OBJECT IDENTIFICATION IN RFID TECHNOLOGY

Subhasish Dhal

OBJECT IDENTIFICATION IN RFID TECHNOLOGY

Thesis submitted to the
Indian Institute of Technology, Kharagpur
for award of the degree

of

Doctor of Philosophy

by

Subhasish Dhal

under the supervision of

Prof. Indranil Sen Gupta

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

SEPTEMBER 2014

APPROVAL OF THE VIVA-VOCE BOARD

Date: / / 20

Certified that the thesis entitled **"OBJECT IDENTIFICATION IN RFID TECHNOLOGY"** submitted by SUBHASISH DHAL to the Indian Institute of Technology, Kharagpur, for the award of the degree of Doctor of Philosophy has been accepted by the external examiners and that the student has successfully defended the thesis in the viva-voce examination held today.

(Member of DSC) (Member of DSC) (Member of DSC)

(Supervisor 1)

(External Examiner) (Chairman)

CERTIFICATE

This is to certify that the thesis entitled **"OBJECT IDENTIFICATION IN RFID TECHNOLOGY"**, *submitted by* **SUBHASISH DHAL** *to the Indian Institute of Technology, Kharagpur, for the award of the degree of Doctor of Philosophy, is a record of bona fide research work carried out by him under my supervision and guidance. The thesis, in my opinion, is worthy of consideration for the award of the degree of Doctor of Philosophy in accordance with the regulations of the Institute.*

Indranil Sen Gupta,
Professor,
Department of Computer Science and Engineering
IIT Kharagpur

Date:

DECLARATION

I certify that

a. The work contained in this thesis is original and has been done by myself under the general supervision of my supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in writing the thesis.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

f. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

SUBHASISH DHAL

ACKNOWLEDGMENTS

ABSTRACT

Radio Frequency Identification (RFID) technology is a powerful tool for identification of any object uniquely. Bar code or similar kind of technologies are also can be used for identification of the objects. However, the accuracy of object identification is far better in RFID technology since it does not have any line-of-sight constraint. The popularity of this technology has also been boosted due to its low cost. However, this technology becomes a cost effective solution by compromising the resources (hardware). Therefore, implementation of any application in this technology is a challenging task. This technology is one of the most pervasive computing technology and vulnerable to various kinds of attacks such as eavesdropping, replay attack, etc. However, we cannot use any standard cryptographic primitive in this technology in order to provide security against these attacks. Proper lightweight cryptography primitives need to be used without compromising the security.

Due to pervasive property of RFID technology, authentication is an important requirement in various RFID applications to restrict the non-legitimate access to certain resources. Many lightweight authentication schemes have been proposed till date. In the present thesis, we have tried to solve the location privacy problem that is present in many existing authentication schemes. Our solution can be applied over any existing authentication scheme to overcome the location privacy problem. However, this solution requires a small amount of additional hardware (288 gates) in RFID tag for its implementation. We also identified the vulnerability of de-synchronization attack in the authentication scheme proposed by Khor et al. (2010) and suggested a solution to fix this problem.

The detection probability of an object is low when it is attached with a single RFID tag. This can be improved by attaching multiple number of tags in a manner such that if

any part of the object is within the coverage area of the reader, at least one tag attached to the object will be visible. The existing authentication schemes suffer from low object detection probability since they assume that the objects are attached with single tag. Theses schemes cannot be extended to multi-tag environment, since they use one set of security related information for an object and keeping the same information in multiple tags attached to the same object is vulnerable. The adversary can easily compromise all the tags by compromising only a single tag. Therefore, an authentication scheme needs to be designed in multi-tag environment which can increase the difficulty for the adversary without compromising the detection probability of the object. In the present thesis, we propose two lightweight and secure authentication schemes in multi-tag environment. In the first authentication schemes, all the tags are allowed to response on a request from a RFID reader. This increases a traffic congestion in the communication medium between the object and the RFID reader. In order to overcome this problem, we propose another authentication scheme. In this scheme, an object is attached with multiple number of active tags among which one of them performs the authentication task. If this tag is not present within the coverage area of the reader, it obtains information through the other tags attached to the same object. Hence, the detection probability of an object does not decrease. Though the proposed scheme uses multiple tags, the traffic congestion does not increase as a result of single response from each object.

Sometime, an object needs to be found from a large set. Any authentication scheme can be extended for this purpose. However, this approach is inefficient since $\frac{n}{2}$ number objects in average need to be authenticated in order to search an object. Similar to the authentication schemes, existing object searching schemes which assume the objects are attached with single tag cannot be extended to multi-tag environment. In the present thesis, we propose a secure and lightweight object searching scheme in multi-tag environment.

Some applications require the coexistence of a group of two or more relevant objects which can help to perform a particular event. Absence of one or more objects in the group can produce a wrong outcome. An assurance in the form of a proof of coexistence of the desired objects can help to execute the event without any error. This problem can be solved using RFID technology. The existing proof generation schemes assume that the objects are attached with single tag and these schemes cannot be extended to multi-tag environment. In this thesis, we initially propose a secure and lightweight proof generation and validation scheme in multi-tag environment for a group of two objects. Later we extend the protocol to adopt a group of arbitrary number of objects.

Various kinds of analyses have been conducted in the thesis in order to evaluate the applicability of the proposed schemes. A thorough analysis has been performed as a part of this work to find the possible attacks that can occur during the execution of the proposed protocols. Formal and informal analyses have been carried out to verify the robustness of the proposed schemes. This shows that the proposed protocols can satisfy most of the security requirements. Since the RFID technology has various resource constraints, the thesis also analyzes and compares the resource requirements of the proposed protocols in terms of computation, communication and storage.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Radio Frequency Identification (RFID) technology is a powerful tool for identification of any object uniquely. In this technology, the reader scatters electromagnetic signal and a tag attached to an object responds with the information kept in its memory. The RFID reader then uses this information according to the need of any application. Sometime, the information about an object is so large that the tag memory is not sufficient to keep the whole information. In those situations, the information about all objects are kept in a workstation called backend server. A unique identifier (id) is kept in the tag memory to identify the tag and hence identify the corresponding object. The other information about this object are kept in the backend server. The RFID reader reads the tag id to identify the object and obtains the detail information from the backend server. There are many applications which use this technology for identification process. For example, the books in the library are attached with RFID tags and each tag contains unique id such as accession number while the other information about the books are kept in the backend server. The librarian issues a book by using a RFID reader. The reader scatters electromagnetic signal and the tags attached to the books to be issued backscatter (reflect) their unique identifiers. The reader collects these identifiers and the corresponding records in the backend server are updated. Similarly, when a book is returned, the identifier of the returned book is collected and the corresponding record is updated in the backend server. In this technology, the reader can identify many books simultaneously and most of the library services become automated. Therefore, the RFID technology helps the library system to be efficient. The bar code technology or Quick Response (QR) code technology can serve this purpose. However, these technologies have line of sight constraint and more than one books cannot be read simultaneously. On the other hand, the RFID technology does not have any such constraint since the radiation of electromagnetic signal by any RFID reader is omni-directional. Hence the bar code and QR code technologies are inefficient compared to the RFID technology. The another problem with bar code and QR-code

is that the printed code in an object may be erased after some years and hence the corresponding reader may be unable to read any information from it. However, the RFID tag is usually attached in such a way that the durability of the chip is much higher than either bar code or QR code. However, though this technology has many benefits, many security and privacy issues are of great concern due to its pervasiveness property. Therefore, any application using RFID technology needs to be secured from the possible security and privacy vulnerabilities.

1.1 Motivation

The detection probability of an object is less when it is attached with single RFID tag [**?**]. According to the findings in [**?**], the detection probability of the object can be increased by attaching multiple number of tags to it. In addition to this, it increases the reader-tag communication distance in the presence of metals, liquids, radio noise and adverse environmental conditions. Thus, it is highly applicable to those applications where reliability, availability, and safety are major requirements [**?**].

Above all these, the security is a major concern as we have mentioned earlier. RFID tags are easily accessible and hence the sensitive information within the tags are no longer secure. The security requirements such as privacy, authentication and integrity etc. are thus stringent requirements in this technology.

Due to the resource constraints of RFID devices, the implementation of any RFID based application is a very challenging task. Therefore, the challenge is to devise any application which will be robust in terms of security as well as efficient. We have chosen three areas, namely, object authentication, object searching and coexistence proof generation.

