Cyclic group based mutual authentication protocol for RFID system

Pramod Kumar Maurya1 · Satya Bagchi1

Published online: 9 October 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract
Widespread deployment of RFID system arises security and privacy concerns of users. There are several proposals in the literature to avoid these concerns, but most of them provide reasonable privacy at the cost of search complexity on the server side. The search complexity increases linearly with the number of tags in the system. Some schemes use a group based approach to solve the search complexity problem. In this paper, we proposed a group based authentication protocol for RFID system which is based on some characteristics of cyclic groups. The scheme uses only bitwise XOR, circular left shift (<<) and modulo operations for the computational work. Also, the scheme does not use any pseudo-random number generator on the tag side. We use two benchmark metric based on anonymity set to measure the privacy level of the system when some tags are compromised by an adversary. We present some simulation results which show that the scheme preserves the high level of privacy and discloses very less amount of information when some tags are compromised. Furthermore, its formal and informal analysis show that our scheme preserves information privacy as well as untraceability and also withstand various well-known attacks.

Keywords RFID system · Cyclic group · Anonymity · Authentication protocol · Security · Privacy

1 Introduction
RFID technology is becoming a most promising technology in industries to improve the efficiency of tracking and managing goods. Because of its convenience use and low-cost, we encounter this technology in various applications like supply chain management, logistics, access control, manufacturing, e-health, passport verification etc [5, 8, 16, 22].

RFID system is made up of three entities: tags, readers, and a back-end server. Each tag comprises a microchip for storing and processing data, and an antenna for receiving and transmitting data. The server stores all the information about the tags and connected with the readers via a secure channel while the readers communicate with the tags over an insecure channel [4, 7, 9, 21].

With the widespread adoption of the RFID system in our daily life, security and privacy concerns are also arise critically [2, 6, 15, 17, 20]. We can use cryptographic tools to avoid these concerns but the main obstacle to deploying these tools in the RFID system is tight constraints on power, memory and computational capability on the tags.

To enhance the security and privacy of the RFID system and to reduce the computational complexity, researchers have been proposed a large number of authentication schemes. In the literature, there are several symmetric key based authentication schemes that resist various types of threats such as tag impersonation attack, replay attack, tag location tracking, private data leakage etc [12, 23]. However, these schemes do not suitable for a large RFID system because the search complexity of the server to search a tag in the database (DB) is \(O(N)\), where \(N\) refers to the total number of tags in the system. The search complexity increases linearly with the number of tags in the system.

To reduce the search complexity of the server, researchers have been proposed several tree-based authentication schemes for the RFID system. These schemes reduce the search complexity of the server from \(O(N)\) to \(O(\log N)\). In 2004, Molnar and Wagner [13] proposed a tree-based symmetric key authentication scheme.
The scheme uses a balanced secret key-tree in which each tag is associated with a leaf of the tree. In addition, a unique secret key is associated with each branch of the tree. In the scheme, each tag stores all the keys along with the path from the root to itself in its internal memory. In this way, each tag shares some secret keys with some other tags in the system. A tag uses all of its keys during an authentication process, starting from the first level and proceeding towards the last level of the tree. To authenticate a tag in the tree, the server identifies the first level key that is used by the tag from the pool of the first level keys. After that, the server identifies the second level key from the pool of the second level keys that reside below the identified first level key in the tree. The server continues this process until the tag authentication succeed. The scheme reduces its search complexity from linear to logarithmic. However, it violates the privacy of the other tags when some tags are compromised by an adversary. In 2005, Nohara et al. [14] proposed a similar kind of authentication scheme. The scheme provides higher privacy than [13] in case of one tag is compromised. Buttyán et al. [3] proposed optimal key trees for tree-based private authentication scheme in 2006. They used different branching factors at different levels of the tree to enhance the privacy level of the scheme. Also, they introduce a benchmark metric for measuring privacy level of the system when some tags are compromised.

As we mentioned above, in the tree-based authentication schemes, each tag shares some secret keys with some other tags in the system. So, if a tag is compromised by an adversary, the tag reveals some secrets of the other tags. That means the compromised tag scarifies some privacy of the other tags that share a secret key with the compromised tag. This is the main drawback of the tree-based approach. To overcome this problem, In 2007, Avoine et al. [1] introduced a group based symmetric key authentication scheme. They improved the trade-off between scalability and privacy by dividing the set of all tags into some distinct groups. In the scheme, all the tags are divided into some distinct groups with equal size. Each tag has a secret group-key $K\text{ID}$ which is common for each tag that belongs to the same group. In addition, each tag has a unique secret key $K\text{ID}$ as well as a unique identification number $ID$. To authenticate a tag, the reader queries the tag by sending a nonce $r_1$. The tag generates a nonce $r_2$ and forms a response messages $E_K(r_1||r_2||ID)||E_{K\text{ID}}(r_1||r_2)$. The response message has two parts, $E_K(r_1||r_2||ID)$ and $E_{K\text{ID}}(r_1||r_2)$. In the first part, the tag uses group-key $K$ for encryption and in the second part, the tag uses its unique secret key $K\text{ID}$ for encryption. Upon receiving the response message from the tag, the reader decrypts the first part of the response message by using all the group-keys until it succeeds. If so, the reader uses $ID$ of the tag to get the unique secret from the database and decrypts the second part. In the worst case, the scheme performs $\gamma + 1$ decryptions to authenticate a tag, where $\gamma$ denotes the total number of groups in the system. Also, the authors analyze the privacy level of the system by using privacy metric when a single tag is compromised as well as any number of tags are compromised. With the group-based approach, the scheme achieves better privacy than the tree-based approach when some tags are compromised in the system.

