as well as unsafe for the society. Also RFID tags are susceptible to traceability, forward traceability, backward traceability, cloning, de-synchronization, impersonation, replay, denial of service, man in middle and side channel attacks. Thus by considering all the requirements and problems in our mind, to the best of our knowledge, we present in this paper a new idea of threshold

RFID system and develop secure and efficient threshold RFID mutual authentication protocol using PUF function.

1.2 Organization

The rest of the paper is organized as follows: Sect. 2 briefly review the related work. Definitions are described in Sect. 3. Extended threshold RFID framework is presented in Sect. 4. PUF based threshold RFID authentication protocol is proposed in Sect. 5. Formal and informal security proofs are given in Sect. 6 followed by the performance evaluation in Sect. 7. Eventually, we conclude the paper in Sect. 8.

2 Related Work

To design privacy preserving, secure and efficient RFID authentication protocol, large number of RFID frameworks have been proposed in recent years. In 2005, Avoine [6] introduced the first RFID privacy model based on untraceability notion, but his model is just capable to consider 3-pass RFID protocols. In 2006, Lim and Kwon [7] broaden Avoine's privacy model by formally introducing forward and backward untraceability. Afterwards, Juels and Weis [8] proposed RFID privacy model depending upon indistinguishability of tags. In 2007, Vaudenay [1] proposed simulation based comprehensive RFID security and privacy model in which adversary's capabilities are classified into $\{WIDE, NARROW\} \times \{STRONG, DESTRUCTIVE, \ FORWARD, \ WEAK\}$ classes. Also in his model, an adversary has full capability to create unregistered fake tags. After his work, large number of models have been proposed in recent years to extend it.

In 2008, Paise and Vaudenay [9] enriched his model [1] to analyze mutual authentication protocols. Later in 2010, a new framework for RFID privacy based on zero knowledge formulation was proposed by Deng et al. [10] with the aim to analyze those protocols in which after each protocol execution, entities secret information may be updated. Afterwards in 2011, Hermans et al. [11] propose a new indistinguishability privacy model but his model has several drawbacks [12]. In 2013, Coisel and Martin [12] give a platform by examining the existing well known RFID models preserving privacy [1, 6–11] and analyze their advantages and drawbacks.

Till now Vaudenay model [1] comes out to be one of the most comprehensive and powerful privacy model approximately. So we extend his framework to make the RFID system acceptable for threshold secret sharing system. The concept of threshold secret sharing scheme independently introduced by Shamir [5] and Blakley [13] in 1979. Shamir's work is based on polynomial interpolation while Blakley give geometric approach solution of safeguarding cryptographic keys. Since then in order to reduce the storage capacity and to detect fake shares many schemes [14–16] have been proposed in literature.

Highest level of feasible privacy; *STRONG* privacy in Vaudenay model is achieved only by public key cryptography and symmetric cryptography based authentication protocols at the max attain *DESTRUCTIVE* or *FORWARD* privacy [1]. But it is infeasible to implement such high cost cryptographic primitives on low cost RFID tags. A low cost cryptographic primitive PUF have been widely studied in recent years [3, 17–19] to achieve highest privacy level. Security is enhanced in these PUF based authentication protocols [3, 17–19] because of tamper resistance properties of PUF device; execution of any physical attack on the device is of no use. So to enhance security and privacy and to resist tag compromising or cloning attacks in RFID tags, we employ PUF functions in our threshold RFID system.

3 Definitions

In this section, we define negligible function, collision resistant one way hash function and physically unclonable function.

Definition 1 Negligible Function. A function $\epsilon(\lambda) : \mathbb{N} \rightarrow \mathbb{R}$ depends upon the security parameter λ is said to be negligible function in λ if for every $l > 0$ there exist a number $m \in \mathbb{N}$ such that

$$\epsilon(\lambda) <, \frac{1}{\lambda^l} \qquad \forall \quad \lambda > m$$

$$(M.Bellare, Anoteonnegligiblefunctions\,[20])$$

Definition 2 One Way Hash Function. One way hash function

$$h(.) : \{0,1\}^* \rightarrow \{0,1\}^l$$

maps an arbitrary length message to a fixed length message $l \in \mathbb{N}$, as defined in [21] such that

1. $h(.)$ is pre-image resistant as for any $y \in \{0,1\}^l$, it is infeasible to find $x \in \{0,1\}^*$ such that $h(x) = y$.
2. $h(.)$ is second pre-image resistant as for any $x \in \{0,1\}^*$, it is infeasible to find $x' \in \{0,1\}^*$, $x' \neq x$ such that $h(x) = h(x')$.
3. $h(.)$ is collision resistant. Let the advantage of an adversary A in finding collision for one way hash function $h(.)$ be $ADV_A^{HASH}(t)$, i.e. advantage of an adversary A is the probability to randomly select a pair (x, x') such that $h(x) = h(x')$ and $x \neq x'$. Then one way hash function is secure and collision resistant if for any sufficiently small $\epsilon > 0$,

   $$ADV_A^{HASH}(t) \;=\; Pr\,[(x,x') : x \neq x',\, h(x) = h(x')] \;< \epsilon$$

   where the probability in the advantage is evaluated over the random choices made by the adversary A within the execution time t

3.1 Physically Unclonable Function

Physically unclonable function(PUF) is an emerging security technology that maps a set of challenges to a set of responses in unique unpredictable way

$$P(z,.) : Challenge \rightarrow Response$$

and based upon untraceable complex physical characteristics, say z of each device like RFID tags, smart cards, etc. [18].

Definition 3 Physically Unclonable Function (PUF). For security parameter λ and $m(\lambda)$, $l(\lambda) \in \mathbb{N}$; an ideal physically unclonable function

$$P(z,.) : \{0,1\}^m \rightarrow \{0,1\}^l$$

based on untraceable complex physical characteristics z of each device such that

1. For any particular device, PUF function gives same response for the same challenge as for any $x, x' \in \{0,1\}^m$ and for any physical characteristics z; if $P(z,x) = y$, $P(z,x') = y'$ and $x = x'$ then $Pr[y = y'] = 1$. While for the different devices PUF function gives different responses for the same challenge as for any $x, x' \in \{0,1\}^m$ and $P(z,x) = y$ for the device having physical characteristics z and $P(z',x') = y'$ for the other device having physical characteristics z', if $x = x'$ then $Pr[y = y'] < \epsilon$.
2. PUF function is unpredictable as it behaves like random functions. For any probabilistic polynomial time probability of an adversary A to distinguish response of PUF function and random number is at most negligible as for any $y \in \{0,1\}^l$ it is infeasible to find $x \in \{0,1\}^m$ such that $P(z,x) = y$.
3. Execution of any physical attack on the device for the purpose of exploring the structure of PUF function will cause the destruction of its respective physical characteristics and PUF function can not be evaluated correctly for that particular device. In any probabilistic polynomial time, advantage of an adversary A to execute physical attack on the device is at most negligible, $ADV_A(t) < \epsilon$.

4 Threshold RFID Framework

First of all the notations used throughout the paper are summarized in Table 1. Then in this section, we present the RFID system set up procedures, adversary oracle model and security and privacy experiment for threshold RFID system by doing modification in Vaudenay privacy model [1] to meet our requirements. We extend his model to make the RFID system acceptable for threshold secret sharing system among n tags. RFID system

Table 1 Notations

x	Server secret key
ID_i	Identity of ith tag
K_{i_1}, K_{i_2}	Secret keys of ith tag
s	Secret shared key
s_i	Shared secret key component for ith tag
n	Number of tags
t	Threshold value less than n
z_i	Untraceable complex physical characteristics of ith tag
r_{i_k}	Random strings of l bits generated by pseudo random generator for $1 \le k \le 4$
T	Current date and time of input device
δT	Expected time interval for a transmission delay
$h(.)$	Secure one way hash function $h(.) : \{0,1\}^* \rightarrow \{0,1\}^l$
$P(z, .)$	Physically unclonable function $P(z,.) : \{0,1\}^m \rightarrow \{0,1\}^l$
$\oplus$	Bitwise XOR operation
$\|\|$	Concatenation operation

consists of a secure backend server S, a few readers R and a set of tags T. RFID tags are assumed to be efficient enough to use low cost basic cryptographic primitives such as pseudo random number generator, one way hash function and physically unclonable function. Communication channel between R and S are assumed to be secure while R and T are connected through an insecure communication channel. RFID threshold system performs the following procedure:

4.1 System Model

In RFID based threshold authentication protocol, each tag is being initialized by their shared secret key component s_i, This can be done by the following procedures:

1. SETUPSERVER$(1^{\lambda}) \rightarrow (pk_s, x, s, t, f(.), DB_s)$: Generate public pk_s, private key x of the server and shared secret key s depending upon the security parameter λ. It also generate $t-1$ degree polynomial $f(.)$ corresponding to secret parameters x and s. To setup tags, algorithm generates the partial secret key component s_i corresponding to ith tag depends upon the identity parameter ID_i and polynomial function $f(.)$ for all $1 \leq i \leq n$. To store secret information about tags server creates an empty database DB_s.
2. SETUPREADER$(1^{\lambda}, pk_s) \rightarrow (pk_r, sk_r)$: Generate public/private key pair of the reader (pk_r, sk_r) depending upon the security parameter λ. An execution of threshold RFID protocol π is initialized by reader R via sending random number. R has a secure communication channel with the server while R and T communicate through an insecure channel. Also tags are operated only when they are in the readers field of communication.
3. SETUPTAG$(ID_i, z_i, r_1, r_2, pk_s) \rightarrow (K_{i_1}, K_{i_2}, InitState_{T_i})$: Tag having identity ID_i generates its secret keys K_{i_1} and K_{i_2} via physically unclonable function and one way hash function corresponding to its physical characteristics z_i and random numbers r_1 and r_2. Also creates tags initial state $InitState_{T_i}$ with ID_i. Contrary to the Vaudenay privacy model, instead of storing secret key directly in its non-volatile memory, tag save it in its physical characteristics. Also to set up i^{th} tag, server generates partial secret key component s_i and save it indirectly in tag memory.
4. IDENTPROTOCOL(π): Execute a polynomial time interactive protocol π between S, R and T. If the tag is legitimate then server accepts it and produces an output ID_i otherwise output is $\perp$. Only after successful authentication of atleast t tags out of n, secret key can be recovered. Also tag as well as server update their memory after successful protocol session π.

4.2 Adversary Model

An adversary A 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 as defined in [1]:

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 free tags between all the existing ones with given probability distribution $distr$. New pseudonym $vtag_i$ 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 now A is not able to call *vtag* in its oracles.
4. *Launch*(π): An adversary authorized R to initiate a new session of the protocol π between R and T.
5. *SendReader*$(m, \pi) \rightarrow m'$: An adversary A may send a message m of his choice to the reader in the protocol execution π which output m'.
6. *SendTag*$(m, vtag) \rightarrow m'$: An adversary A may send any message m to the drawn tag *vtag* which responds with m'.
7. *Result*(π): This oracle outputs 1 if session of the protocol π is successfully executed and shared secret key can be recovered otherwise it outputs 0.
8. *Corrupt*$(vtag)$: This oracle outputs the volatile as well as the non volatile memory of the drawn tag *vtag*.

4.3 Adversary Classes

In Vaudenay privacy model [1], *STRONG* class adversary has full access to all the above oracles without any restriction. *DESTRUCTIVE* class adversary has no ability to use any other oracle on *vtag* after querying *Corrupt*$(vtag)$ oracle. *FORWARD* class adversary can just use *Corrupt*$(vtag)$ oracle only once. *WEAK* class adversary is not allowed to use *Corrupt*$(vtag)$ oracle. *NARROW* class adversary has no access to *Result* oracle query while *WIDE* adversary can access *Result* oracle. Thus obviously we have:

$$WEAK \subseteq FORWARD \subseteq DESTRUCTIVE \subseteq STRONG$$

4.4 Security Privacy Notions

In this section, we discuss security notions in which non legitimate tags and non legitimate readers are rejected by the server as well as the privacy notions which presents the untraceablity of tags. Vaudenay security model [1] give emphasis on all the attacks in which an adversary has capability to forge a legitimate tag except the cloning attack. For this purpose we have presented PUF functions in tags so that cloning of tags will become infeasible. Also contrary to the Vaudenay model, compromisation of tags as well as readers both can be done by the malicious adversary. Corruption of any reader will provide an adversary secure and discontinuous communication with the server.

Definition 4 Tag Authentication. A RFID system attains tag authentication if the success probability of strong adversary A for identifying a non legitimate tag is at most negligible.

Definition 5 Reader Authentication. A RFID system attains reader authentication if the success probability of strong adversary A for identifying a non legitimate reader is at most negligible.

Privacy is explained by means of the the blinder B and trivial adversary, as defined in [1]. B simulates *Launch*, *SendReader*, *SendTag* and *Result* oracles without having any 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.

Definition 6 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

$$|Pr(A\,succeeds) - Pr(A^B\,succeeds)| < \epsilon(\lambda)$$

4.4.1 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, as defined in [1]:

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*, *SendTag* and *Result* oracles.
2. *Challenge Phase* : An adversary A obtains the hidden table, which maps *vtag* to identity of the tag. An adversary A get access to two uncorrupted challenge tags and then randomly select any one from them. A evaluates oracles on that particular tag according to her class.
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.

Definition 7 Privacy. An RFID system is said to be P-private if $\forall\ A \in P$,

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

Definition 8 Forward Untraceable. Let the tag T be corrupted in session i and reveals the corresponding secret keys. An RFID system is said to be forward untraceable if $\forall\ A \in P$, probability of A to trace the tag T in session i' $(i' > i)$ is at most negligible.

Definition 9 Backward Untraceable. Let the tag T be corrupted in session i and reveals the corresponding secret keys. An RFID system is said to be backward untraceable if $\forall\ A \in P$, probability of A to trace the tag T in session i' $(i' < i)$ is at most negligible.

5 Proposed PUF based Threshold RFID Mutual Authentication Protocol

In this section, we implement the proposed threshold RFID system by designing an efficient (t, n) threshold RFID mutual authentication protocol based on physically unclonable function. It is a mechanism of distributing a secret s among a set of n RFID tags in such a way that any group of t or more tags will recover the secret s in the probabilistic polynomial time by using physically unclonable function $P(z, .) : \{0,1\}^m \rightarrow \{0,1\}^l$, one way hash function $h(.) : \{0,1\}^* \rightarrow \{0,1\}^l$, bitwise XOR operation and pseudo random number generator function, as depicted in Figs. 1 and 2. The scheme enable to generate n shares s_i for each tag by the shared secret s and then enable to initialize n tags with these partial key components s_i. Any set of t or more tags will enter into the system and then only after successful mutual authentication server will recover the shared secret key s using lagrange interpolation. We used in our authentication protocol Shamir's threshold secret sharing scheme [5] for distribution and recovery of keys but anyone can use any other secret

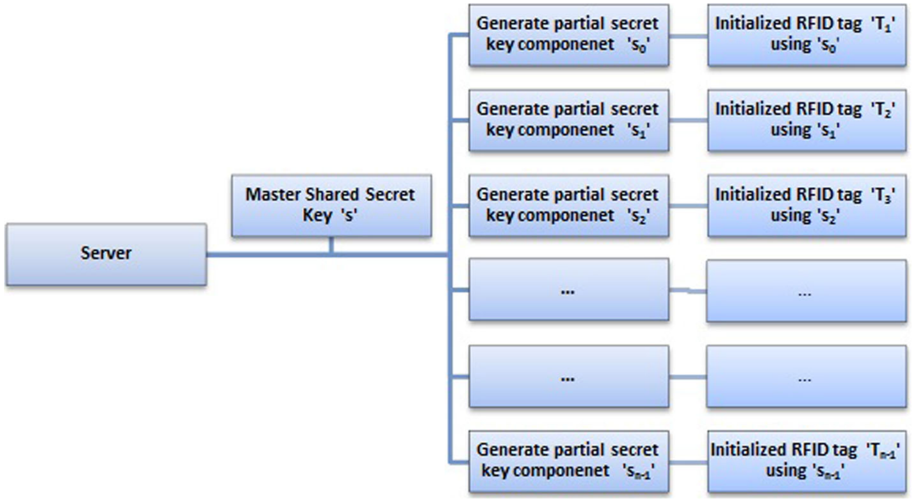
Server
Master Shared Secret Key 's'
Generate partial secret key componenet 's0'
Initialized RFID tag 'T1' using 's0'
Generate partial secret key componenet 's1'
Initialized RFID tag 'T2' using 's1'
Generate partial secret key componenet 's2'
Initialized RFID tag 'T3' using 's2'
...
...
...
...
Generate partial secret key componenet 'sn-1'
Initialized RFID tag 'Tn-1' using 'sn-1'