1.1.1 Object authentication

Pervasiveness property of the RFID technology makes the RFID tags freely available to the non legitimate users. Hence, it can be misused in many forms. For example, the items in a shopping mall can be attached with RFID tags and be identified with the help of a RFID reader. The tags are assigned unique identifiers. Suppose an unauthorized person enters into the shopping mall and wants to buy a costly item by paying lesser cost than the original cost of the item. To achieve his objective, he can use a fake RFID reader and read the identifier of the tag attached to a less costly item. He then accesses the tag attached to the intended costly item and replace the identifier of the tag attached to the costly item with the identifier of the tag attached to the less costly item. The buyer also can replace the original RFID tag attached to the intended

item with a fake RFID tag having an identifier of the tag attached to a less costly item. When he will go for paying the cost of the item, the accountant will use a RFID reader and read the identifier of the tag attached to the costly item. The reader will wrongly read the identifier of the lesser costly item and the backend server will relate with the record kept for the lesser costly item. Therefore, the retrieved cost will be less than the original cost of the item. However, the accountant cannot understand this fraud purchase since the cost determination process is automatic. Therefore, the buyer can be successful to purchase a costly item by paying lesser cost than the original cost of the item. Similarly, there are many applications where this kind of fraud operation can be performed silently. This fraud operation is allowed since the RFID tag does not verify the validity of the entity which is accessing it and it blindly performs any operation according to the instruction of a non-legitimate entity. Similarly, the RFID reader also does not verify the response from a fake RFID tag and it blindly retrieves the information kept in the database of the backend server. Therefore, any entity involved in an application must perform any operation on a request from the other entity after verifying the legitimacy of the request. Thus the fraud operations can be prevented. Hence, authentication is a major security requirement in the RFID technology. However, during the authentication process, the adversaries can access the communication channel and perform various kind of attacks in order to disrupt the authentication process and gain valuable information.

There are many security threats that can be performed by the adversaries during the authentication process. Following are the possible attacks which can be mounted during the authentication process.

- **Passive attacks:** The adversary $\mathcal{A}$ silently extracts secret information about the legitimate objects.
  - *Eavesdropping:* $\mathcal{A}$ silently listens to the communication and tries to extract the secret information such as identifier, session key, secret key, etc.
  - *Location privacy:* $\mathcal{A}$ tries to find out a pattern from the requests and responses, and tries to trace the object.
  - *Location privacy between two successful sessions:* Between two consecutive successful sessions, $\mathcal{A}$ can try to trace an object.
- **Active attacks:** The adversary not only listens to the vital information but also tries to disrupt the authentication process. Any adversary may mount the following active attacks:
  - *Man-in-the-middle attack:* $\mathcal{A}$ may modify the information communicated through insecure medium and thus can disrupt the authentication process.

- *Replay attack:* The authentication information of a legitimate session may be saved and replayed for successful validation in later sessions.
- *Forward secrecy and Backward secrecy:* Compromising the secret information used in one valid session, $\mathcal{A}$ may try to obtain the secret information to be used in later or previous sessions.
- *De-synchronization attack:* In some situations, the information such as identifier, session key etc. for an object are updated and then communicated from either reader to object or object to reader in each successful session. However, if an adversary blocks the updated information then there can be a synchronization problem between backend server and the object.
- *Impersonation attack:* $\mathcal{A}$ may clone a legitimate tag and use the cloned tag to impersonate the legitimate tag.

Many authentication protocols based on the RFID technology exist in the literature. These protocols can be classified into various families. There are many hash operation based authentication schemes. One simple authentication scheme in this family is the protocol proposed by Wies et al. [**?**]. This protocol though simple, suffers from location privacy problem and many other vulnerabilities. The authors have suggested a modification to this scheme and proposed Randomized Access Control (RAC) to mitigate the location privacy problem. However, the modified scheme also suffers from the impersonation attack and many other vulnerabilities. The other protocols in this family are described in [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**]. Since these protocols use hash operation, the power consumption in tag is high. HB family has started by Hopper and Blum in [**?**] and this family has extended in [**?**] [**?**] [**?**] [**?**]. The protocols in this family use multiplication operation and add some noise on the authentication information. The security of these protocols is based on the hardness of Learning Parity with Noise (LPN) problem. However, the applied noise may induce ambiguity. There is another protocol proposed in [**?**] which uses multiplication operation. This protocol suffers from some attacks reported in [**?**]. A set of protocols exist in the literature which uses substring function and this family is started by Li et al. in [**?**] and this family is extended in [**?**] [**?**] [**?**]. The protocols in this family also suffer from the ambiguity problem. Because there can be common substring in the responses from more than one tag. There are few protocols which suggest the use of one or more physical properties of the RFID tag and these properties can be converted into a unique fingerprint. This fingerprint can be used as the identity of the tag. It can help to prevent the adversaries to clone the RFID tags and consequently can help to prevent the impersonation of a tag. The protocols proposed in [**?**] [**?**] are such protocols which use fingerprint as the identifier. However, the power consumption in these protocols is high due to the fingerprint generation operation. There

is another family of protocols which use Pseudo Random Number Generation (PRNG) function. This function is used mainly to update the secure information such as session key, identifier etc. The protocols in [**?**] [**?**] [**?**] [**?**] are the members of this family. Though the authors in these protocols claim that the protocols provide the forward secrecy requirement, the adversary can easily compute the future secrets using the compromised secrets and the PRNG function. Elliptic curve cryptography based protocols are another family of protocols which includes the protocols in [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**]. The security of most of these protocols is based on the hardness of Elliptic curve discrete logarithmic problem (ECDLP). However, use of one or more ECC multiplication increases the power consumption in RFID tags. There is another kind of protocol [**?**] in the literature which authenticates more than one objects simultaneously. This protocol saves the computation by reusing the response of one tag as the random number for the other tag. However, this protocol carries unnecessary computations in the event of an attack. Because any attack is detected at the last stage of the authentication process.

Many authentication protocols claim that their protocols are secure from the tracking attack. These protocols suggest the modification of the secure information such as session key, identifier etc. in each successful sessions in order to prevent the tracking attack. However, the secure information are not modified during the time between two successful sessions. Hence, the adversary can receive same response from a tag multiple time during this period and track the tag. The authentication protocol such as [**?**] [**?**] [**?**] suffer from this attack. Hence there is a need of a remedy to this problem which can fix this vulnerability in the suffered existing protocols. In addition to this attack, the protocol proposed in [**?**] suffers from the de-synchronization attack. Hence the corresponding vulnerability needs to be fixed.

All the protocols exist in the literature suffer from one or more security implications and these schemes consider that an object is attached with single RFID tag. Hence the detection probability of the object in their schemes is less. To increase the detection probability, an authentication scheme should incorporate multiple number of tags in the same object [**?**], where it is sufficient that at least one tag must be visible to the reader if any part of the same object is within the communication range of the reader.

The existing authentication schemes can be modified to incorporate multiple number of tags in each object. However, we have to keep one set of security related information for an object to all the tags attached to the object since the existing schemes use only one set of security related information for the object. If an adversary somehow compromise a tag, he can compromise the other tags attached to the same object. Therefore, the effort requires for the adversary to compromise an object will be less. This raises the need of separate authentication protocols which can use multiple number tags to increase the difficulty for the adversary to mount any

attack.

1.1.2 Searching a desired object using RFID technology

Object searching is the process of identifying a particular object from a pool of objects. For example, the librarian in a library equipped with RFID technology searches a book using RFID reader before issuing the book. The reader issues a search request which includes the identification information of the tag attached to the desired book. The tag attached to the desired book responds and the reader indicates the existence information of the book.

The searching process can be achieved using an authentication protocol. In the authentication process, the reader issues a request message and the authentication process continues until the desired tag responds or all the tags in the coverage area of the reader have been processed. However, this process requires to verify the authentication information of $\frac{n}{2}$ number of tags on the average, where n is the number of tags within the coverage area of the reader. On the other hand, if the search process requires only the desired tag to respond then the reader needs to verify only the response from the desired tag. Therefore, this kind of searching process is efficient. Adversaries can utilize the insecure medium to implement a number of attacks during the searching process similar to the attacks during the execution of an authentication protocol. In addition to this, the adversary can mount Information leakage attack. This is a passive kind of attack. According to the searching mechanism only the desired tag responds to a search query. Though this response is encrypted and secured from various kind of active or passive attacks, the adversary can know the existence information of the desired tag. Therefore, some information are leaking to the adversary in spite of the highly secured encryption process. Suppose the desired tag is attached to a precious jewelry. The response from the tag attached to the precious jewelry can reveal the existence information. Therefore, the adversary can easily find the jewelry item.