In 2017, Rahman et al. [18] proposed a secure anonymous group based authentication scheme. The scheme is similar to Avoine et al. except that the scheme uses a different grouping technique to achieve better privacy than [1]. In the scheme, the set of all tags is divided into some distinct groups of equal size. Each group has a unique pool of identifiers. Each tag stores a secret group-key which is common for all the tags that belong to the same group. Also, each tag stores a unique secret key and a set of identifiers from the pool of identifiers of the group. Each tag shares some of its identifiers with some other tags of its group. So, each identifier in each group is associated with a unique set of secret keys of the tags that share the identifier with each other. To authenticate a tag, the reader queries a tag by passing a nonce to the tag. The tag responds to the reader by sending a response message. The response message consists of two parts same as in [1]. In the first part, the tag uses group-key for encryption and in the second part, the tag uses its unique secret key for encryption. After receiving the response message from the tag, the reader decrypts the first part of the response message by trying all the group-keys until it succeeds. If the reader decrypts the first part successfully, the reader gets the identifier of the tag which is used in the first part. After that, the reader tries to decrypt the second part of the response message by using all the secret keys that are associated with the identifier in the group until it succeeds. The scheme needs $\gamma + |\pi|$ decryptions to authenticate a tag, where $\gamma$ denotes the total number of groups in the system and $|\pi|$ refers to the total number of tags associated with the identifier. The scheme uses privacy metric same as in [1] to measure the privacy level of the system. Also, the scheme uses information leakage metric based on Shannon information theory [19] to measure the information leakage in bits of the system. Although the scheme needs more storage requirements as well as more computational work than the Avoine et al. [1], the scheme provides higher privacy than when some tags are compromised by an adversary.

After reviewing the work done, we would like to propose a cyclic group based authentication scheme. In our approach, we divide the set of all tags into some subgroups of a finite cyclic group. Also, we use some different kind of
techniques to improve privacy as well as minimize computational cost.

Rest of the paper is organized as follows: In Sect. 2, we discuss preliminaries and details of our system model. We present adversary model in Sect. 3. Group based authentication scheme for RFID system is proposed in Sect. 4. The formal and informal analysis are given in Sect. 5. Section 6 illustrates the performance of the proposed scheme. In Sect. 7, we measure the level of privacy of the system when some tags are compromised by an adversary. We discuss simulation results in Sect. 8. Finally, conclusions are made in Sect. 9.

2 Preliminaries and system model

In this section, we give a brief overview of a cyclic group [10] and using its properties, we develop our RFID system model. In this system model, we assume that a reader and the server communicate with each other via a secure communication channel. For simplicity, we assume that the reader and the server are combined into one entity, called reader.

Suppose $G$ be a nonempty set together with an operation * that combines any two elements of $G$ is in $G$. The set $G$ together with this operation is a group if it holds group’s law. The order of the group $G$ is the total number of elements in the group. It is denoted by $|G|$. Let $H$ be a non-empty subset of the group $G$. We say that $H$ is a subgroup of $G$ if it is itself a group under the operation of $G$.

**Definition 2.1** A group $G$ is called cyclic if there exists an element $a \in G$ such that $G = \langle a \rangle = \{a^n | n \in \mathbb{Z}\}$. The element $a$ is called a generator of $G$.

Some important characteristics of cyclic groups are as follows:

1. Suppose $G = \langle a \rangle$ be a cyclic group of order $n$. Then $G = \langle a^k \rangle$ iff $gcd(k, n) = 1$.
2. Every subgroup of a cyclic group is cyclic.
3. Suppose $G = \langle a \rangle$ be a cyclic group of order $n$. The order of any subgroup of $G$ is a divisor of $n$.
4. For each positive divisor $k$ of $n$, the group $G$ has exactly one subgroup of order $k$ denoted by $\langle a^{n/k} \rangle$.

We construct a system model for RFID system with the help of characteristics as mentioned earlier of cyclic groups. In this system model, we divide the set of all tags into some distinct clusters. For this, we choose a finite cyclic group $G = \langle a \rangle$ of order $n$. According to the system requirement, we randomly choose some distinct subgroups $H_j = \langle a_j \rangle, j = 1, 2, \ldots, p$, where $a_j = a^{n/k}$ for some positive divisor $k$ of $n$. With each element except the identity of a subgroup, we can associate a tag together with some secret parameters. In this way, we can divide the set of all tags into some distinct subgroups of a finite cyclic group. The total number of the tags that are associated with each subgroup may be different. i.e. the size of each cluster need not be the same. Here, we define the components and parameters of the system.