Fig. 1 Initialization process of threshold RFID system

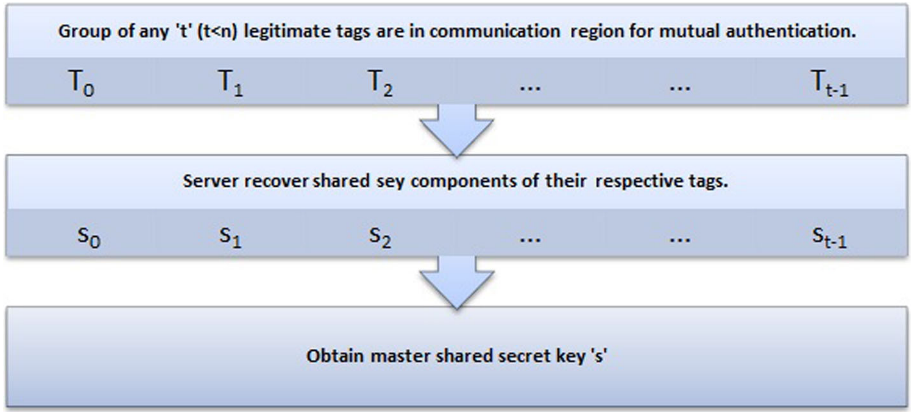
Group of any 't' (t<n) legitimate tags are in communication region for mutual authentication.
T0
T1
T2
...
...
Tt-1
Server recover shared sey components of their respective tags.
S0
S1
S2
...
...
St-1
Obtain master shared secret key 's'

Fig. 2 Authentication and key recovery process of threshold RFID system

sharing scheme in our RFID based authentication protocol to achieve their requirements. In order to enhance anonymity and untraceability of tags, we use dynamic security parameters which are updated after each successful run of mutual authentication protocol.

The proposed protocol consists of three phases: Initialization Phase, Authentication Phase, Updation and Key Recovery Phase.

5.1 Initialization Phase

In an initialization phase, as defined in Shamir's secret sharing scheme [5], server distribute the shared secret key s among n tags and initialized n tags as described follows with their partial shared secret key components:

1. Server foremost chooses a secret lagrange polynomial $f(y)$ of degree $t-1$ over $GF(p)$ for prime p

$$f(y) = a_0 + a_1 y + a_2 y^2 + \ldots + a_{t-1} y^{t-1}$$

such that the secret is

$$f(0) = a_0 = s \oplus x$$

where x is the server's secret key, s is the secret to be shared among n tags and $a_1, a_2, a_3, \ldots, a_{t-1}$ are random coefficients over $GF(p)$.

2. To initialize ith tag, firstly server finds its respective partial secret key component by computing:

$$s_i = f(ID_i) \qquad \forall \quad 0 \leq i \leq n-1$$

3. Then ith tag generates two random numbers $r_{i_1}, r_{i_2} \in \{0,1\}^l$ and computes its secret keys K_{i_1} and K_{i_2} along with the secret messages α_i and β_i by using physically unclonable function and one way hash function, where

$$K_{i_1} = P(z_i, r_{i_1} \oplus ID_i), \quad \alpha_i = h(K_{i_1} \oplus ID_i)$$
$$K_{i_2} = P(z_i, r_{i_1} \oplus r_{i_2}), \quad \beta_i = h(K_{i_2} \oplus ID_i)$$

and z_i are the physical characteristics of ith tag. w.l.o.g we assume that the identity message ID_i is padded with zero bits to make the bit size of ID_i as long as the output of hash function, i.e. l.

4. Finally to initiate ith tag with its partial secret key component, server computes:

$$\gamma_i = s_i \oplus h(\alpha_i) \oplus h(\beta_i)$$

5. Server stores the list of triplets $(ID_i, \alpha_i, \beta_i)$ in its database for n tags while i^{th} tag stores its random numbers along with γ_i and ID_i, i.e., the quadruple $(ID_i, r_{i_1}, r_{i_2}, \gamma_i)$ in its memory. As RFID tags are not tamper proof, any adversary who has the capability to corrupt the tag, can easily get the secret keys and the initial states stored on tags so we have not stored direct secret keys in the tags memory.

5.2 Authentication Phase

In an authentication phase, as described in Table 2, server, reader and tag follow the following steps to mutually authenticate any group of t legal tags, while any group of atmost $t-1$ tags cant recover any information about the secret in probabilistic polynomial time. Without loss of generality, we assume that t legitimate tags, for $0 \leq j \leq t-1$ will login into the system.

1. When t tags are in communication region, reader generates pseudo random numbers $r_{j_3} \in \{0,1\}^l$ for all $0 \leq j \leq t-1$ and send it to tags via an insecure communication channel.
2. Upon receiving random number r_{j_3}, jth tag itself generate a pseudo random number $r_{j_4} \in \{0,1\}^l$ and then finds its secret value K_{j_1} and α_j by using its specific PUF function which cant be cloned, where

$$K_{j_1} = P(z_j, r_{j_1} \oplus ID_j), \quad \alpha_j = h(K_{j_1} \oplus ID_j)$$

Table 2 Authentication phase

Server $(ID_j, \alpha_j, \beta_j)$	Reader	t-Tags $(ID_j, z_j, r_{j_1}, r_{j_2}, \gamma_j)$
	Generate r_{j_3} $\overset{[}{\underset{]}{\longrightarrow}} r_{j_3}$	
		Generate r_{j_4} $K_{j_1} = P(z_j, r_{j_1} \oplus ID_j)$ $\alpha_j = h(K_{j_1} \oplus ID_j)$ $M_{j_1} = ID_j \oplus h(\alpha_j) \oplus r_{j_4}$ $M_{j_2} = h(ID_j \oplus \alpha_j \oplus r_{j_3} \oplus r_{j_4})$ Delete K_{j_1} and α_j $K_{j_2} = P(z_j, r_{j_1} \oplus r_{j_2})$ $\beta_j = h(K_{j_2} \oplus ID_j)$ $M_{j_3} = h(\beta_j)$ $M_{j_4} = h(M_{j_2} \oplus M_{j_3} \oplus \beta_j \oplus r_{j_3} \oplus r_{j_4} \oplus T_j)$ Delete K_{j_2} and β_j $\overset{[}{\underset{]}{\longleftarrow}} (M_{j_1}, M_{j_4}, T_j)$
	Verify $(T_j^* - T_j) \leq \delta T_j$ $\overset{[}{\underset{]}{\longleftarrow}} (M_{j_1}, M_{j_4}, T_j, r_{j_3})$	
Find t triplets $(ID_j, \alpha_j, \beta_j) \in DB$ for which $r_{j_4}{}^* = M_{j_1} \oplus ID_j \oplus h(\alpha_j)$ $M_{j_2}{}^* = h(ID_j \oplus \alpha_j \oplus r_{j_3} \oplus r_{j_4}{}^*)$ $M_{j_4}{}^* = h(M_{j_2}{}^* \oplus h(\beta_j) \oplus \beta_j \oplus r_{j_3} \oplus r_{j_4}{}^* \oplus T_j)$ $M_{j_4}{}^* \overset{?}{=} M_{j_4}$ $M_{j_5} = h(M_{j_2}^* \Vert h(\beta_j) \Vert r_{j_3} \Vert r_{j_4}^*)$ $\overset{[}{\underset{]}{\longrightarrow}} M_{j_5}$		
	$\overset{[}{\underset{]}{\longrightarrow}} M_{j_5}$	
		$M_{j_5}{}^* = h(M_{j_2} \Vert M_{j_3} \Vert r_{j_3} \Vert r_{j_4})$ Verify $M_{j_5}{}^* \overset{?}{=} M_{j_5}$

Then tag computes an authentication factor M_{j_1} and M_{j_2} from K_{j_1} and α_j and immediately delete K_{j_1} and α_j from its volatile memory, where

$$M_{j_1} = ID_j \oplus h(\alpha_j) \oplus r_{j_4}, \quad M_{j_2} = h(ID_j \oplus \alpha_j \oplus r_{j_3} \oplus r_{j_4})$$

3. Subsequently jth tag finds its another secret parameter K_{j_2} and β_j and computes authentication messages M_{j_3} and M_{j_4} from them and thenceforth K_{j_2} and β_j both are deleted from its volatile memory, where

$$K_{j_2} = P(z_j, r_{j_1} \oplus r_{j_2}), \quad \beta_j = h(K_{j_2} \oplus ID_j)$$
$$M_{j_3} = h(\beta_j), \quad M_{j_4} = h(M_{j_2} \oplus M_{j_3} \oplus \beta_j \oplus r_{j_3} \oplus r_{j_4} \oplus T_j)$$

where T_j is the current time stamp and consequently jth tag sends the request message (M_{j_1}, M_{j_4}, T_j) to the reader.