The object searching using RFID technology has not been focused adequately in the literature. Tan et al. [**?**] proposed four object searching schemes in 2008. These schemes are vulnerable to many threats including the tracking attack, ID disclosure attack etc. Ahmed et al. [**?**] suggest three protocols in the same year without using any backend server. First two protocols among these protocols are actually the extensions of an authentication protocol and therefore efficiency of these protocols are less. They modified these protocols where the desired tag verifies the search request and responds while the undesired tag responds with fake information. These schemes are also vulnerable to the attacks like De-synchronization attack, etc. Kulseng et al. [**?**] suggest Physically Unclonable Function (PUF) operations in order to prevent cloning attempt by the adversaries. They also suggest the use of Linear Feedback Shift Register (LFSR) to update

various security information. They proposed three protocols in 2009. These protocol suffer from various attacks such as Information leakage attack, De-synchronization attack, etc. Moreover, due to the use of PUF and LFSR operation, the computation requirement in tag is high. Hoque et al. [**?**] proposed a hash operation based protocol in 2010, namely, S-search which satisfies most of the security requirements. However, due to the use of many hash operations, the computation requirement in this scheme is high. Another hash operation based protocol is proposed by Yoon et al. in 2011. They use a counter value to prevent the replay attack. However, this scheme is also vulnerable to the information leakage attack. Lee et al. [**?**] suggest the use of Elliptic curve cryptography (ECC) based protocol. The hardness of the elliptic curve digital signature algorithm (ECDSA) is the basis of the security of this protocol. This scheme also suffers from the information leakage attack and forward secrecy problem. Moreover, due to the use of ECC multiplication operation, the computation is high in RFID tag. Zheng et al. [**?**] use an approximation technique using bloom filter to identify a set of objects in 2013. They proposed the two-phase Compact Approximator based Tag Searching protocol (CATS) in 2013. However, due to the false positive property of the bloom filter, a few undesired tags may be selected and according to Min et al. [**?**], CATS does not work when the size of the wanted tag set is much higher than the number of tags in the coverage area of the reader. This scheme also inefficient when the false positive ration is high. Min et al. [**?**] proposed an iterative tag searching protocol (ITSP) which is basically a modification of CATS. Though this scheme is efficient still it suffers from the false positive property of the bloom filter. Moreover, the number of iteration is equal to the number of hash functions and hence the number of interactions between the tags and the reader is high.

The existing object searching schemes using RFID technology concentrate on various security issues and they try to optimize the resources. Therefore, they try to maintain a balance between the security and resources. However, all the schemes are vulnerable to one or more attacks and a few schemes use heavyweight operations which may not be applicable to low cost RFID tags. These protocols assume that the objects are attached with single tag and hence the coverage area of the objects is less as we have mentioned earlier. Attachment of multiple number of tags can increase the coverage area of an object and these multiple tags can be utilized to enhance the security. Many existing object searching protocols avoid the information leakage attack by permitting the undesired tags within the coverage area of the reader to respond with fake information. However, the reader and/or backend server has to filtered these fake responses. This filtration process introduces useless computations in the reader and/or backend server. This useless computation needs to be minimized. Hence there is a need of a lightweight and secure object searching protocol in multi-tag environment which can prevent the information leakage attack with minimized useless computations due to the fake responses from the undesired tags.

1.1.3 Coexistence proof generation and validation for multiple relevant objects

The third area we have explored is the generation of a proof of coexistence of multiple number of relevant objects. There are many applications which requires a proof of coexistence of multiple number of relevant objects. For example, nurses in a hospital are usually assigned to the patients for necessary medications. They are responsible to take care of the medicines, check up etc. However, a careless nurse can do a mess in her duty. She can put an injection to a wrong patient which can cause a severe damage to that patient and consequently this can cause the patient dead. Therefore, there is a need of an automatic process by which the nurse can be alerted when she is preparing for the medication. The system will verify whether the correct nurse is providing a correct medication to a correct patient. In order to perform this verification, the system will generate a coexistence proof of correct medication, correct nurse and correct patient. It will then verify this proof and if it finds that the proof is correct, then the nurse will be allowed to provide the medication. Otherwise, the nurse will be alerted. This proof also can be kept to show the correctness of the medication process.

The proof generation process can be automated using the RFID technology. For example, the proof generation process for the medication process in hospital can be automated using RFID technology. In this technology, the patient is attached with a RFID tag A, the medication box is attached with a RFID tag B and the assigned nurse is attached with another tag C. During the medication process, a RFID reader will identify the tags and it will generate a valid proof if all A, B, C are present simultaneously. A verifier will then verify this proof and instruct the nurse accordingly. The verifier will generate a warning if it finds that the proof is not valid. Therefore, the nurse will be alerted in case of a wrong attempt of medication by the nurse. This proof can be saved for future verification process if necessary.

During the proof generation process, there can be many attacks from a non legitimate entity. The attacks consists of those attacks which can happen during the authentication process (We have discussed earlier). In addition to this, there are some other attacks as follows:

- **Denial of proof attack:** Adversary $\mathcal{A}$ may put a non-legitimate tag and if the response from this tag is added into the proof, it will be invalid. Thus in the presence of all the relevant objects the generated proof is invalid.

- **Relay attack:** Suppose one or more objects in a given schedule are not within the coexisting coverage range. The adversary $\mathcal{A}$ will put one or more intermediate transceivers and relay the search request to the desired object. Thus the proof generation process will generate a valid proof although the desired objects are not present within the coexisting range.

There are a few protocols which address the coexistence proof generation problem. In 2004, Juels had formulated the problem of generating the coexistence proof, namely yoking proof [**?**]. Implementation of this scheme is very simple. However, Saito and Sakurai [**?**] show that this scheme is vulnerable to replay attack where the attacker can generate a valid proof in absence of any tag among the two relevant tags. Saito and Sakurai [**?**] proposed another yoking proof protocol in 2005 in order to adopt the prevention mechanism against the replay attack. They use a time stamp information in their scheme which perhaps can prevent the attacker to reuse the stored information. However, Piramuthu shows in [**?**] that a predicted time stamp information can break the security against the replay attack. Hence Piramuthu had modified this scheme in 2006 and included random nonce to ensure that a valid proof cannot be generated in absence of any of the desired tag. However, this modification faces many other attacks such as the tracking attack, attack against forward secrecy, denial of proof attack, relay attack etc. Bolotonny and Rubin [**?**] have proposed a separate anonymous proof generation protocol for more than two tags in 2006. In this protocol, the timer operation is assumed as the discharging rate of the underlying capacitor. However, this capacitor may be recharged through other sources of radio signals which can introduce noise in timer operation. This scheme also vulnerable to the attacks such as denial of proof attack, relay attack. etc. Peris-Lopez et al. proposed a coexistence proof generation protocol [**?**], namely, clumping proof in 2007. They use an encrypted time stamp to avoid the problems in [**?**] [**?**] [**?**]. However, this protocol also vulnerable to the attacks such as de-synchronization attack, denial of proof attack, etc. In the same year, Chih-Chung Lin et al. proposed two protocols, namely, *sec-TS-proof* and *chaining proof*. The first protocol is actually the modification version of the protocol proposed in [**?**]. It can prevent the relay attack. The second protocol assumes that there is no online server during the proof generation process and use a times tamp database (a module of of the RFID reader) as a time stamp server. This protocol is also vulnerable to the attacks similar to *sec-TS-proof*. In 2009, Burmester et al, proposed a protocol [**?**] which is claimed to be an anonymous and provide forward secrecy. They suggest the use of the channel in the data link layer to link the relevant tags in order to prevent the denial of proof attack. However, in this protocol, the objects can be traced by an adversary. Yao et al. [**?**] proposed another scheme in 2010. Though this protocol uses light weight operations, it is vulnerable to the attacks such as De-synchronization attack, relay attack, denial of proof attack, etc. In the same year, Duc and Kim [**?**] have proposed a proof generation protocol based on secret sharing mechanism. This scheme is very simple. However, traceability and the relay attack can be mounted in this scheme. In 2010, there were another paper [**?**] where Nai-Wei Lo et al. proposed three protocols, namely, online verifier based protocol (OVBP), efficient online verifier based proocol (EOVBP) and offline timestamp server baed protocol (OTSBP). All the

protocols proposed in this paper uses three heavyweight operations (PRNG, MAC and Nun) and this makes the protocols inefficient. Moreover, if the adversary put many invalid tags, the proof generation process will be delayed for a long time.

The existing proof generation protocols address various security issues during the proof generation process and these protocols try to manage the limited resources. In order to reduce the resource requirement, they try to use lightweight operations. However, there are no such protocol which consider that an object can be tagged with multiple number of RFID tags. To increase the detection probability of the objects, multi-tag environment can be assumed where each object is attached with multiple number of RFID tags. Multiple number of tags attached to an object also can help to increase the security. However, the existing approaches cannot be directly applicable to the multi-tag environment. This is because the signature of one tag can be re-signed by another tag attached to the same object whereas this is expected to be signed by a tag attached to another relevant object. Thus proof generation process will be disturbed. Also, we have to keep one set of security related information for an object to all the tags attached to the object and if the adversary manages to compromise any tag, he can compromise the other tags. Hence, he can easily compromise the object. However, if we keep separate security related information to the tags attached to the object, the adversary has to face more difficulty in order to compromise an object. Therefore, a separate protocol needs to be designed which can suitably be applied to multi-tag environment and increase the difficulty for the adversary to compromise an object.