2.1 Reader

In our system, we assume that the reader and the back-end server work as a single entity, called reader. The reader stores all the information about each tag that is associated with each subgroup in the system. The reader possesses a unique identification number, a key, and a pair of nonces for each tag in the system. Suppose a tag $T_{ji}$ is associated with an element $a_j$ of a subgroup $H_j = \langle a_j \rangle$. The reader uses $i$ as an index in the system database (DB) for the tag $T_{ji}$. The reader stores a unique identification number $ID_{ji}$ and a key $K_{ji}$ for the tag $T_{ji}$. In addition, the reader also stores a pair of nonce $(r^i_{ji}, r^i_{jnew})$ for the tag $T_{ji}$ in the DB. $r^i_{jnew}$ refers to a nonce that is used in the previous authentication session. $r^i_{jnew}$ denotes a nonce which is used in the current authentication session. Initially, $r^i_{jold}$ is set to null and $r^i_{jnew}$ be a nonce. The database lookup table for the reader is shown in Table 1. Table 1 shows that the set of all tags is divided into some randomly chosen subgroups of the group $G$. With each subgroup $H_j$, we can assign a maximum number of $|H_j| - 1$ tags, where $|H_j|$ denotes the order of the subgroup $H_j$. However, suppose $H_p$ is the highest order subgroup in the system with order $P$ and $|H_p|$ is the second highest order subgroup (say, $|H_q| = Q$) in the system. For untraceability, we can utilize only $Q - 1$ number of elements of the subgroup $H_p$ in the system in such a way so that the value of index $i$ is same for both the subgroups (as shown in Table 1).

2.2 Tag

All the tags of the system are divided into some distinct subgroups of a finite cyclic group as mentioned above. Each tag possesses inverse of an element of a subgroup that is associated with it in the database and an index. In addition, each tag receives a unique identification number, a key, and a nonce. The tag $T_{ji}$ possesses $(a_j)^{-1}$, index $i$, a unique identification number $ID_{ji}$, a key $K_{ji}$, and a nonce $R_4$ inside its internal memory. Initially, $R_4$ is same as $r^i_{jnew}$ which is stored in the reader’s lookup table for the tag $T_{ji}$. In this system model, we use $(a_j)^{-1}$ instead of $(a_j)$. Suppose the tag $T_{ji}$ is compromised by an adversary, the adversary knows all the secret parameters that are stored in
the tag’s internal memory, i.e. \((a_j^i)^{-1}, t, ID_{ji}, K_{ji}, r_{ji}\), and \(R_{ji}\). However, the adversary cannot guess the element \((a_j^i)\) (the tag \(T_{ji}\) is associate with \((a_j^i)\) in the DB) without knowledge of the subgroup \(H_{ji}\) or the mother group \(G\).

In this system model, the chosen finite cyclic group as well as all the subgroups and their generators are private. In addition, all the parameters that are stored in a tag’s internal memory are also private except index \(i\). We assume that all the parameters in the system model, i.e., unique identification number, key, nonces, and elements of the group have a length of \(L\)-bits.

### 3 Adversary model

In this section, we present the ability of an adversary \(\mathcal{A}\). The adversary is capable to interact with the RFID system \(S\) and also, eavesdrops, intercepts, and modifies any transmitted message between any reader and any tag in the system. Our adversarial model is similar to the model proposed by Juels and Weis [11] with some modifications to meet our requirement. \(\mathcal{A}\) is also able to send the following queries to an oracle.

1. **SendTag** \((m, T_{ji}) \rightarrow m'\)
   The adversary \(\mathcal{A}\) may send a message \(m\) to a tag \(T_{ji}\) which responds with message \(m'\).

2. **SendReader** \((m, R) \rightarrow m'\)
   \(\mathcal{A}\) can interact with a reader \(R\) by sending a message \(m\). The reader \(R\) responds with message \(m'\).

3. **DrawTags** \((S)\)
   The adversary has access to a set of tags at any time from the system with this oracle query.

4. **Corrupt** \((T_{ji})\)
   \(\mathcal{A}\) is able to access the volatile memory as well as the nonvolatile memory of the tag \(T_{ji}\).

We also bound the adversary \(\mathcal{A}\) to use SendTag and SendReader queries by \(r\) and \(t\), respectively. \(\mathcal{A}\) can perform \(s\) number of computational steps. At a time, \(\mathcal{A}\) is able to send Corrupt message to atmost \((n_j - 2)\) number of tags where \(n_j\) is the total number of tags obtained from DrawTags query.

### 3.1 Privacy experiment

We denote privacy experiment for an RFID system \(S\) by \(\text{EXP}^\text{priv}_{\mathcal{A}, S}[k, n_1, r, s, t]\), where \(r, s,\) and \(t\) represent the capability of an adversary to use SendTag, computations steps and SendReader respectively. Also, \(k\) represents a security parameter. An RFID authentication protocol is considered to be private if the adversary has no significant advantage in this experiment.

The main goal of the adversary in the experiment is to distinguish between two different tags within its computational and interaction limits. The experiment is composed in three phases as follows:

1. **Learning phase** The adversary \(\mathcal{A}\) interacts with the system \(S\) and inquiries oracle queries without exceeding its bound and analyze them.
2. **Challenging phase** \( \mathcal{A} \) selects two uncorrupted tags from the pool obtained by DrawTags oracle. \( \mathcal{A} \) randomly selects any one from them. The adversary evaluates oracles on that particular tag.

3. **Guessing phase** \( \mathcal{A} \) outputs a guess bit \( b \). \( \mathcal{A} \) is expected to produce 1 if it succeeds, otherwise 0.