4. After receiving the request message (M_{j_1}, M_{j_4}, T_j), reader first checks the validity of time stamp T_j, for all $0 \leq j \leq t-1$, by verifying $(T_j' - T_j) \leq \delta T_j$ to accept or reject the authentication request. If it finds incorrect, the authentication request is rejected else the reader sends $(M_{j_1}, M_{j_4}, T_j, r_{j_3})$ to the server along with the counter to avoid computation exhaustive attacks as counter value is increased with each reply of the tag and after the predefined wrong attempts server immediately lock the tag for some specific period.
5. Eventually server finds t triplets $(ID_j, \alpha_j, \beta_j)$, $0 \leq j \leq t-1$ which belongs to the database and satisfy the authentication factor M_{j_4}

$$M_{j_4}{}^* = h\left(M_{j_2}{}^* \oplus h(\beta_j) \oplus \beta_j \oplus r_{j_3} \oplus r_{j_4}{}^* \oplus T_j\right)$$

where

$$r_{j_4}{}^* = M_{j_1} \oplus ID_j \oplus h(\alpha_j), \quad M_{j_2}{}^* = h\left(ID_j \oplus \alpha_j \oplus r_{j_3} \oplus r_{j_4}{}^*\right)$$

If it is not verified for any $0 \leq j \leq t-1$, then the session is terminated instantly.

6. After verifying the legality of jth tag, server computes mutual authentication factor M_{j_5} and sends it to the reader, where

$$M_{j_5} = h\left(M_{j_2}^* \| h(\beta_j) \| r_{j_3} \| r_{j_4}^*\right)$$

7. Reader directly sends M_{j_5} to the tag.
8. Finally, tag verify the authenticity of received M_{j_5} by the computed $M_{j_5}{}^*$, where

$$M_{j_5}{}^* = h(M_{j_2} \| M_{j_3} \| r_{j_3} \| r_{j_4})$$

This equivalency authenticates the legitimacy of the reader. Thus mutual authentication can be done.

5.3 Updation and Key Recovery Phase

In order to enhance anonymity and untraceability of tags, server and tag compute new dynamic security parameters only after successful mutual authentication session. As described in Table 3 server, reader and tag perform the following steps to update and recover the secret from any group of t legitimate tags:

Table 3 Updation and key recovery phase

Server $(ID_j, \alpha_j, \beta_j)$	Reader	t-Tags $(ID_j, z_j, r_{j_1}, r_{j_2}, \gamma_j)$
		$K_{j_1}^N = P(z_j, r_{j_1} \oplus r_{j_3} \oplus M_{j_3} \oplus ID_j)$ $\alpha_j^N = h(K_{j_1}^N \oplus ID_j)$ $K_{j_2}^N = P(z_j, r_{j_1} \oplus r_{j_2} \oplus r_{j_3} \oplus r_{j_4} \oplus M_{j_3})$ $\beta_j^N = h(K_{j_2}^N \oplus ID_j)$ $M_{j_6} = \gamma_j \oplus h(M_{j_2} \oplus M_{j_3} \oplus r_{j_4})$ $M_{j_7} = \alpha_j^N \oplus h(M_{j_1} \oplus M_{j_2} \oplus M_{j_3} \oplus r_{j_4})$ $M_{j_8} = \beta_j^N \oplus h(M_{j_1} \Vert M_{j_2} \Vert M_{j_3} \Vert r_{j_4})$ $M_{j_9} = h(\alpha_j^N \Vert \beta_j^N \Vert \gamma_j \Vert r_{j_3} \Vert r_{j_4} \Vert M_{j_2} \Vert M_{j_3})$ $M_{j_9} = M_{j_9}^L \Vert M_{j_9}^R$ $\overset{[}{\underset{]}{\longleftarrow}} (M_{j_6}, M_{j_7}, M_{j_8}, M_{j_9}^L)$
	$\overset{[}{\underset{]}{\longleftarrow}} (M_{j_6}, M_{j_7},$ $M_{j_8}, M_{j_9}^L)$	
$\gamma_j^* = M_{j_6} \oplus h(M_{j_2} \oplus h(\beta_j) \oplus r_{j_4})$ $\alpha_j^{N^*} = M_{j_7} \oplus h(M_{j_1} \oplus M_{j_2} \oplus h(\beta_j) \oplus r_{j_4})$ $\beta_j^{N^*} = M_{j_8} \oplus h(M_{j_1} \Vert M_{j_2} \Vert h(\beta_j) \Vert r_{j_4})$ $M_{j_9}^* = h(\alpha_j^N \Vert \beta_j^N \Vert \gamma_j \Vert r_{j_3} \Vert r_{j_4} \Vert M_{j_2} \Vert h(\beta_j))$ $M_{j_9}^* = M_{j_9}^{L^*} \Vert M_{j_9}^{R^*}$ $M_{j_9}^{L^*\,?} = M_{j_9}^L$ $\overset{[}{\underset{]}{\longrightarrow}} M_{j_9}^R$		
	$\overset{[}{\underset{]}{\longrightarrow}} M_{j_9}^R$	
		$M_{j_9}^{R^*} \overset{?}{=} M_{j_9}^R$ $\gamma_j^N = \gamma_j \oplus h(\alpha_j^N) \oplus h(\beta_j^N)$ $\oplus M_{j_1} \oplus M_{j_3} \oplus r_{j_4} \oplus ID_j$ Update r_{j_1} by $r_{j_1} \oplus r_{j_3} \oplus M_{j_3}$ Update r_{j_2} by $r_{j_2} \oplus r_{j_4}$ Update γ_j by γ_j^N
$s_j = \gamma_j \oplus h(\alpha_j) \oplus h(\beta_j)$ Store $(ID_j, \alpha_j^N, \beta_j^N)$ For $0 \le j \le t-1$, obtain s_j and $(ID_0, s_0), (ID_1, s_1), (ID_2, s_2), \ldots, (ID_{t-1}, s_{t-1})$ $f(0) = \sum_{i=0}^{t-1} s_i \prod_{\substack{0 \le j \le t-1 \\ j \ne i}} \frac{ID_j}{ID_j - ID_i} \; (mod\, p)$ Recover $s = f(0) \oplus x$		

1. Foremost, after successful authentication, jth tag computes its new dynamic secret parameters $K_{j_1}^N$, $K_{j_2}^N$, α_j^N and β_j^N, where

$$K_{j_1}^N = P\left(z_j, r_{j_1} \oplus r_{j_3} \oplus M_{j_3} \oplus ID_j\right), \quad \alpha_j^N = h\left(K_{j_1}^N \oplus ID_j\right)$$
$$K_{j_2}^N = P\left(z_j, r_{j_1} \oplus r_{j_2} \oplus r_{j_3} \oplus r_{j_4} \oplus M_{j_3}\right), \quad \beta_j^N = h\left(K_{j_2}^N \oplus ID_j\right)$$

Next *j*th tag send stored secret parameter γ_j along with the new secret parameters α_j^N and β_j^N via sending the messages M_{j_6}, M_{j_7} and M_{j_8}. Also send the left half bits of the message M_{j_9} to verify the authenticity of the communicated message, where

$$M_{j_6} = \gamma_j \oplus h\left(M_{j_2} \oplus M_{j_3} \oplus r_{j_4}\right),\ M_{j_7} = \alpha_j^N \oplus h\left(M_{j_1} \oplus M_{j_2} \oplus M_{j_3} \oplus r_{j_4}\right)$$
$$M_{j_8} = \beta_j^N \oplus h\left(M_{j_1} \| M_{j_2} \| M_{j_3} \| r_{j_4}\right),\ M_{j_9} = h\left(\alpha_j^N \| \beta_j^N \| \gamma_j \| r_{j_3} \| r_{j_4} \| M_{j_2} \| M_{j_3}\right)$$

2. Server then find γ_j, α_j^N and β_j^N from M_{j_6}, M_{j_7} and M_{j_8} respectively and compare left half bits of the message M_{j_9} with the computed $M_{j_9}^{L^*}$. If it find so then server sends right half bits of the message M_{j_9} to the j^{th} tag so that tag also update its memory, where

$$\gamma_j^* = M_{j_6} \oplus h\left(M_{j_2} \oplus h(\beta_j) \oplus r_{j_4}\right),\ \alpha_j^{N^*} = M_{j_7} \oplus h\left(M_{j_1} \oplus M_{j_2} \oplus h(\beta_j) \oplus r_{j_4}\right)$$
$$\beta_j^{N^*} = M_{j_8} \oplus h\left(M_{j_1} \| M_{j_2} \| h(\beta_j) \| r_{j_4}\right),\ M_{j_9}^* = h\left(\alpha_j^N \| \beta_j^N \| \gamma_j \| r_{j_3} \| r_{j_4} \| M_{j_2} \| h(\beta_j)\right)$$