1.2 Objective of the thesis

The objective of this thesis are

i) To fix the traceability problem in the existing authentication protocols: Most of the existing authentication protocols claim that they can prevent the tracking attack. Tracing of object is done on the basis of an unique response sent by the tag attached to it during the time when reader sends a request to it. Tracing an object, an attacker can collect several behavioral information of the object. The gathered information may extract someones' purchasing preferences or critical personal information such as location of any individual. For instance, RFID tags produce traces that may subsequently be used to track the position of an object attached with an individual. Various authentication schemes [**?**] [**?**] have been proposed to change the unique response of the tag after successful session in such a way that the attacker will have new response from the tag next time he will try to trace. Therefore, he will not be able to track the object. However, if the tag is read by a legitimate reader after a long time, then any non-legitimate reader may try

to trace the object during the time between two successive and successful sessions. Since non-legitimate readers do not have any valid authentication information, the attacker will not be able to successfully authenticate the tag and hence the tag will reply with same response when any reader sends request to it. Thus, the non-legitimate reader will be able to trace the object based on the response pattern by sending requests to it several time between two successful sessions. Therefore, it is possible for the attacker to track an object during the time between two successful sessions. The objective of this thesis is to propose a solution which can be implemented over the existing authentication protocols that suffered from this problem.

De-synchronization attack can be implemented in the transmission path between a tag and the reader and this can cause a synchronization problem between the object and the reader. The Fingerprint based authentication protocol [**?**] proposed by Khor et al. suffers from this problem. In their scheme, after authenticating the tag, the reader updates its session key value K_i and sends authentication information to tag. If the reader is authenticated by the tag, it updates the session key K_i. However, if the last message sent by the reader is blocked, the authentication information in the tag and the reader will become inconsistent. This is because the reader has already updated the session key. On the other hand, the tag has previous session key. Now, if a new session starts again, the valid tag will not be able to authenticate successfully. We have identified this problem and our objective is to fix this problem as part of the work in this thesis.

ii) To design authentication protocol for multi-tag RFID system: The existing authentication protocols emphasize on the security vs resource requirement. However, these schemes did not look into the detection probability of the object. Hence they use single tag in each object and only one set of security related information is kept in the tag attached to an object. Therefore, an adversary can easily compromise an object. The objective of this thesis is to propose separate authentication protocol which is suitable for multi-tag environment and can utilize multiple tags attached to an object which can increase the detection probability of the object and the difficulty for the adversary.

iii) To design object searching protocol for multi-tag RFID system: In similar to the existing authentication protocols, the existing object searching protocols consider single tag in each object. Hence the objects in these schemes also suffer from less detection probability and the difficulty for the adversaries is less. These schemes also incurred a huge amount of useless computations due to the fake responses from the undesired tags. The objective of thesis is to design a lightweight object searching scheme in multi-tag environment with negligible useless computation for fake responses.

iv) To design coexistence proof generation and validation protocol for multi-tag RFID system: In similar to the existing authentication protocols and the existing object searching

protocols, there is no such coexistence proof generation and validation protocol which consider the multi-tag environment i.e. the objects are attached with multiple number of RFID tags in order to increase the detection probability. These protocols cannot be modified to adopt in multi-tag environment since there is a probability that a signature of a tag can be resigned by another tag attached to the same object. The objective of this thesis is to design a coexistence proof generation and validation protocol for multiple relevant objects in multi-tag environment.

Due to various resource limitations of RFID technology and its pervasiveness property, the goal of this thesis is to design the protocols to keep a balance between the security and resource requirements which can enables the protocols to be applicable. An evaluation is to be carried out which can confirm the applicability of the protocols to be designed.

1.3 Contribution of the thesis

In this thesis, we have developed a solution which can be applied over an existing authentication scheme that suffers from the tracking attack during the time between two successful sessions. In addition to this, we have designed lightweight and secure protocols in multi-tag RFID system. The protocols are applicable to i) authenticate an object, ii) searching an object and iii) generating the coexistence proof of multiple number of relevant object. Proper analysis has been conducted in this thesis to evaluate the protocols. The contribution of the thesis are summarizes below.

i) Solution to the tracking attack during the time between two successful sessions: This solution is applicable over the authentication scheme which suffers from the tracking attack during the time between two successful sessions. In this solution, we use a pairwise secret information PIN and this information is updated using a Linear Feedback Shift Register (LFSR)operation. We also use a counter value which indicates how many time the PIN is updated. The RFID tag applies the LFSR operation over the most updated PIN value and then applies XOR operation over the message to be sent by the tag in the underlying authentication protocol. It then updates the counter value and the resultant encrypted information is sent to the backend server through the reader along with the updated counter value. The backend server applies the LFSR operation over the corresponding initial PIN value COUNT number of time and then applies the XOR operation over the received information. Thus it verifies the authentication of the tag. When COUNT reaches to its maximum value, the COUNT is reset to 0. The process of applying the LFSR by the backend server can be optimized by having a table to store some of the most frequent COUNT values and corresponding PIN values. Thus, instead of applying LFSR repeatedly COUNT number of times, we may directly get PIN from the table. Therefore,

the delay can be reduced for calculation of the updated PIN value.

The propose solution solves the tracking problem during the time between two successful sessions. Each time the reader sends the request to tag, the tag updates its counter and finds a new PIN to be used for XORing with the original message. Therefore, each response from the tag will be unique. We assume that the environment consists of many tags. Therefore, the tracking based on the COUNT value will be difficult. The proposed solution has to implement the LFSR operation and keep the counter value in the tag memory. To implement a counter of 16 bits, the required gate count is 144, and to apply the LFSR operation over a 16 bit PIN value, it requires a 16 bit LFSR i.e 144 gates. Therefore, the additional cost for this purpose is 288 gates. In summary, the existing authentication schemes which suffer from the tracking attack can be modified by incorporating this proposed solution to prevent from the tracking attack which requires to use 244 additional gates. This thesis also fix the de-synchronization problems in [**?**] by keeping the old information along with the most updated information about an object in the backend server.

In multi-tag environment, the objects are attached with multiple number of RFID tags in such a way that if any part of the object is under coverage area of the reader, there is atleast one tag which is within the coverage area of the reader. The existing authentication schemes can be modified to incorporate multiple number of tags in each object. However, we have to keep one set of security related information for an object to all the tags attached to the object since the existing schemes use only one set of security related information for the object. If an adversary somehow compromise a tag, he can compromise the other tags attached to the same object. Therefore, the effort requires for the adversary to compromise an object will be less. This motivated us to design separate authentication scheme for multi-tag environment.

ii) Object Authentication Using RFID Technology in Multi-tag environment: This thesis proposes two lightweight and secure authentication scheme in multi-tag environment. The components involved in these authentication schemes are a set of objects, a RFID reader and a backend server. In addition to this, each object is attached with m number of RFID tags and each tag is loaded with unique security related information such as session key, identifier etc. In the first authentication scheme, each tag attached to an object is assigned an index value and this index value is unique corresponding to the tags attached to an object. However, any two or more objects have the tags with same index value. In the backend server, a set of records is kept for the objects under consideration. Each record contains m number of sub records. These sub records contain the information of the tags attached to the corresponding object. An object is authenticated if the reader obtains atleast a threshold number of valid responses from the tags attached to the object. Depending on the application environment, an appropriate threshold value

needs to be chosen which can balance between the security and the detection probability of the object. This scheme uses basic operations in order to keep the scheme lightweight satisfying the possible security requirements. It also increases the difficulty for the adversary to compromise any object since the adversary has to compromise atleast the threshold number of tags in order to compromise an object. However, this scheme requires that all the tags attached to an object need to be allowed to prove the authentication for the object. Allowing all the tags attached to the object will increase the traffic between the reader and object. Therefore, we propose the second authentication protocol in this thesis.