\( \text{EXP} \) succeed if \( b = 1 \).

### 3.2 Privacy definition \(((r, s, t)-\text{privacy})\)

According to Juels and Weis [11], an RFID authentication protocol with security parameter \( k \) is \((r, s, t)-\text{private} \) if

\[
\Pr[\text{EXP}_{\mathcal{A},S}^{\text{priv}}[k, n_1, r, s, t] \text{ succeeds in guessing } b] \leq \frac{1}{2} + \frac{1}{\text{poly}(k)},
\]

where \( \text{poly}(k) \) is any polynomial function of \( k \).

### 4 Process

In this section, we propose a group based authentication protocol which works under all circumstances required for RFID system. Used notations in this protocol are given in Table 2, and proposed protocol is shown in Fig. 1. The workflow of the proposed scheme is as follows:

1. **msg** \(_1\) : \( T_{ji} \rightarrow R \) : \( \{i, \alpha\} \)

   The tag \( T_{ji} \) computes \( \alpha = (a_j)^{-1} \oplus R_4 \) and forms a request message \( \text{msg}_1 = \{i, \alpha\} \). The tag sends \( \text{msg}_1 \) to a reader \( R \).

2. **msg** \(_2\) : \( R \rightarrow T_{ji} \) : \( \{\beta, \gamma\} \)

   After receiving the tag’s request message \( \text{msg}_1 \), the reader uses \( i \) as an index (as in Table 1) to perform the following steps for all the subgroups until it finds the right tag:

   (a) It calculates the inverse of \( a_j \) in \( H_j \), where \( a_j \) is the generator of the subgroup \( H_j \).

   (b) The reader computes \( \alpha' = (a_j)^{-1} \oplus r_{j,\text{adv}} \) and \( \alpha'' = (a_j)^{-1} \oplus r_{j,\text{rev}} \) and checks whether \( \alpha' \) or \( \alpha'' \) is equal to the received \( \alpha \) or not. If so, it gets the right tag \( T_{ji} \) (say) inside the subgroup \( H_j \). If fails, the reader terminates the protocol. Suppose \( \alpha'' \) is same as the received \( \alpha \). So, the nonce \( r_{j,\text{rev}} \) is used in the current authentication session.

   (c) The reader generates two nonce \( R_1 \) and \( R_2 \), and computes \( \beta = R_1 \oplus K_{ji}, \gamma = R_2 \oplus K_{ji} \), where \( K_{ji} \) is the key of the tag \( T_{ji} \). The reader forms a response message \( \text{msg}_2 = \{\beta, \gamma\} \) and transmits it to the tag.

3. **msg** \(_3\) : \( T_{ji} \rightarrow R \) : \( \{\delta\} \)

   Upon receiving the response message \( \text{msg}_2 \), the tag \( T_{ji} \) extracts \( R_1 \) and \( R_2 \) from \( \beta \) and \( \gamma \) respectively with the help of its key \( K_{ji} \). It computes \( \delta = \{(ID_{ji} \oplus R_1) \ll wt(R_1)\} \mod ((a_j)^{-1} \oplus R_2) \) and sends \( \delta \) inside the response message \( \text{msg}_3 = \{\delta\} \) to the reader.

4. **msg** \(_4\) : \( R \rightarrow T_{ji} \) : \( \{\zeta, \eta\} \)

   After receiving message \( \text{msg}_3 \) from the tag \( T_{ji} \), the reader calculates \( \delta' = \{(ID_{ji} \oplus R_1) \ll wt(r_{j,\text{rev}})\} \mod ((a_j)^{-1} \oplus R_2) \) for the tag \( T_{ji} \) and checks whether \( \delta' \) is equal to the received \( \delta \) or not. If it holds, the reader authenticates the tag \( T_{ji} \) otherwise terminates the session. If tag’s authentication succeed, the reader generates a nonce \( R_3 \) and computes \( \zeta = R_3 \oplus K_{ji}, \quad \eta = \{(ID_{ji} \oplus R_3) \ll wt(R_2)\} \mod ((a_j)^{-1} \oplus R_1) \) and forms a response message \( \text{msg}_4 = \{\zeta, \eta\} \). Simultaneously, also assigns nonce of the current session \( r_{j,\text{rev}} \) in \( r_{j,\text{adv}} \) and updates nonce stored in \( r_{j,\text{adv}} \) by \( R_3 \). It sends \( \text{msg}_4 \) to the tag \( T_{ji} \).

5. Upon receiving message \( \text{msg}_4 \), the tag \( T_{ji} \) extracts \( R_3 \) from \( \zeta \) and calculates \( \eta' = \{(ID_{ji} \oplus R_3) \ll wt(R_2)\} \mod ((a_j)^{-1} \oplus R_1) \). The tag checks whether \( \eta' \) is equal to the received \( \eta \) or not.
If so, the tag authenticates the reader and updates $R_4 = R_3$ for further communication.