3. After verification of right half bits of the message M_{j_9}, tag accepts the server request to update its secret parameters. Tag update its random parameters r_{j_1} and r_{j_2} by $r_{j_1}^N = r_{j_1} \oplus r_{j_3} \oplus M_{j_3}$ and $r_{j_2}^N = r_{j_2} \oplus r_{j_4}$ respectively. Also tag updates its secret parameter γ_j by γ_j^N, where

$$\begin{aligned}
\gamma_j^N &= \gamma_j \oplus h\left(\alpha_j^N\right) \oplus h\left(\beta_j^N\right) \oplus M_{j_1} \oplus M_{j_3} \oplus r_{j_4} \oplus ID_j \\
&= \left(s_j \oplus h(\alpha_j) \oplus h(\beta_j)\right) \oplus h(\alpha_j^N) \oplus h(\beta_j^N) \oplus \left(ID_j \oplus h(\alpha_j) \oplus r_{j_4}\right) \\
&\quad \oplus h(\beta_j) \oplus r_{j_4} \oplus ID_j \\
&= s_j \oplus h(\alpha_j^N) \oplus h(\beta_j^N) \\
K_{j_1}^N &= P\left(z_j, r_{j_1} \oplus r_{j_3} \oplus M_{j_3} \oplus ID_j\right) \\
&= P\left(z_j, r_{j_1}^N \oplus ID_j\right) \\
K_{j_2}^N &= P\left(z_j, r_{j_1} \oplus r_{j_2} \oplus r_{j_3} \oplus r_{j_4} \oplus M_{j_3}\right) \\
&= P\left(z_j, \left(r_{j_1} \oplus r_{j_3} \oplus M_{j_3}\right) \oplus \left(r_{j_2} \oplus r_{j_4}\right)\right) \\
&= P\left(z_j, r_{j_1}^N \oplus r_{j_2}^N\right)
\end{aligned}$$

4. Consequently server retrieves partial secret key component s_j from γ_j, where

$$s_j = \gamma_j \oplus h(\alpha_j) \oplus h(\beta_j)$$

Server stores $(ID_j, \alpha_j^N, \beta_j^N)$ in its database and to save the protocol from desynchronization attack server will not replace the new ordered pair $(ID_j, \alpha_j^N, \beta_j^N)$ with the existing one $(ID_j, \alpha_j, \beta_j)$ at that particular time and maintain the pair $(ID_j, \alpha_j, \beta_j)$ till synchronized authentication is done.

5. Thus server obtains partial secret key components of any t tags out of n, i.e., $s_0, s_1, s_2, \ldots, s_{t-1}$. Then by getting $(ID_0, s_0), (ID_1, s_1), (ID_2, s_2), \ldots, (ID_{t-1}, s_{t-1})$ server finds the shared secret key parameter $f(0)$ by using lagrange polynomial:

$$f(0) = \sum_{i=0}^{t-1} s_i \prod_{\substack{0 \le j \le t-1 \\ j \neq i}} \frac{ID_j}{ID_j - ID_i} \,(mod\, p)$$

Eventually the shared secret s is recovered by

$$s = f(0) \oplus x$$

where x is the secret key of the server.

6 Security and Privacy Analysis

6.1 Formal Security Proof

In this section, we present formal security analysis of our threshold RFID based mutual authentication protocol and demonstrate that our protocol is secure against various active and passive attacks and achieves destructive privacy. Our mutual authentication protocol is provably secure against side channel attacks. Also tags are untraceable and provides forward as well as backward untraceability.

Theorem 1 *Let an adversary A has full potential of side channel attacks on the tag T_j whose secret keys are K_{j_1} and K_{j_2}. Then an adversary A can either extract the secret key K_{j_1} or the K_{j_2} but not both if $P(z_j, .)$ is an ideal PUF function and $h(.)$ is one way hash function.*

Proof As we know that secret keys K_{j_1} and K_{j_2} are not directly store in the non-volatile memory of jth tag T_j and it is computed via PUF function having physical characteristics z_j and secret random parameters r_{j_1} and r_{j_2} along with ID_j only during an implementation of the protocol, where

$$K_{j_1} = P(z_j, r_{j_1} \oplus ID_j) \quad \text{and} \quad K_{j_2} = P(z_j, r_{j_1} \oplus r_{j_2})$$

In our protocol run firstly secret parameter $\alpha_j = h(K_{j_1} \oplus ID_j)$ and the messages M_{j_1} and M_{j_2} are computed from the secret key K_{j_1} by using hash function and then immediately K_{j_1} and α_j are deleted from its volatile memory. Again in the same manner the secret key K_{j_2} is used in the hash function to compute secret parameter $\beta_j = h(K_{j_2} \oplus ID_j)$ and the messages M_{j_3} and M_{j_4} and then immediately K_{j_2} and β_j are also deleted from its volatile memory. When an adversary A with his full capabilities employs side channel attacks on the tag T_j, then by the properties of PUF function its physical characteristics has been changed and secret parameters cant be calculated accurately. Thus following two cases arises:

1. If an adversary A employs side channel attacks on the tag T_j to get K_{j_1}, then the physical structure of T_j will be damaged and K_{j_2} cant be evaluated. Thus an adversary advantage to extract K_{j_2} when K_{j_1} is known is atmost negligible;

$$ADV_A = Pr[K_{j_2} | K_{j_1}] < \epsilon(\lambda)$$

2. If an adversary A employs side channel attacks on the tag T_j to extract K_{j_2}, then A's advantage to get K_{j_1} is at most negligible as K_{j_1} is already deleted from its volatile memory;

$$ADV_A = Pr[K_{j_1}|K_{j_2}] < \epsilon(\lambda)$$

Thus an adversary A can either extract the secret key K_{j_1} or the K_{j_2} but not both. □

Theorem 2 *Our proposed protocol attains tag authentication if $P(z_j, .)$ is an ideal PUF function and $h(.)$ is one way hash function.*

Proof Let us assume that our proposed protocol does not attain tag authentication, thus there exist an adversary A who acts like a legitimate tag T_j to the reader with non negligible success probability. Also for given r_{j_3}, A's success probability to generate M_{j_1}, M_{j_4} and T_j is non negligible. By the formal model defined in Sect. 4, privacy experiment is composed of following three phases:

1. *Learning Phase*: A get access to set of tags by querying *DrawTag* oracle and analyzes the protocol run between R and T. A can call any oracle query on T. A calls *Free* oracle query to free the chosen tag.
2. *Challenge Phase*: A get access to the tag T_j by querying *DrawTag* oracle and analyze the protocol run between R and T_j. An adversary A is not permitted to call *Corrupt* oracle on that particular tag T_j which makes an adversary unaware of volatile and non volatile information of tag T_j. A call *SendReader* or *SendTag* queries on T_j but A can't evaluate the secret keys K_{j_1} or K_{j_2} in spite of the fact that how many times A analyze the protocol run or how many times A call *SendReader* or *SendTag* queries. A call free oracle query and free the chosen tag.
3. *Guess Phase*: Eventually, when A try to impersonate the target tag T_j by convincing R then R returns a bit ID' for the corresponding tag.

A wins the experiment or successfully impersonates the target tag T_j if $ID' = ID_j$ but for this A has to simulate $P(z_j, .)$ and $h(.)$ functions so that for a given random challenge r_{j_3}, A correctly generates M_{j_1}, M_{j_4} and T_j which contradicts the properties $P(z_j, .)$ and $h(.)$ functions. Thus

$$Pr[ID' = ID_j] < \epsilon(\lambda)$$

Hence our proposed protocol attains tag authentication. □

Theorem 3 *Our proposed protocol attains reader authentication if $P(z_j, .)$ is an ideal PUF function and $h(.)$ is one way hash function.*

Proof Let us assume that our proposed protocol does not attain reader authentication, thus there exist an adversary A who acts like a legitimate reader R to the tag T with non negligible success probability. An adversary A get access to the target tag T_j by querying *DrawTag* oracle and observe the m protocol runs between R and T_j to get $r_{j_{3_k}}, M_{j_{1_k}}, M_{j_{4_k}}, T_{j_k}, M_{j_{5_k}}, M_{j_{6_k}}, M_{j_{7_k}}, M_{j_{8_k}}, M^L_{j_{9_k}}$ and $M^R_{j_{9_k}}$ protocol transcripts for $1 \le k \le m$. A try to impersonate the reader R to the tag T_j, by keeping this goal in her mind, A chooses the random challenge $r_{j_{3_k}}$ from the set $\{r_{j_{3_1}}, r_{j_{3_2}}, r_{j_{3_3}}, \ldots, r_{j_{3_m}}, \}$. W.l.o.g let us assume that $r_{j_{3_k}} = r_{j_3}$. Thus following two cases arises:

1. For given r_{j_3}, if the target tag returns with the same M_{j_1}, M_{j_4} and T_j, then the success probability of an adversary A to respond with the correct authentication factor M_{j_5} is 1

but the parameters M_{j_1} and M_{j_4} depends upon the random challenge r_{j_4} which makes its probability negligible, i.e.

$$Pr[Correct\, M_{j_5}] < \epsilon$$

2. Either to guess or to calculate correct M_{j_5}, an adversary A has to be aware of either (K_{j_1}, K_{j_2}) or (α_j, β_j) or (M_{j_2}, M_{j_3}) but the possibility of guessing or calculating these variables directly depends upon the PUF function and one way hash function which make its probability negligible, i.e.

$$Pr[Correct\, M_{j_5}] < \epsilon(\lambda)$$

Hence our proposed protocol attains reader authentication. □

Theorem 4 *Our proposed protocol attains Destructive Privacy if $P(z_j, .)$ is an ideal PUF function and $h(.)$is one way hash function.*

Proof Let us assume that our proposed protocol does not attain destructive privacy thus there exist a destructive adversary, whose success probability to differentiate real RFID system with the simulation based blinder B generated system is non negligible. Blinded adversary A's destructive privacy game is composed of following three phases:

1. *Learning Phase*: An adversary A foremost get access to set of tags say n by querying *DrawTag* oracle and analyzes the protocol run between R and T. A can send any oracle query on T including *Corrupt* oracle. Finally A calls *Free* oracle query to free the chosen tag, i.e.

$$CreateTag(ID_i) \quad for \quad 0 \leq i \leq n-1$$

$$vtag \leftarrow DrawTag(ID_j) \quad for \quad j \in \{0, 1, 2, ..., n-1\}$$

$$\pi \leftarrow Launch$$

$$r_{j_3} \leftarrow SendReader(Init, \pi)$$

$$M_{j_1}, M_{j_4}, T_j \leftarrow SendTag(r_{j_3}, vtag)$$

$$M_{j_5} \leftarrow SendReader(M_{j_1}, M_{j_4}, T_j, \pi)$$

$$M_{j_6}, M_{j_7}, M_{j_8}, M^L_{j_9} \leftarrow SendTag(M_{j_5}, vtag)$$

$$M^R_{j_9} \leftarrow SendReader(M_{j_6}, M_{j_7}, M_{j_8}, M^L_{j_9}, \pi)$$

$$z_j, r_{j_1}, r_{j_2}, \gamma_j \leftarrow Corrupt(vtag)$$

$$\begin{cases} 1 & \text{if authentication done} \\ 0 & \text{otherwise} \end{cases} \leftarrow Result(\pi)$$

$$Free(vtag)$$

2. *Challenge Phase*: An adversary A get access to two uncorrupted tags $vtag_i$ and $vtag_j$ as its challenge tags and then randomly choose $vtag_b$, $b \in \{i, j\}$ among them. A analyzes the protocol run between R and $vtag_b$ and evaluates all oracles on $vtag_b$ but A is not

permitted to call *Corrupt* oracle on that particular tag $vtag_b$ which makes an adversary unaware of volatile and non volatile information of tag T_b. A calls *Free* oracle query to free the chosen tag.

$$
\begin{aligned}
& CreateTag(ID_i) \quad and \quad CreateTag(ID_j) \\
& Choose\ b \in \{i,j\} \\
& vtag_b \leftarrow DrawTag(ID_b)\ b \in \{i,j\} \\
& \pi \leftarrow Launch \\
& r_{b_3} \leftarrow SendReader(Init, \pi) \\
& M_{b_1}, M_{b_4}, T_b \leftarrow SendTag(r_{b_3}, vtag_b) \\
& M_{b_5} \leftarrow SendReader(M_{b_1}, M_{b_4}, T_b, \pi) \\
& M_{b_6}, M_{b_7}, M_{b_8}, M^L_{b_9} \leftarrow SendTag(M_{b_5}, vtag_b) \\
& M^R_{b_9} \leftarrow SendReader(M_{b_6}, M_{b_7}, M_{b_8}, M^L_{b_9}, \pi) \\
& Free(vtag_b)
\end{aligned}
$$

3. *Guess Phase*: Eventually, an adversary A's privacy game simulation comes to an end with the guess output bit b' for the corresponding tag.

A wins the experiment or its success probability is non negligible if $b' = b$, i.e.

$$Pr[b = b'] = \frac{1}{2} + \epsilon(\lambda)$$

and to achieve this either A knows the secret or she can simulate $P(z_j, .)$ and $h(.)$ functions but she does not know the secret as well as $P(z_j, .)$ and $h(.)$ functions are one way. So our assumption is wrong. Hence our proposed protocol achieves destructive privacy. □

Theorem 5 *Our proposed protocol is untraceable if* $P(z_j, .)$ *is an ideal PUF function and* $h(.)$*is one way hash function.*

Proof Let us assume that our proposed protocol is traceable thus there exist an adversary, whose success probability to trace the tag is non negligible. Based on the traceability definition of [1], an adversary A has not given any permission to call *Corrupt* and *Result* oracles and A's privacy game is composed of following three phases:

1. *Learning Phase*: An adversary A foremost get access to number of tags say n by querying *DrawTag* oracle and analyzes the protocol run between R and T. A can send any oracle query on T excluding *Corrupt* and *Result* oracles. Finally A calls *Free* oracle query to free the chosen tag, i.e.

$$
\begin{aligned}
&CreateTag(ID_i) \quad for \quad 0 \leq i \leq n-1\\
&vtag \leftarrow DrawTag(ID_j) \quad for \quad j \in \{0, 1, 2, ..., n-1\}\\
&\pi \leftarrow Launch\\
&r_{j_3} \leftarrow SendReader(Init, \pi)\\
&M_{j_1}, M_{j_4}, T_j \leftarrow SendTag(r_{j_3}, vtag)\\
&M_{j_5} \leftarrow SendReader(M_{j_1}, M_{j_4}, T_j, \pi)\\
&M_{j_6}, M_{j_7}, M_{j_8}, M_{j_9}^L \leftarrow SendTag(M_{j_5}, vtag)\\
&M_{j_9}^R \leftarrow SendReader(M_{j_6}, M_{j_7}, M_{j_8}, M_{j_9}^L, \pi)\\
&Free(vtag)
\end{aligned}
$$

2. *Challenge Phase*: An adversary A get access to two uncorrupted tags $vtag_i$ and $vtag_j$ as its challenge tags and then randomly choose $vtag_b$, $b \in \{i,j\}$ among them. A queries *SendTag* oracle on $vtag_b$ by sending the previously used learned random variable r_{j_3} and then calls *Free* oracle query to free the chosen tag.

$$
\begin{aligned}
&CreateTag(ID_i) \quad and \quad CreateTag(ID_j)\\
&Choose\ b \in \{i,j\}\\
&vtag_b \leftarrow DrawTag(ID_b),\ b \in \{i,j\}\\
&M_{j_1}^*, M_{j_4}^*, T_j^* \leftarrow SendTag(r_{j_3}, vtag_b)\\
&Free(vtag_b)
\end{aligned}
$$

3. *Guess Phase*: Eventually, an adversary A's privacy game simulation comes to an end with the guess output bit b' for the corresponding tag.

A wins the experiment or its success probability is non negligible if $b' = b$, that is possible only if

$$Pr[M_{j_1}^* = M_{j_1}] = 1 \quad and \quad Pr[M_{j_4}^* = M_{j_4}] = 1$$

but neither $M_{j_1}^* = M_{j_1}$ nor $M_{j_4}^* = M_{j_4}$ as M_{j_1} and M_{j_4} depend upon the pseudo random variable r_{j_4} which is different in each protocol run. So our assumption is wrong. A is unable to trace $vtag_b$ and eventually

$$|Pr(A\, succeeds) - Pr(A^B\, succeeds)| < \epsilon(\lambda) \qquad \square$$

Theorem 6 *Our proposed protocol attains forward untraceability if* $P(z_j, .)$ *is an ideal PUF function and* $h(.)$ *is one way hash function.*

Proof Let us assume that our proposed protocol is forward traceable thus there exist an adversary, who knows all the secrets of ith session and her success probability to trace the tag in $i'(i' > i+1)$ session is non negligible [22]. An adversary A's privacy game is composed of following three phases:

1. *Learning Phase*: An adversary A foremost get access to set of tags say n by querying *DrawTag* oracle and analyzes the protocol run of ith session between R and T by querying any oracle. Also A has full access of all the key parameters $r^i_{j_1}$, $r^i_{j_2}$, $K^i_{j_1}$, $K^i_{j_2}$ and γ^i_j of ith session. Updation algorithm updates $(r^i_{j_1}, r^i_{j_2}, \gamma^i_j)$ by $r^{i+1}_{j_1}, r^{i+1}_{j_2}, \gamma^{i+1}_j$. Finally A calls *Free* oracle query to free the chosen tag.