The second authentication scheme proposed in this thesis also assumes that every object is attached with multiple number of RFID tags. In similar to the first authentication scheme, this scheme also keeps the information for the objects in the backend server. For each object, a tag called master tag is selected among all other tags attached to the corresponding object which is responsible to provide the authentication information in a particular session. In the same session, another tag is selected randomly to perform the authentication task in the next session. The record in the backend server contains additional secret information for the master tag. The authentication of the object is successful only after successfully assignment of the new master tag. Suppose an adversary somehow succeeds to clone the previously assigned master tag and tries to act as legitimate entity. However, since the master responsibility is not fixed to a particular tag and the adversary does not know the newly selected master tag, he cannot use the cloned tag alone to impersonate the object. He either has to guess and clone the newly selected master tag or clone all the tags attached to the object. This increases the difficulty for the adversary. If the tags are active, they can communicate with each other and hence the tag which belongs to the coverage area of the reader can help to reach the information from the reader to master tag or vice versa. Therefore, active tag as tag entity is the best choice for this scheme which can help to keep intact the detection probability in multi-tag arrangement. However, active tag is costlier than passive tag. We assume that the technology trends can make it affordable. Though the multi-tag concept increases the coverage area, the adversary cannot obtain any additional information. Therefore, the increased coverage area does not help the adversary. An authentication scheme can be used to search an object from a large set of objects. However the efficiency in this process is low which motivates the researchers to design separate protocol in order to solve the object searching problem. Howevr, in similar to the existing authentication schemes, the existing object searching schemes also cannot be applied in multi-tag environment which motivated us to design a separate object searching scheme for multi-tag environment.

iii) Object searching using RFID technology: This thesis proposes a separate lightweight

and secure object searching scheme for multi-tag environment. In the proposed scheme, each object is attached with multiple number of RFID tags and the search process requires at least the threshold number of legitimate responses in order to successfully detect the desired object. The threshold value is decided upon the application environment to balance between the security and detection probability of the object. In this scheme, the tags attached to the undesired objects respond to a search query using fake information with a certain probability which helps to prevent the information leakage attack. Probability of the useless computation due to these fake responses in the proposed scheme is negligible.

The object authentication and searching can be combined and this combination can be helpful in some other application. We found that there is an application which requires to generate a proof of coexistence of multiple number of relevant objects and this application needs the combination of object authentication and object searching mechanism. Because the relevant objects need to be searched and validated in order to generate a valid proof. Hence, we combine the object authentication and object searching to solve the proof generation problem.

iv) Protocol to generate and validate a proof of coexistence of multiple number of relevant objects This thesis proposes a separate proof generation and validation scheme for multi-tag environment due to the fact that the existing schemes are based on single tag environment and they cannot be applied on multi-tag environment. Initially, we propose a solution which can generate the proof of coexistence of a set of two relevant objects. We extend this solution to incorporate a set of more than two objects. The proposed scheme searches and validated the set of desired objects one by one and generates the proof incrementally, i.e. it uses the response of one object as the part of the request information for the next desired object while this part of the request information for the first desired object is a random number. It includes the response from an object into the proof after successfully verified the authentication of atleats the threshold number of responses from the tags attached to the desired object. The reader generates the proof and submits the proof to the backend server. This proof can be helpful in future in order to prove the coexistence of the set of relevant objects.

Various kind of analysis have been conducted in this thesis in order to evaluate the applicability of the proposed schemes. A thorough analysis has been done as a part of this work to find the possible attacks that can happen during the execution of the proposed protocols. Formal and informal analysis have been carried out to verify the robustness of the proposed schemes. The analysis shows that the proposed protocols can satisfy most of the security requirements. Each protocols also has been compared with the set of selected existing relevant protocols to investigate the strength of the proposed protocols in respect to the robustness criteria. Since the RFID technology has various resource constraints, this thesis analyzes and compared the

resource requirements in terms of computation, communication and storage.

In summary, the areas of contribution of this thesis are of three folds, i) object authentication, ii) object searching and iii) coexistence proof generation and validation. This thesis proposes a solution to fix the tracking attack problem in the existing authentication schemes and then fix the de-synchronization problem in [?]. It also proposed two authentication protocols, an object searching protocol and a coexistence proof generation protocol. The proposed protocols in this thesis are based on multi-tag RFID system and these protocols have been analyzed properly to evaluate the applicability.

1.4 Organization of the thesis

The rest of the thesis is organized as follows.

In **Chapter ??**, we describe the existing tools and mechanisms related to the work in this thesis.

In **Chapter 2**, we reviewed the existing protocols and analyzed the advantages and disadvantages of these protocols.

In **Chapter 3**, we analyzed that some existing authentication protocols using RFID technology suffer from the tracking attack. We also developed a solution in this chapter which can fix the tracking attack problem in the existing authentication schemes. This chapter also includes a solution to the de-synchronization problem in [**?**].

In **Chapter 4**, We have designed two authentication protocols which can be applicable in multi-tag environment. In this chapter, we initially describe and analyze the first protocol. However, this protocol has the problem of a traffic congestion between the reader and the object. To remedy, the second protocol is proposed and analyzed in this chapter which has less traffic congestion. A comparison is made with the existing protocols in this chapter

In **Chapter 5**, We have designed an object searching protocol applicable to multi-tag environment and this protocol is analyzed in this chapter for its applicability. We also have compared the proposed scheme with the existing schemes in this chapter.

In **Chapter 6**, a proof generation and validation protocol for multiple relevant objects in multi-tag environment has been propose and analyzed. This scheme also has been compared

with the existing proof generation and validation scheme.

Finally, **Chapter 7** concludes the thesis and provide a summary of the contribution. It also indicates the issues which have been opened and can be looked upon as the future research work.

Chapter 2

Background and Literature Review

Manual identification process is no longer require due to the introduction of Automatic Identification and Data Capture (AIDC) Technology [**?**,**?**]. In this technology, the labels are not required to read or enter data into an IT system manually. The labels are scanned and the corresponding data kept in the IT system are updated. Therefore the AIDC technology increases the speed and accuracy in the identification process.

2.1 Barcode technology

In 1952, Woodland et al. [**?**] show the world a unique way to capture the product information using a set of parallel lines with varying widths which can be referred to as linear or one dimensional. This kind of representation of data introduces the barcode concept. It became a popular AIDC technology. The barcode is read using a special optical scanner called barcode reader and it is decoded to obtain the actual data. Special softwares had also been developed for the devices like desktop printers and smart phones etc. which can be used to scan and interpret the barcodes. In order to scan the barcode, the scanning device is placed along the barcode in such a way that there is an appropriate alignment between the scanning device and the barcode. There are many mapping techniques or symbologies available such as continuous, discrete, two-width, many-width etc. [**?**]. In continuous mapping, one character starts with a bar and ends with a white space while in discrete mapping, each character starts and ends with a bar, and the inter-character spaces are ignored. The characters are recognized according to the width of the bars and spaces. In two-width scheme, there are two type of widths, namely, wide and narrow. In many-width scheme, there is a basic width and the other widths are the multiple of this width. Figure 2.1 depicts a sample barcode which encoded a name information.

Many variations were introduced in the barcode design to keep more information. For ex-

Figure 2.1: Sample barcode

ample, use of colours in barcode enables to keep more information in compared to traditional black and white barcode [**?**] [**?**] [**?**]. The barcode helps in many applications. For example, it can provide detailed up-to-date information on the progress of a business strategy which can help to confidently take any strategical decision. It is also useful in inventory tracking, logistics and supply chain management. However this technology has many pitfalls. The printed barcode can be erased or can be faded due to various environmental conditions and hence durability of the code is less. However, a barcode verifier can check the quality of a code by performing a series of tests such as edge determination, minimum reflectance, symbol contrast, minimum edge contrast, modulation, defects, and decodability etc.

In order to keep more information, two dimensional barcode have been proposed. Although these codes are made using various symbols like rectangles, dots, hexagon etc., they are referred as barcode. There are varieties of 2D barcode available in the market such as Quick Response (QR) code, aztec barcode, data matrix, semacode, etc.

Figure 2.2: Sample QR code

Among these 2D barcodes, the most popular is QR code. Figure 2.2 shows a samples of a QR code which encode the text “The author of this thesis is Subhasish Dhal”.

2.2 RFID technology

Radio Frequency Identification (RFID) technology advances the AIDC technology and improves the identification process. It relies upon radio frequencies and hence it does not require any line of sight. The speed in this technology is higher than barcode based AIDC technology since many objects can be identified simultaneously. It also can identify the objects over greater distance. These abilities of RFID technology further reduces the manual involvement in identification process. For example, each item does not require to remove from a shopping cart in order to scan the cart which reduces the transaction cost for the retailers and check out time of the customers. The another benefit in RFID technology is that it consists of rewritable memory, environmental sensors and many security features which enables it to catch the history of events. It also introduces the dynamic or smart labels which can capture the data about the object in absence of any people. However, these features are absent in barcode or similar kind of technologies.

2.2.1 RFID system components

The components involved in RFID technology are RFID tags (also known as transponders), RFID reader, and enterprise subsystem. RFID tags are small chips which are usually attached to or embedded in the objects. The tags contain identifiers of the objects which are the key information in order to identify the objects. RFID readers are the electronic devices which wirelessly communicate with the RFID tags in order to identify the corresponding objects with the help of a server. The reader also communicates with the enterprise subsystem. However, the communication between the reader and the enterprise subsystem can be either wired or wireless. Figure 2.3 illustrates a simple RFID system.