In the proposed scheme, we mask the unique identification number $ID_{ji}$ by using circular left shift operation and modulo operation for un-traceability in the messages $msg_3$ and $msg_4$. But we do not mask the key $K_{ji}$ in the messages $msg_2$ and $msg_4$. Because if we mask the key $K_{ji}$ in the messages $msg_2$ and $msg_4$, the tag needs more computational work to extract nonces i.e., $R_1$, $R_2$ and $R_3$ from the messages $msg_2$ and $msg_4$. RFID tags have very resource constraints, so we try to use very less computational work on the tag side in the proposed scheme.

5 Security and privacy analysis

In this section, we present a formal and informal analysis of our proposed scheme with respect to the adversary model. The formal analysis shows that the proposed scheme preserves privacy and un-traceability. Also, its informal analysis shows that the proposed scheme is secure against various well-known attacks.

5.1 Formal security analysis

Theorem 1 The proposed scheme attains information privacy with respect to an adversary $\mathcal{A}$.

Proof We assume that the proposed scheme does not preserve information privacy. So the success probability of the adversary to win the experiment is non-negligible. $\mathcal{A}$’s privacy game is composed in three phases as follows:
• **Learning phase** The adversary gets a set of $n_1$-tags by querying DrawTags oracle. $\mathcal{A}$ can send any oracle queries to a tag $T_{ji}$ (say) without exceeding its computational bound and analyzes them. $\mathcal{A}$ can use Corrupt oracle to atmost $n_1 - 2$ tags.

$$T_{ji} \leftarrow \text{DrawTag}(S)$$

$\text{msg}_1 = \{i, x\} \leftarrow \text{SendTag}(\text{init}, T_{ji})$

$\text{msg}_2 = \{\beta, \gamma\} \leftarrow \text{SendReader}(\text{msg}_1, R)$

$\text{msg}_3 = \{i, \delta\} \leftarrow \text{SendTag}(\text{msg}_2, T_{ji})$

$\text{msg}_4 = \{\zeta, \eta\} \leftarrow \text{SendReader}(\text{msg}_3, R)$

$$\{i, (\delta')^{-1}, K_{ji}, ID_{ji}, R_4\} \leftarrow \text{Corrupt}(T_{ji}).$$

• **Challenge phase** The adversary $\mathcal{A}$ selects two uncorrupted tags say, $T_{ji}$ and $T_{mo}$, from the set of tags obtained by DrawTags query as its challenge tags. Let $T_0 = T_{ji}$, $T_1 = T_{mo}$, and $b \in \{0, 1\}$. $\mathcal{A}$ randomly selects $T_b$ among them and analyzes all queries run on it. Note that $\mathcal{A}$ is not able to use Corrupt oracle on that particular tag $T_b$.

$$\text{msg}_1 = \{i, x\} \leftarrow \text{SendTag}(\text{init}, T_b)$$

$$\text{msg}_2 = \{\beta, \gamma\} \leftarrow \text{SendReader}(\text{msg}_1, R)$$

$$\text{msg}_3 = \{i, \delta\} \leftarrow \text{SendTag}(\text{msg}_2, T_b)$$

$$\text{msg}_4 = \{\zeta, \eta\} \leftarrow \text{SendReader}(\text{msg}_3, R).$$

• **Guess phase** Eventually, the adversary outputs a guess bit $b'$ for the corresponding tag.

## Theorem 2

The proposed scheme provides un-traceability with respect to the adversary $\mathcal{A}$.

**Proof** Let us assume that the proposed scheme is traceable, i.e. the adversary can trace a tag at any time. This means $\mathcal{A}$ is able to distinguish between two tags. We show that our assumption is wrong with the help of $\mathcal{A}$’s privacy game which is as follows:

• **Learning phase** $\mathcal{A}$ uses DrawTags query for the system $S$ and gets access to $n_1$-tags. For all the tags, $\mathcal{A}$ sends SendTag and SendReader queries to get transmitted information between a reader and tags. The adversary analyzes all the transmitted messages. The adversary can use Corrupt query for almost $n_1 - 2$ tags because the goal of the privacy game is to distinguish between two uncorrupted tags.

$T_{ji} \leftarrow \text{DrawTag}(S)$

$\text{msg}_1 = \{i, x\} \leftarrow \text{SendTag}(\text{init}, T_{ji})$

$\text{msg}_2 = \{\beta, \gamma\} \leftarrow \text{SendReader}(\text{msg}_1, R)$

$\text{msg}_3 = \{i, \delta\} \leftarrow \text{SendTag}(\text{msg}_2, T_{ji})$

$\text{msg}_4 = \{\zeta, \eta\} \leftarrow \text{SendReader}(\text{msg}_3, R)$

$$\{i, (\delta')^{-1}, K_{ji}, ID_{ji}, R_4\} \leftarrow \text{Corrupt}(T_{ji}).$$

• **Challenge phase** The adversary selects two uncorrupted tags $T_{ji}$ and $T_{mo}$ to which it did not send Corrupt query in the learning phase. $\mathcal{A}$ randomly selects $T_b : b \in \{js, mt\}$ among them. The adversary queries all the oracle queries except Corrupt query to the tag $T_b$ and evaluates them.

$$\text{msg}_1' = \{i, x\} \leftarrow \text{SendTag}(\text{init}, T_b)$$

$$\text{msg}_2' = \{\beta, \gamma\} \leftarrow \text{SendReader}(\text{msg}_1', R)$$

$$\text{msg}_3' = \{i, \delta\} \leftarrow \text{SendTag}(\text{msg}_2', T_b)$$

$$\text{msg}_4' = \{\zeta, \eta\} \leftarrow \text{SendReader}(\text{msg}_3', R).$$

• **Guess phase** $\mathcal{A}$ outputs a guess bit $b'$.