$$CreateTag(ID_i) \quad for \quad 0 \leq i \leq n-1$$

$$vtag \leftarrow DrawTag(ID_j) \quad for \quad j \in \{0, 1, 2, ..., n-1\}$$

$$\pi^i \leftarrow Launch$$

$$r^i_{j_3} \leftarrow SendReader(Init, \pi^i)$$

$$M^i_{j_1}, M^i_{j_4}, T^i_j \leftarrow SendTag(r^i_{j_3}, vtag)$$

$$M^i_{j_5} \leftarrow SendReader(M^i_{j_1}, M^i_{j_4}, T^i_j, \pi^i)$$

$$M^i_{j_6}, M^i_{j_7}, M^i_{j_8}, M^{L^i}_{j_9} \leftarrow SendTag(M^i_{j_5}, vtag)$$

$$M^{R^i}_{j_9} \leftarrow SendReader(M^i_{j_6}, M^i_{j_7}, M^i_{j_8}, M^{L^i}_{j_9}, \pi^i)$$

$$r^{i+1}_{j_1}, r^{i+1}_{j_2}, \gamma^{i+1}_j \leftarrow UpdationAlg(r^i_{j_1}, r^i_{j_2}, \gamma^i_j)$$

$$Learn\ r^i_{j_1}, r^i_{j_2}, K^i_{j_1}, K^i_{j_2}\ and\ \gamma^i_j$$

$$Free(vtag)$$

2. *Challenge Phase*: An adversary A get access to two uncorrupted tags $vtag_i$ and $vtag_j$ as its challenge tags and randomly choose $vtag_b, b \in \{i,j\}$ among them. A analyzes the protocol run of $i'(i' > i+1)$ session say $i+2$ session between R and $vtag_b$ and evaluates all oracles on $vtag_b$ except *Corrupt* oracle. A calls *Free* oracle query to free the chosen tag.

$$CreateTag(ID_i) \quad and \quad CreateTag(ID_j)$$

$$Choose\ b \in \{i,j\}$$

$$vtag_b \leftarrow DrawTag(ID_b)\ b \in \{i,j\}$$

$$\pi^{i+2} \leftarrow Launch$$

$$r^{i+2}_{b_3} \leftarrow SendReader(Init, \pi^{i+2})$$

$$M^{i+2}_{b_1}, M^{i+2}_{b_4}, T^{i+2}_b \leftarrow SendTag(r^{i+2}_{b_3}, vtag_b)$$

$$Free(vtag_b)$$

3. *Guess Phase*: Eventually, an adversary A's privacy game simulation comes to an end with the guess output bit b' for the corresponding tag.

A wins the experiment or its success probability is non negligible if $b' = b$, that is possible only if she can compute $r^{i+2}_{j_1}$, $r^{i+2}_{j_2}$ and γ^{i+2}_j but due to lack of knowledge about random numbers used in $(i+1)$th session causes ambiguity for her to differentiate b and b'. So our assumption is wrong. A is unable to trace $vtag_b$ in $i' > i$ session and eventually

$$|Pr(A\,succeeds) - Pr(A^B\,succeeds)| < \epsilon(\lambda) \qquad \square$$

Theorem 7 *Our proposed protocol attains backward untraceability if* $P(z_j, .)$ *is an ideal PUF function and* $h(.)$ *is one way hash function.*

Proof Our proposed protocol is also backward untraceable as there exist an adversary, who knows all the secrets of ith session but her success probability to trace the tag in $i'(i' < i - 1)$ session is negligible.

Note: Proof similar to Theorem 6.6. □

6.2 Informal Security Analysis

In spite of attaining tag/reader authentication, achieve destructive privacy, untraceabilily, forward/backward untraceability, resist side channel attack, our protocol is also secure against the following known attacks:

6.2.1 Resist Denial of Service Attack

To resist the server from denial of service attack, reader sends authentication message $(M_{j_1}, M_{j_4}, T_j, r_{j_3})$ to the server along with the counter to avoid computation exhaustive attacks as counter value is increased with each reply of the tag and after the predefined wrong attempts server immediately locks the tag for some specific period.

6.2.2 Resist Man-in-the-Middle Attack

An adversary has no ability to act as the middle man in between the reader and the tag or to modify the communicated messages. An adversary can intercept in the transaction only if he aware of either (K_{j_1}, K_{j_2}) or (α_j, β_j) or (M_{j_2}, M_{j_3}), but the possibility of guessing or calculating these variables directly depends upon the PUF function and one way hash function which make its probability negligible.

6.2.3 Resist Replay Attack

For an adversary, to replay an authentication messages $(M_{j_1}, M_{j_4}, T_j, r_{j_3})$ of one session into another session is useless as the authenticity of the request is verified by checking the freshness of the time stamp T_j and M_{j_4} which enables our protocol to prevent strongly the replay attack.

6.2.4 Resist De-synchronization Attack

Server stores $(ID_j, \alpha_j^N, \beta_j^N)$ in its database and to save the protocol from de-synchronization attack server will not replace the new ordered pair $(ID_j, \alpha_j^N, \beta_j^N)$ with the existing one $(ID_j, \alpha_j, \beta_j)$ in that particular polynomial time and will maintain the pair $(ID_j, \alpha_j, \beta_j)$ till synchronized authentication session can be done. So it will become infeasible for an adversary to de-synchronize the protocol by modifying the communicated messages between reader and the tag.

6.2.5 Resist Cloning Attack

An adversary is unable to clone the registered tag by creating fake tag as each and every tag has its own physical characteristics like supply voltage, temperature, electromagnetic interference, etc. So inbuilt specific PUF function gives different responses for the same challenge for two different tags which makes our protocol secure against cloning or tag compromising attack.

7 Performance Analysis

In this section, as described in Table 4 we analyze and evaluate performance and efficiency of our authentication protocol with the related Molnar-Wagner [23], Bassil et al.'s [24], Kardas et al.'s [17], Zhuang et al.'s [25], Dekhordi-Farzaneh [26], Akglayan [27] and Asadpour-Dashti [28] schemes in terms of storage, tag computation, server computation, communication cost, privacy level, reader complexity level, threshold authentication, key updation, secret key recovery process and various known attacks like side channel attack, traceability attack, impersonation attack, cloning attack, De-synchronization attack, achieve mutual authentication and attain forward and backward untraceability.

Let t_h denotes the time complexity for hash operation, t_p denotes the time complexity for PUF evaluation, t_{xor} denotes the time complexity for xor operation, t_{rot} denotes the time complexity for rotation function and t_{rec} denotes the time complexity for reconstruction function. Since the time complexity for xor operation is negligible, thus we ignore the computational complexity for xor operation. W.l.o.g we assume that the random numbers and the time stamp are as long as the output of one way hash function say, l and the identity message ID_j is padded with zero bits to make the bit size of ID_j as long as l.

In our proposed protocol tags memory, parameters ID_j, r_{j_1}, r_{j_2} and γ_j are stored in our protocol along with its physical characteristics z_j like supply voltage, temperature, electromagnetic interference, etc. which has negligible storage cost. Thus storage cost is $4l$ bits. In our protocol, mutual authentication request messages $\{r_{j_3}\}$, $\{M_{j_1}, M_{j_4}, T_j\}$ and $\{M_{j_5}\}$ require $l + 3l + l = 5l$ bits. Thus communication overhead becomes $5l$ bits during mutual authentication process. During the Authentication phase, tag requires 7 hash computations and 2 PUF evaluations. Thus the total computation cost at tag side is at most $7t_h + 2t_p$. While the computational overhead of Lagrange interpolation technique for the recovery of key is at server side as server has no constraint for resources and for authentication server's computation cost is atmost $O(n)$, where n is the number of tags.

As compared to other's schemes [17, 23–28], increased requirement of computational complexity is not baseless as additional computational cost provides safety from various known attacks, achieves high privacy level and accomplish threshold authentication, as described in Table 4. Thus due to low storage, computation and communication cost and due to provide security against various attacks and due to provide high security level and due to provide threshold authentication; we demonstrate that our protocol is efficient enough to be used practically over insecure networks.