Figure 2.3: RFID system

2.2.1.1 RFID tag

A typical RFID tag consists of an antenna and an integrated circuitry. The antenna captures radio signal from a RFID reader and passes this signal to the integrated circuitry. The integrated circuitry processes this signal, and responds through antenna after necessary computations. Some kinds of RFID tags use power in the received signal for computation and response while some other kinds of tag has their own battery. The integrated circuitry is composed of modulator, demodulator, encoder, decoder, an AC/DC converter circuit, a controller and sequencer, and memory. Figure 2.4 shows a simple block diagram of the circuit inside a typical RFID tag. The

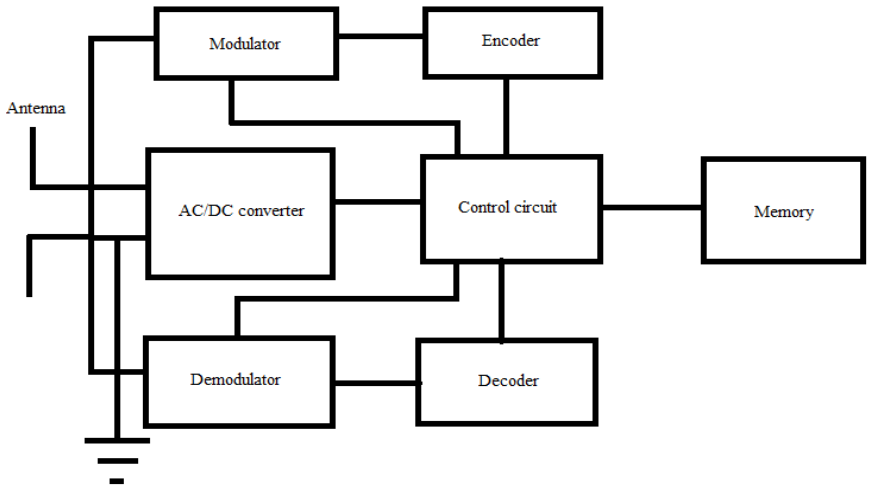

Figure 2.4: Baic components in a RFID Tag

received signal is demodulated and decoded, and then processed in the integrated circuitry. The memory inside the integrated circuitry is used to store data about the object. It is a non-volatile memory and is either write once read many (WORM) or re-writeable memory. The tag when response, retrieves the necessary information (e.g., identifier) from the memory and produces a backscattering signal after encoding and modulating the response information.

Tag characteristics: The RFID tags differ greatly according to cost, size, performance, and security mechanism. The significant characteristics of the tags are identifier format, power source, operating frequencies, and form factor. The identifier of a tag helps to identify an object and if their is a certain format for the identifier then it can further improve the efficiency in the identification process. The mostly used identifier format is Electronic Product Code (EPC)

which was developed by EPCglobal . However, a standard format can be vulnerable to various business and privacy risks. Hence, many organizations use their own identifier format which cannot reveal any information.

Necessary power is required in tag in order to perform various operations such as storing and retrieving information, responding with radio signal, and performing necessary computation. The tags can be classified in four categories according to the power sources.

- **Passive:** It obtains power from the electromagnetic energy it receives from the reader and uses this power for performing all the necessary tasks. The limitation in extracted energy restricts its computation and communication ability. However, it is cheaper and smaller in absence of a bulky power source and hence it is used in many applications.

- **Active:** It has its own battery for performing computation and communication which enables it to perform complex computations and communicate over greater distance. However, the battery life is limited and this kind of tag is expensive due to an internal power source which makes it bulky.

- **Semi-active:** This kind of tag has an internal power source which it uses to communicate with the reader. Hence it also can communicate over longer distance. In some articles, this kind of tag is known as semi-passive tag

- **Semi-passive:** This kind of tag has a battery which is used for performing various computations. In comparison, they are cheaper than active tags and have greater functionalities than passive tags. In some articles, this kind of tag is known as semi-active.

The RFID tags can operate in various frequency range (form UHF to LF). High frequency signal can carry more data and hence the readers can read more tags in a given period of time. The high frequency tags relies on backscattering or far field propagation while the low frequency tags use inductive coupling. The operating frequency is also plays a significant role for the communication distance. Usually this distance varies from 0.1 to 100 meters.

The tags can also be classified according to the form factor i.e., the size, shape, packaging, and handling features. The active tags are usually larger than the passive tags since they have onboard power supply. The tags which are integrated with environmental sensors are also larger. However, these tags have more computing functionalities.

2.2.1.2 RFID reader

The RFID reader emanates electromagnetic signal wirelessly through its antenna and receives the responses from the tags. This device can be a stationary or a mobile and has to comply

with the standard of the tags in order to communicated with the tags i.e., it has to use the same frequency, same communication protocol etc. The communication between the tag and reader can occur in two ways.

- **Reader talks first:** In this communication, the reader initiates by broadcasting a signal and the tags within the coverage area of the reader receive this signal, and respond to the reader.
- **Tags talks first:** In this communication, the tags initiate the communication. The active tags are periodically transmit signal until the power supply lasts. On the other hand, the passive tags transmit the signal when they get power from the reader's signal.

This communication is easily accessible to the attackers. As a result, they can utilize this to mount various kinds of attacks. Also, the responses from the tags can collide in this communication path since multiple tags are permitted to response simultaneously. In order to avoid collisions, the reader follows certain algorithms like singulation which can distinguish the original intended entity. However, some RFID applications do not need any such collision avoidance algorithm e.g., the animal tracking system using RFID technology, since there is a rare probability of any collision in a close proximity. Therefore, the power consumption due to the execution of the collision avoidance algorithm can be avoided in these applications.

2.2.1.3 Enterprise subsystem

Sometimes, the reader identifies an object with the help of the enterprise subsystem. The enterprise subsystem prepares the information received from the reader and processes the received information with the help of a network of computers. The major components in the enterprise subsystem are as follows:

- **Middle-ware:** It is responsible for preparing the information received from the reader which can be processed by the computers running softwares. It filters the duplicates, erroneous, and incomplete information.
- **Analytic system:** It composed of a number of web servers, databases and the set of application specific softwares which processes, retrieves, and stores the data in order to accomplish the application specific tasks.
- **Network infrastructure:** It interconnects the web servers in the enterprise subsystem. The most important characteristics of this components are the underlying topology, communication protocols, and security issues

2.3 Multi-tag RFID system

The objects are identifiable if the tags attached to them are detected by the RFID reader and hence the tags need to be within the coverage area of the reader. The detection probability of an object is less if the object is attached with a single tag [**?**]. Because the average incidence angle between the tag and the radio signal is less in this arrangement. Bolotnyy et al. [**?**] show that the average incidence angle can be increased by attaching multiple number of tags in proper alignment. Therefore, the detection probability of the object can be increased in this arrangement.

2.3.1 Incidence angle

The detection probability of a tag depends on two criteria (i) The tag needs to be within the coverage area of a reader and (ii) The voltage induced in the tag is sufficient. The voltage induced in the tag depends on the incidence angle between the tag antenna and the perpendicular direction of the radio signal. There are two kinds of coupling between the tag antenna and radio signal (i) inductive coupling and (iii) backscattering (far field propagation). Inductive coupling takes place when the frequency of the radio signal from the reader is low. Figure 2.5 shows the incidence angle between the tag antenna and the perpendicular direction of the radio signal in inductive coupling mechanism. The angle between tag antenna and the perpendicular direction

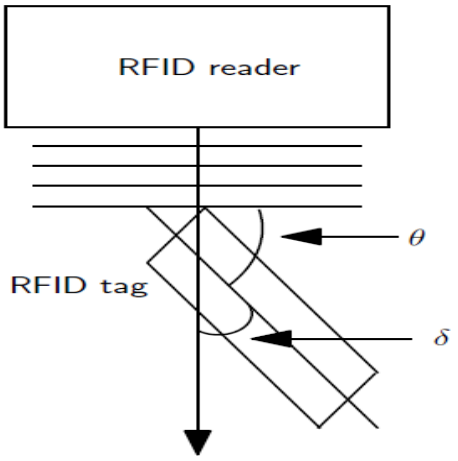

Figure 2.5: Angle between tag and reader

of the radio signal is θ and the voltage induced in the tag is $V_0 = 2\pi f N S B_0 cos\theta$, where f = Frequency of the arrival signal, N = Number of turns in the coil of the loop, S = Area in the loop in meter (m^2), B_0 = Strength of the arrival signal, θ = Angle of the arrival signal. Therefore, if the incidence angle (θ) is low, the induced voltage will be high and maximum when $\theta = 0^o$. This

equation can be slightly modified to $V_0 = 2\pi f N S B_0 sin\delta$, where $\delta = 90^0 - \theta$. In this equation, the induced voltage (V_0) will be increased on increase of the δ value. Similarly, in far field propagation, the induced voltage depends on the gain of the tag antenna which is proportional to Poynting's vector $p = E \times H$ where E is the electric field intensity and H is the magnetic field intensity. Both H and E are equivalent to $sin\delta$. Therefore, the induced voltage V_0 is equivalent to $sin^2\delta$. According to these equations, for both inductive coupling and far field propagation, the expected incidence angle δ needs to closer to 90^o in order to achieve maximum induced voltage.