The adversary wins the game if $b' = b$ but it is possible only when if

$$\Pr[\text{msg}_1' = \text{msg}_1] = 1$$

Since the message $\text{msg}_1$ depend upon the tag’s nonce $R_4$ which is different in each protocol run. So our assumption is wrong. Hence the adversary is unable to trace the tag. □

### 5.2 Informal security analysis

#### 5.2.1 Replay attack resistance

An adversary can eavesdrop the wireless channel and keeps all the transmitted messages between a reader and a tag. The adversary uses these message into another session to disguise itself as the tag or the reader to deceive the other one. In the proposed scheme, it is infeasible for an adversary to forge messages as a valid tag/reader because each transmitted message incorporates a fresh nonce in each authentication session which can not get by the adversary (since the nonce XOR with some other secret parameters). This makes all the replayed message by the adversary are illegal message. Thus the scheme prevents strongly the replay attack.

#### 5.2.2 De-synchronization attack resistance

For each tag $T_{ji}$, the server stores two nonces $r_{\text{old}}$ and $r_{\text{new}}$ in its database to save the scheme from the de-synchronization attack. The server also updates these values after a successful authentication session. An adversary intercepts
or modifies any transmitted message in one session in such a way so that a tag does not update the value of the stored nonce. The server can authenticate the legitimate tag by its old value stored in the database into another session. So it is not possible for an adversary to de-synchronize the scheme.

5.2.3 Man-in-middle attack resistance

An adversary is unable to act as a middle-man between a legitimate reader and a legitimate tag because it is infeasible for the adversary to intercept any transmitted message without knowing the key, unique identification number, and knowledge about the cyclic group. The probability of guessing or calculating these values from the transmitted message is negligible because a fresh nonce is used in each transmitted message.

6 Performance analysis

In this section, we present the efficiency of our proposed scheme in terms of tag computation, reader computation, and storage, as described in Table 3. The proposed scheme’s search complexity is $O(\gamma)$ which is the same as in Avoine et al. [1] but better than Rahman et al. [18]. During the authentication phase, the scheme performs only bitwise logical operations such as XOR and circular left shift whereas Avoine et al. and Rahman et al. perform symmetric key encryption and decryption. Also, the proposed scheme does not use any pseudo-random number generator function for generating nonce on the tag side. We assume that all the parameters used in the proposed scheme are $L$-bits long. On the tag side, our scheme keeps five items. Thus the storage cost is $5L$ bits. The proposed scheme also provides mutual authentication between the reader and tags. When we compare with [1, 18] in terms of computation, the proposed scheme performs very less computation which is optimal for the real world tiny powered tags.

7 Measurement of privacy

In this section, we analyze the privacy level of our proposed scheme in terms of anonymity set and data leakage in bits. For the anonymity sets, we use privacy metric introduced by [3]. Also, we use another metric says information leakage proposed by Shannon [19] to measure the information (in bits) disclosed by the proposed scheme when some tags are compromised.

Both the metric use disjoint partition sets of tags for observation. When some tags are compromised, the set of all tags is partitioned in such a way so that the adversary can not distinguish the tags that belong to the same partition but the adversary can distinguish the tags belong to different partitions. Here, $|P_i|$ denotes the size of such partition $P_i$ and $\frac{|P_i|}{N}$ is the probability that a randomly chosen tag belongs to partition $P_i$.

7.1 Level of privacy based on anonymity set

The level of privacy $\mathcal{R}$ based on anonymity set is characterized as average anonymity set size normalized with the total number of tags $N$ [1, 3, 18].

$$\mathcal{R} = \frac{1}{N} \sum_i |P_i| \frac{|P_i|}{N} = \frac{1}{N^2} \sum_i |P_i|^2.$$  \hspace{1cm} (1)

In the proposed scheme, if a tag is compromised, it does not leak any information about the subgroup in which it belongs. For this reason, the adversary can not distinguish between two tags whether they belong to the same subgroup or not. So, if $C$ is the total number of compromised tags in the system, we partitioned the system into $C$ number of anonymity sets with size 1 and one another anonymity set of size $(N - C)$. Using Eq. 1, the level of privacy $\mathcal{R}$ achieved by our scheme is

$$\mathcal{R} = \frac{1}{N^2} \{ C + (N - C)^2 \},$$ \hspace{1cm} (2)

where $N$ is the total number of tags in the system and $C$ is the total number of compromised tags in the system.

| Protocol                     | Entity | Avoine [1] | Rahman [18] | Proposed protocol |
|------------------------------|--------|------------|-------------|------------------|
| Symmetric encryption/decryption | T      | 2          | 2           | ×                |
| R                            | 2      | 2          | 2           | ×                |
| Search complexity            | R      | $O(\gamma)$| $O(\gamma + |\pi|)$| $O(\gamma)$        |
| No. of PRNG                  | T      | 1          | 1           | ×                |
| Required memory              | T      | $3L$       | $(m + 2)L$  | $5L$             |
| Mutual authentication        | ×      | ×          | ✓           |                  |

T: tag-side; R: reader-side; $\gamma$: total number of groups in the system; $|\pi|$: total number of secret keys of a tag associated with the identifier $ID_x$; $m$, number of identifier is assigned to each tag.