Table 4 Efficiency evaluation

Protocols	Molnar [23]	Bassil [24]	Kardas [17]	Zhuang [25]	Dekhordi [26]	Akgn [27]	Asadpour [28]	Ours
Storage cost	$2l$	$10l$	$3l$	$5l$	$3l$	$4l$	$5l$	$4l$
Tag computation cost	$t_h + t_p$	$7t_{rot} + t_p$	$4t_h + 2t_p$	$10t_{rec} + 4t_{rot}$	$2t_h$	$4t_h + 2t_p$	$3t_h$	$7t_h + 2t_p$
Server computation cost	$O(n)t_h + t_p$	$4t_{rot}$	$2t_h$	$9t_{rec} + 4t_{rot}$	$O(n)t_h + t_p$	$3t_h$	$3t_h$	$5t_h$
Communication cost	$4l$	$7l + 6 * 5byte$	$6l$	$6l$	$5l$	$5l$	$6l$	$5l$
Privacy	Narrow weak	Narrow weak	Narrow destructive	Narrow weak	Narrow weak	Narrow destructive	No privacy	Narrow destructive
Mutual authentication	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Threshold authentication	No	No	No	No	No	No	No	Yes
Key updation	No	Yes	No	Yes	Yes	No	Yes	Yes
Secret key recovery process	No	No	No	No	No	No	No	Yes
Resist side channel attack	No	No	Yes	No	No	No	No	Yes
Resist traceability attack	No	No	Yes	No	Yes	Yes	Yes	Yes
Resist impersonation attack	Yes	No	Yes	No	Yes	Yes	Yes	Yes
Resist coning attack	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Resist De-synchronization attack	N.A	No	N.A.	Yes	Yes	N.A	Yes	Yes
Attain forward untraceability	No	No	No	No	No	No	No	Yes
Attain backward untraceability	No	No	No	No	No	No	Yes	Yes
Reader complexity	$O(logn)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$

8 Conclusion

In this paper, we extend well-known Vaudenay's RFID privacy model to make the RFID system acceptable for threshold secret sharing system among n tags. To resist tag compromising or cloning attack, we employ PUF function in tags. Next, to implement threshold RFID system we have designed (t, n) threshold RFID mutual authentication protocol based on physically unclonable function with the aim to ensure secure communication through an insecure channel, to resist tag compromising attack and to enhance shared control of the secret among multiple tags. It is a method of distributing a secret s among a set of n RFID tags in such a way that any group of t or more tags will recover the secret s only after successful mutual authentication by using lagrange interpolation. In order to enhance anonymity and untraceability of tags, we use dynamic security parameters which are updated after each successful run of mutual authentication protocol. In-spite of low storage capacity and limited computation and communication cost, our mutual authentication protocol achieves destructive privacy, untraceabilily, forward/backward untraceability, withstand against side channel attack, denial of service attack, man-in-the-middle attack, replay attack, de-synchronization attack and cloning attack which makes our protocol secure and efficient to be used practically over insecure networks.

References

1. Vaudenay, S. (2007). On privacy models for RFID. In K. Kurosawa (Ed.), *Advances in Cryptology – ASIACRYPT 2007. Lecture Notes in Computer Science* (Vol. 4833, pp. 68–87). Berlin: Springer.
2. Kaul, S. D., & Awasthi, A. K. (2013). RFID authentication protocol to enhance patient medication safety. *Journal of medical systems*, *37*(6), 1–6.
3. Tuyls, P., & Batina, L. (2006). RFID-tags for anti-counterfeiting. In D. Pointcheval (Ed.), *Topics in Cryptology – CT-RSA 2006. Lecture Notes in Computer Science* (Vol. 3860, pp. 115–131). Berlin: Springer.
4. Dodis, Y., Ostrovsky, R., Reyzin, L., & Smith, A. (2008). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, *38*(1), 97–139.
5. Shamir, A. (1979). How to share a secret. *Communications of the ACM*, *22*(11), 612–613.
6. Avoine, G. (2005). Adversarial model for radio frequency identification. *IACR Cryptology ePrint Archive, 2005*, 49.
7. Lim, C., & Kwon, T. (2006). Strong and robust RFID authentication enabling perfect ownership transfer. *Information and Communications Security*, *4307*, 1–20.
8. Juels, A., & Weis, S. A. (2009). Defining strong privacy for RFID. *ACM Transactions on Information and System Security (TISSEC)*, *13*(1), 1–7.
9. Paise, R.I., & Vaudenay, S. (2008). Mutual authentication in RFID: Security and privacy. In Proceedings of the 2008 ACM symposium on information, computer and communications security, (pp. 292–299).
10. Deng, R. H., Li, Y., Yung, M., & Zhao, Y. (2010). A new framework for RFID privacy. *Computer Security-ESORICS*, *2010*, 1–18.
11. Hermans, J., Pashalidis, A., Vercauteren, F., & Preneel, B. (2011). A new RFID privacy model. *Computer Security-ESORICS*, *2011*, 568–587.
12. Coisel, I., & Martin, T. (2013). Untangling RFID privacy models. *Journal of Computer Networks and Communications, 2013*, 1–26. doi:10.1155/2013/710275.
13. Blakley, G. R. (1979). Safeguarding cryptographic keys. *Proceedings National Computer Conference*, *48*, 313–317.
14. Desmedt, Y. G., & Frankel, Y. (1994). Homomorphic zero-knowledge threshold schemes over any finite abelian group. *SIAM Journal on Discrete Mathematics*, *7*(4), 667–679.
15. Kurihara, J., Kiyomoto, S., Fukushima, K., & Tanaka, T. (2008). A new (k, n)-threshold secret sharing scheme and its extension. In: T. C. Wu, C. L. Lei, V. Rijmen & D. T. Lee (Eds.), *Information security. ISC 2008. Lecture Notes in Computer Science* (Vol. 5222, pp. 455–470). Berlin: Springer.

16. Wang, D., Zhang, L., Ma, N., & Li, X. (2007). Two secret sharing schemes based on Boolean operations. *Pattern Recognition*, *40*(10), 2776–2785.
17. Kardas, S., Celik, S., Yildiz, M., & Levi, A. (2012). PUF-enhanced offline RFID security and privacy. *Journal of Network and Computer Applications*, *35*(6), 2059–2067.
18. Kardas, S., Celik, S., Bingol, M.A., Kiraz, M.S., Demirci, H., & Levi, A. (2014). k-strong privacy for radio frequency identification authentication protocols based on physically unclonable functions, wireless communications and mobile computing
19. Sadeghi, A.R., Visconti, I., & Wachsmann, C. (2010). PUF-enhanced RFID security and privacy, Workshop on secure component and system identification (SECSI)
20. Bellare, M. (2002). A note on negligible functions. *Journal of Cryptology*, *15*(4), 271–284.
21. Stinson, D. R. (2005). *Cryptography: Theory and practice*. Boca Raton: CRC press.
22. Alagheband, M. R., & Aref, M. R. (2014). Simulation based traceability analysis of RFID authentication protocols. *Wireless Personal Communications, 77*(2), 1019–1038.
23. Molnar, D., & Wagner, D. (2004). Privacy and security in library RFID: Issues, practices, and architectures. In Proceedings of the 11th ACM conference on computer and communications security, ACM, 210–219
24. Bassil, R., El-Beaino, W., Itani, W., Kayssi, A., & Chehab, A. (2012). PUMAP: A PUF-based ultralightweight mutual-authentication RFID protocol. *International Journal of RFID Security and Cryptography*, *1*(1/2), 58–66.
25. Zhuang, X., Zhu, Y., & Chang, C. C. (2014). A new ultralightweight RFID protocol for low-cost tags: *R^2AP*. *Wireless Personal Communications*, *79*(3), 1787–1802.
26. Dehkordi, M. H., & Farzaneh, Y. (2014). Improvement of the hash-based RFID mutual authentication protocol. *Wireless personal communications*, *75*(1), 219–232.
27. Akgn, M., & alayan, M. U. (2015). Providing destructive privacy and scalability in RFID systems using PUFs. *Ad Hoc Networks*, *32*, 32–42.
28. Asadpour, M., & Dashti, M. T. (2015). Scalable, privacy preserving radiofrequency identification protocol for the internet of things. *Concurrency and Computation Practice and Experience*, *27*(8), 1932–1950.

Sonam Devgan Kaul is a research scholar in Department of Mathematics, Gautam Buddha University, Greater Noida, India. She has teaching and research experience of more than Six years. She has a number of publications to her credit in various reputed International SCI journals related to security of lightweight authentication on smart cards, RFID, etc.

Amit Kumar Awasthi is Assistant Professor of Mathematics in Department of Mathematics, School of Vocational Studies & Applied Sciences, Gautam Buddha University. He has completed his Ph.D. studies in cryptography at the Dr BR Ambedkar University. He has teaching experience of more than 15 years. He has published several papers in various reputed International SCI journals. His research interest is in Cryptanalysis, Network security, Smartcards, Authentication, Natural Language Progressing and Numerical analysis.