2.3.2 Tag orientation

The angle between the tag antenna and the radio signal is not fixed since the readers and the tags can move. Therefore, even if the tag is within the coverage area of the reader, it may not be induced with sufficient voltage which can help to perform its operations. This necessitates the use of multiple number of tags in an object which can ensure that at least one tag will be induced with sufficient voltage. However, the orientation of the tag attachment is an important task since the direction of the signal from the reader is arbitrary. For a single tag, the tag can be attached anywhere in the object since it does not affect the incidence angle. For two tags, the tags can be attached perpendicular to each other i.e., one tag can be attached in X-Y plane while the other tag in X-Z plane. For three tags, the tags can be attached pairwise perpendicularly in X-Y, X-Z, Y-Z planes. Figure 2.6 shows the tag orientation for three tags. Similarly, for n tags, the tags

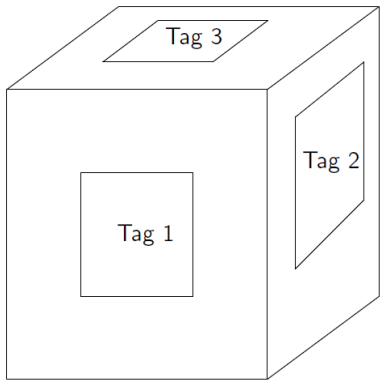

Figure 2.6: Tag orientation for three tags

are placed in parallel to the faces of an n dimensional shape in order to maximize the incidence angle to any one tag. The theoretical analysis reveals that the equation for the average incidence angle for one tag is:

$$\frac{\int_0^{\frac{\pi}{2}} x(2\pi cos(x))\mathrm{d}x}{2\pi} \approx 32.7.$$

The equation for the average incidence angle for two tags is:

$$\frac{\int_0^{\frac{\pi}{4}} x(2\pi cos(x))\mathrm{d}x + \int_{\frac{\pi}{4}}^{\frac{\pi}{2}} (\frac{\pi}{2} - x)(2\pi cos(x))\mathrm{d}x}{\pi} \approx 48.0.$$

The equations for more number of tags are very complex and hence a simulation has been conducted in [**?**] which shows that the average incidence angle for three tags is 58.11 and for four tags is 61.86. Therefore, the expected incidence angle is increased in the increase of the number of tags and this in turns increases the detection probability of the object.

2.4 Security and privacy issues in RFID technology

The communication between reader and tags is wireless. Hence, this communication is easily accessible to non-legitimate entities and they can misuse it. There are many security and privacy issues involved in this communication [**?**, **?**] [**?**, **?**, **?**, **?**, **?**, **?**] [**?**, **?**]. The sensitive content in the tags can be revealed through an insecure interrogation. Therefore, the entities involved in this communication should be authenticated. The tags usually reply with a constant answer on repeated queries and hence the objects associated to the tags can be tracked by a non-legitimate entity. The information can be eavesdropped through the radio signal. Any suitable encryption may prevent this. However, traffic analysis is sill feasible through which the non-legitimate entities can extract secure information. The memory of the tag is also insecure and its contents can be accessible through physical attack and in turn, the content of the tags can be modified. The communication can also be desynchronized and hence denial of service attack is possible which can stop further communication between the reader and the tags.

2.4.1 Cryptographic primitives

There are many cryptographic primitives available in the literature in order to provide security in different kinds of communications. However, the standard cryptographic primitives are not applicable in RFID technology due to the severe resource limitations in tag design. Recently, a few lightweight cryptographic primitives (cipher, hash functions, pseudo random number generation function) have been designed in order to provide security in RFID or similar kind of technologies which are suffer from severe resource constraints.

2.4.1.1 Lightweight ciphers

The ciphers in cryptography can be divided in two categories: symmetric cipher and asymmetric cipher. The symmetric cipher has two broad categories block ciphers and stream ciphers. In

block ciphers, a fixed length block of plain text data is transformed into a same length cipher-text data. The length of the plain text data is usually 64-128 bits. The messages longer than this length are usually encrypted using an efficient mode of operation. There are many block ciphers available in the literature. Feldhofer et al. [?, ?, ?, ?, ?, ?], Satoh et al. [?, ?, ?, ?, ?, ?, ?, ?], and Pramstaller et al. [?, ?, ?, ?, ?, ?, ?, ?] proposed optimized and efficient version of Advanced Encryption Standard (AES -128) for 8-bit architecture. The proposal by Satoh et al. and Pramstaller et al. are more superior in respect to throughput. However, the proposal by Feldhofer et al. requires less chip area and the power consumption is only 4.5 μW. Poschmann et al. [?, ?, ?, ?, ?, ?, ?, ?, ?] proposed a lightweight version of Data Encryption Standard (DES) and its variations (DESX, DESL, DESXL). Howevr, a brute force attack on DES is possible using a special purpose machine COPACOBANA [?, ?, ?, ?, ?, ?, ?, ?, ?] and this machine can break the code within a few days. A key-whitening technique can be employed to form DESX which can provide higher security. However, it increases the key size. Poschmann et al. then proposed DES lightweight extension (DESL) by replacing eight S-boxes with only one S-box. Finally, Poschmann et al. used key-whitening technique (yielding DESXL) in DESL to provide higher security. A ultra-lightweight block cipher, namely, Present was proposed by Bogdanov et al. [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?] which can encrypt 64-bit block of plain text. It has two variations: Present-80 (80 bit key) and Present-128 (128-bit key). Present-80 requires only 32 clock cycles and it is more efficient than AES-128 [?, ?, ?, ?, ?, ?]. However, a deeper security analysis is required [?, ?, ?, ?, ?] to evaluate the security of Present-80 and present-128. The other block ciphers are HIGHT, Clefia, SEA, TEA, etc. Hight is a 64-bit block cipher and it uses 128-bit key [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?] and it requires 34 clock cycles. On the other hand, Clefia is a 128 block cipher and it supports 128, 192, and 256 bit key length [?, ?, ?, ?, ?, ?, ?, ?, ?]. Another block cipher, namely, Scalable Encryption Algorithm (SEA) was developed for low cost embedded applications [?, ?, ?, ?, ?, ?, ?, ?] [?, ?, ?, ?, ?, ?, ?] and it supports key size n and plaintext size n bits. It can be implemented in n-bit processor. In addition to software implementation of SEA, an implementation in field programming gate array (FPGA) had also been verified [?, ?, ?, ?, ?, ?, ?, ?]. Tiny Encryption Algorithm (TEA) and its variations (XTEA, XXTEA) are the another kind of lightweight ciphers applicable in RFID technology [?, ?, ?, ?, ?] [?, ?, ?, ?, ?, ?] [?, ?, ?, ?, ?, ?]. It uses lightweight operations such as addition, XOR, and shift and it can be implemented in 8-bit platform.

In contrast to block ciphers, the stream ciphers operate on a stream of bits. The size of the plain text in these ciphers are not fixed. Grain is a kind of lightweight stream cipher which is suitable for the devices with limited resources (gate count, power consumption, and memory) [?, ?, ?, ?, ?, ?] [?, ?, ?, ?, ?, ?]. It is a bit-oriented synchronous stream cipher and hence the

generation of key stream is independent from the plain text. Two 80-bit shift registers (one Linear feedback shift register (LFSR) and one non-linear feedback shift register (NFSR)) have been used in this cipher. The size of the key and initial vector (IV) are 80-bits and 64-bits respectively. Trivium is another synchronous stream cipher which have been designed to provide a balance between efficiency and resources [?, ?, ?, ?]. It can generate upto 2^{64} bits output from 80-bit key and IV. Other variations of Trivium are Trivium-8, Trivium-16, Trivium-32, Trivium-64. Likewise, there are many other lightweight stream cipher available in the literature such as Lex, Mickey-128, Salsa20, etc. [?, ?, ?, ?, ?, ?]. The hardware evaluation of these stream ciphers can be found in [?, ?, ?, ?, ?, ?] [?, ?, ?, ?] [?, ?, ?, ?, ?, ?, ?].