© Springer
Table 4 The system setup with $N = 2^{10}$ tags that are divided into 32 distinct subgroups of a finite cyclic group $G$

| Subgroup | Order | Total tags | Subgroup | Order | Total tags | Subgroup | Order | Total tags |
|----------|-------|------------|----------|-------|------------|----------|-------|------------|
| $H_1$    | 18    | 17         | $H_{12}$ | 29    | 28         | $H_{23}$ | 40    | 39         |
| $H_2$    | 19    | 18         | $H_{13}$ | 30    | 29         | $H_{24}$ | 41    | 40         |
| $H_3$    | 20    | 19         | $H_{14}$ | 31    | 30         | $H_{25}$ | 42    | 40         |
| $H_4$    | 21    | 20         | $H_{15}$ | 32    | 31         | $H_{26}$ | 43    | 42         |
| $H_5$    | 22    | 21         | $H_{16}$ | 33    | 32         | $H_{27}$ | 44    | 43         |
| $H_6$    | 23    | 22         | $H_{17}$ | 34    | 33         | $H_{28}$ | 45    | 43         |
| $H_7$    | 24    | 23         | $H_{18}$ | 35    | 34         | $H_{29}$ | 46    | 43         |
| $H_8$    | 25    | 24         | $H_{19}$ | 36    | 35         | $H_{30}$ | 47    | 43         |
| $H_9$    | 26    | 25         | $H_{20}$ | 37    | 36         | $H_{31}$ | 48    | 43         |
| $H_{10}$ | 27    | 26         | $H_{21}$ | 38    | 37         | $H_{32}$ | 49    | 43         |
| $H_{11}$ | 28    | 27         | $H_{22}$ | 39    | 38         |          |       |            |

*Total tags* the total number of tags that are assigned in a subgroup in the system

**Fig. 2** Level of privacy of the system based on anonymity set

**Fig. 3** Level of privacy of the system based on information leakage in bits
7.2 Level of privacy based on information leakage in bits

According to Rahman et al. [18], if an adversary partitioned a system with \( N \) tags into \( k \) disjoint sets, then the information leakage in bits can be expressed as follows:

\[
\ell = \sum_{i=1}^{k} \frac{|P_i|}{N} \log_2 \left( \frac{N}{|P_i|} \right),
\]

(3)

In the proposed scheme, if \( C \) is the total number of compromised tags in the system. Then we partitioned the system with \( N \) tags into \( C \) anonymity sets of size 1 and one another anonymity set of size \( (N - C) \). According to our partitions, the information leakage in bits is as follows

\[
\ell = \frac{C}{N} \log_2 N + \frac{(N - C)}{N} \log_2 \left( \frac{N}{N - C} \right),
\]

(4)

8 Experimental results

In this section, we compare our scheme with Avoine et al. [1] and Rahman et al. [18] by using a matlab simulation. The simulation is done using the expressions (1)–(4). In the simulation, we assume that the system has \( N = 2^{10} \) number of tags and all the tags are divided into 32 clusters. For the system setup as mentioned in Sect. 2, we choose an additive cyclic group \( G = \mathbb{Z}/n\mathbb{Z} \), where we choose \( n = 309904504245996706400 \). From the group \( G \), we choose its 32 distinct subgroups that are denoted by \( H_i \), \( i = 1, 2, \ldots, 32 \). We divide the set of all the tags \( N \) in the system into these chosen 32 distinct subgroups of the group \( G \). The maximum number of tags that we can assign in a subgroup \( H_i \) is \( |H_i| - 1 \) as mentioned in Sect. 2. Table 4 shows that the set of all subgroups with their order that is used in this simulation. In addition, the total number of tags that are assigned in each subgroup is also shown in Table 4.

We choose a range of compromised tags from 0 to 600. In the simulation, we run 100 simulations for each value of compromised tags \( C \) in the system. In each simulation run, compromised tags are chosen uniformly random from the subgroups of all tags. Finally, we average all the obtained values overall simulation runs. The simulation results are shown in Figs. 2 and 3. The simulation results of the Fig. 2 shows that the privacy level achieved by the proposed scheme is 94.42% and 98.43% better than Rahman et al. and Avoine et al. respectively, when \( C \) becomes 600 in a similar setup. According to simulation result shown in Fig. 3, the proposed scheme discloses 22.62% and 34.05% less information than Rahman et al. and Avoine et al. respectively when \( C \) becomes 600. Thus the proposed scheme achieves higher improvement in terms of privacy level and data integrity than the other schemes when some tags are compromised by an adversary.

9 Conclusion

In this paper, we have proposed a group based authentication scheme for RFID system based on a cyclic group. The detailed formal analysis shows that it preserves information privacy and untraceability. The informal analysis shows that the scheme resists various existing attacks. The performance analysis illustrates that the scheme uses very fewer resources on tags to perform computational work and storage data. The experimental results show that our scheme preserves high-level privacy and discloses less information when some tags are compromised. Thus, the analysis and prominent features conclude that the scheme is secure and efficient for a low-cost RFID system.

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

Compliance with ethical standards

Conflict of interest The author P. K. Maurya thanks to MHRD, India for financial support of his research work.

References

1. Avoine, G., Buttyan, L., Holczer, T., & Vajda, I. (2007). Group-based private authentication. In IEEE international symposium on a world of wireless, mobile and multimedia networks (pp. 1–6).
2. Avoine, G., Coisel, I., & Martin, T. (2014). Untraceability model for RFID. IEEE Transactions on Mobile Computing, 13(10), 2397–2405.
3. Buttyan, L., Holczer, T., & Vajda, I. (2006). Optimal key-trees for tree-based private authentication. Berlin: Springer.
4. Cao, T., Chen, X., Doss, R., Zhai, J., Wise, L. J., & Zhao, Q. (2016). RFID ownership transfer protocol based on cloud. Computer Networks, 105, 47–59.
5. Chien, H. Y. (2006). Secure access control schemes for RFID systems with anonymity. In 7th international conference on mobile data management (MDM ’06) (pp. 1–4).
6. Chien, H. Y. (2007). SASL: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE Transactions on Dependable and Secure Computing, 4(4), 337–340.
7. Cho, J. S., Jeong, Y. S., & Park, S. O. (2015). Consideration on the brute-force attack cost and retrieval cost: A hash-based radio-frequency identification (RFID) tag mutual authentication protocol. Computers and Mathematics with Applications, 69(1), 58–65.
8. Dimitriou, T. (2016). Key evolving RFID systems: Forward/backward privacy and ownership transfer of RFID tags. Ad Hoc Networks, 57(Part 2), 195–208.
9. Gao, L., Ma, M., Shu, Y., & Wei, Y. (2014). An ultralightweight RFID authentication protocol with CRC and permutation. Journal of Network and Computer Applications, 41, 37–46.
10. Gallian, J. A. (2016). Contemporary abstract algebra. Boston: Cengage Learning.
11. Juels, A., & Weis, S. A. (2009). Defining strong privacy for RFID. ACM Transactions on Information and System Security, 13(1), 7:1–7:23.
12. Kaul, S. D., & Awasthi, A. K. (2017). Privacy model for threshold RFID system based on PUF. Wireless Personal Communications, 95(3), 2803–2828.
13. 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, CCS ’04 (pp. 210–219). ACM.
14. Nohara, Y., Inoue, S., Baba, K., & Yasuura, H. (2005). Quantitative evaluation of unifiable id matching schemes. In Proceedings of the 2005 ACM workshop on privacy in the electronic society, WPES ’05 (pp. 55–60). ACM.
15. Nohl, K., & Evans, D. (2006). Quantifying information leakage in tree-based hash protocols information and communications security. In 8th international conference, ICICS (pp. 228–237).
16. Maurya, P. K., Pal, J., & Bagchi, S. (2017). A coding theory based ultralightweight RFID authentication protocol with CRC. Wireless Personal Communications, 97(1), 967–976.
17. Maurya, P. K., & Bagchi, S. (2018). A secure PUF-based unilateral authentication scheme for RFID system. Wireless Personal Communications. https://doi.org/10.1007/s11277-018-5875-2.
18. Rahman, F., Hoque, M. E., & Ahamed, S. I. (2017). Anonpri: A secure anonymous private authentication protocol for RFID systems. Information Sciences, 379, 195–210.
19. Shannon, C. E. (2001). A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review, 5(1), 3–55.
20. Srivastava, K., Awasthi, A. K., Kaul, S. D., & Mittal, R. C. (2014). A hash based mutual RFID tag authentication protocol in telecare medicine information system. Journal of Medical Systems, 39(1), 1–5.
21. Tian, Y., Chen, G., & Li, J. (2012). A new ultralightweight RFID authentication protocol with permutation. IEEE Communications Letters, 16(5), 702–705.
22. Wang, J., Floerkemeier, C., & Sarma, S. E. (2014). Session-based security enhancement of RFID systems for emerging open-loop applications. Personal and Ubiquitous Computing, 18(8), 1881–1891.
23. Weis, S., Sarma, S., Rivest, R., & Engels, D. (2004). Security and privacy aspects of low-cost radio frequency identification systems. Security in Pervasive Computing, 2802, 201–212.

Pramod Kumar Maurya was born in 1990. He received his M.Sc. in Mathematics from the University of Allahabad, India, 2011. He has completed M.Tech. in Computer Science and Data Processing from IIT Kharagpur, India, 2014. He is currently a research scholar in the Department of Mathematics at NIT Durgapur, India. His research interests include identity authentication, RFID security, and information security.

Satya Bagchi was born in 1982. He completed the B.Sc. and M.Sc. Degrees in Mathematics from the University of Kalyani, West Bengal, India, in 2002 and 2004, respectively. He received the Ph.D. degree in Mathematics from the same university in 2013. From 2006 to 2007 he was a lecturer, Department of Mathematics, A B N Seal College, Cooch Behar, West Bengal, India. Currently, he is an Assistant Professor, Department of Mathematics, National Institute of Technology, Durgapur, India. He has many Ph.D. and M.Sc. students working together as a team. His major current research interests are in RFID security protocol design, cryptography, and coding theory. Dr. Bagchi is a life member of the Cryptology Research Society of India (CRSI).