Although the symmetric ciphers are efficient, they require to exchange the secret keys between the parties involved in a secure communication. In asymmetric ciphers, the keys are not require to exchange in order to perform a secure communication. Two mathematically related keys are used in these ciphers where one key (private key) remains secret and the other key (public key) is publicly distributed. However, these ciphers are inefficient and cannot be implemented in RFID applications. Recently Batina et al. [?] proposed an Elliptic Curve Cryptography (ECC) based encryption scheme which has been commercially accepted and endorsed by the US government. This becomes an attractive public key cryptosystem in compared to traditional RSA or discrete logarithmic cryptosystems.

2.4.1.2 Lightweight hash functions

This function takes any arbitrary input and provides a fixed length output satisfying the following conditions: Preimage resistant, Second preimage resistant and collision resistant. Hash functions are popularly used in many RFID application. However, the selection of appropriate hash function is an important issue. The hash functions built using the Markle-Damgard construction are vulnerable to length extension attack [?]. In addition to this, implementation of most of the hash functions are not possible with limited resources (250-4K gates). For example, the best implementation of SHA-256 requires almost 11K gates and a calculation on a data block of 512-bits requires almost 1120 clock cycles [?, ?, ?, ?]. The other hash functions such as SHA-1, MD5 etc. are also cannot fit in a RFID tag [?]. A "universal hash function" is suggested in [?, ?, ?, ?, ?, ?]. However, this hash function provides only 64-bit output which is not secure from collision because of the birthday paradox (requires around 2^{32} operations) [?]. Recently, a low cost hash function was proposed by Peris et al, namely, Tav-128 which can be implemented with proper security in RFID or similar kind of applications [?, ?, ?, ?, ?, ?, ?]. Only 2578 logic gates are required in order to implement Tav-128. It can be executed in 1568 clock cycles and the throughput is 8.2 kbps at 100 kHz. However, according to the authors, this throughput can

be increased by 25 percent.

2.4.1.3 Lightweight pseudo random number generators

Many RFID applications require to generate appropriate random numbers which is an important requirement to provide security. Many random number generation functions available. However, all the random number generation functions are not applicable in RFID or similar kind of application since these applications have severe resource constraints. A Pseudo Random Number Generation (PRNG) function should consider the hardware limitations of a RFID application while it can satisfy the necessary randomness criteria. The randomness criteria according to the specification of the RFID tags [**?**] based on Electronic Product Code Class-I generation-2 (EPC-C1G2) and ISO/IEC 18000-6:2004/Amd:2006 are (i) Probability of a single *RN16*, (ii) Probability of simultaneously identical sequences, (iii) Probability of predicting an *RN16*. A PRNG, namely, LAMED conforming the specification of EPC-C1G2 have been proposed recently by Peris et al. [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**] using the genetic programming techniques. The output of this PRNG is 32 bits. However, the specification of EPC-C1G2 demands a 16 bits output. In order to provide this requirement, Peris et al. designed a 16-bit version (LAMED-EPC) PRNG. They extend the 32-bit version PRNG by performing the XOR operation over the 16-bit MSBs and 16-bit LSBs of the original output (from 32-bit PRNG) to produce the final output of 16 bits. This provides an additional advantage of availability of a 32-bit PRNG along with 16-bit PRNG. This PRNG satisfies the required randomness criteria which was evaluated using ENT [**?**,**?**,**?**], Diehard [**?**,**?**] [**?**,**?**,**?**,**?**,**?**,**?**,**?**], NIST [**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**], and David Sexton [**?**, **?**, **?**] etc. The authors implemented the LAMED and LAMED-EPC where the former needs 1566 logic gates and the later requires 1585 logic gates. The throughput of LAMED and LAMED-EPC are 17.2 and 8.2 kbps respectively.

2.5 Protocols in RFID technology

Advancement in RFID technology motivates the use of this technology in diverse field of applications [**?**] [**?**]. Hence the engineering research focused on using this technology efficiently. However, due to the pervasiveness property, this technology is vulnerable to various kind of misuses. The researchers put their efforts to adopt this technology in various applications with necessary prevention mechanisms which can help to prevent the possible misuses. The resource constraint makes the adoption of this technology more challenging. Therefore, the researchers are searching for cost effective and secure solutions which can make this technology more popular to be useful in the industries. We visited the literature to get an idea about the research and

developments in the field of RFID technology. We found that the researchers mostly use this technology in those applications where the object identification is a basic requirement. Object authentication, object searching, object binding, object ownership transfer, etc. are such applications which requires to identify one or more legitimate objects. In this thesis, we emphasize on three applications and they are object authentication, object searching and coexistence proof generation and validation for multiple number of relevant objects. Hence, it highlights a number of existing protocols which try to solve the requirements in these applications. In addition to this, it finds out the problems in this direction that need to be solved.

2.5.1 Existing object authentication protocols

Object authentication is a mandatory task which can prevent any non-legitimated access. Due to the pervasiveness property of RFID technology, the authentication process needs to be secured. The classical cryptography techniques cannot be applicable since the RFID technology suffers from limited resources in terms of computation, communication and storage. Therefore, a lightweight cryptography technique is suitable for this technology. However, the lightweight solutions can be vulnerable to various kind of security threats. Therefore, the existing works try to optimize the resources by using lightweight operations which can satisfy most of the security requirements. We classify the existing authentication schemes based on the operations used in various authentication schemes.

2.5.1.1 Hash based authentication schemes

Hash operation can be used to generate authentication information and this can be verified for legitimacy of an entity. Many authors proposed hash operation based authentication schemes in the literature. Wies et al. [**?**] proposed a hash based authentication scheme in 2004. In their scheme, a tag will be given a meta id which is a hashed value of a secret key. It obtains the meta-id from a secure source and saves in its memory. After getting this id, the tag gets locked. Reader initiates a communication and the tag responds with the meta id it has assigned. Reader checks the validity of the tag and sends the corresponding secret key on validation. This secret key is used to unlock the tag and the tag sends its id to reader. The reader consults with the backend server to obtain the detail information about the same object. This scheme is simple, however, suffers from many attacks such as eavesdropping, location privacy, replay attack, etc. For example, an adversary may trace the object since the tag always responds with same metaID.

Wies et al. [**?**] have modified their scheme and proposed a Randomized Access Control (RAC) scheme to avoid the location privacy problem. In this scheme, the tag generates a random number and computes the hash value of this random number and its identifier. Since each

response from the tag consists of a new random number, the location privacy problem is resolved. However, this scheme also cannot avoid the attacks like replay attack, location privacy between two successful sessions, etc.

Tsudik proposed an authentication protocol YA-TRAP in [**?**] which uses time stamp and a random challenge (nonce) in the authentication process. In this scheme, the tag and reader keeps a time stamp value. The reader broadcasts the time stamp it kept along with a random challenge. The tag verifies the validity of the time stamp and responds with a valid authentication information. It generates this information by applying a hash operation over the random challenge it received and the random number it generated using a Pseudo Random Number Generation (PRNG) function. However, it responds with a random information on unsuccessful verification of the timestamp it received. The reader verifies the response and authenticates the tag. This scheme suffers from Denial of service (DOS) attack as mentioned in [**?**] and hence the authors have modified the scheme to remove this attack in [**?**]. The modified scheme YA-TRAP* uses a token called epoch token in addition to the time stamp and random challenge. The value of the epoch token changes slowly which helps to avoid the DOS attack. However, both YA-TRAP and YA-TRAP* suffers from forward security problem. Becasue the random number in the tag is upgraded using a PRNG function and the adversary can upgrade this random number using the same function.

Burmester et al. extends the YA-TRAP proposed by Tsudik [**?**] in [**?**], namely, YA-TRAP$^+$ to remove the DOS attack. In this scheme, the tag responds with a value generated using a hash operation. This hash operation had applied over a code and the concatenation value of the time stamp information and the random challenge it received from the reader. This code is used to indicate that the received timestamp value is older or equal to the timestamp value kept in the tag memory. However, the tag responds with a random value when the received time stamp value is neither older nor equal. Therefore, when the trusted server finds a valid response, it can detect that a few attempts have been made to desynchronize the tag. In this case, the trusted server resynchronized the tags. Thus, it prevents the de-synchronization attack. However, this scheme also suffers from forward and backward secrecy problem. Burmester et al. proposed another protocol in the same paper, namely, O-TRAP. In this protocol, the tag responds with an authentication information which was generated applying a hash operation over a random challenge it received from the reader and the upgraded random number it contains. It upgrades the random number it contains using a keyed hash operation. This protocol is safe from the DOS attack, However, heavy computation required in the case when the adversary successfully mount the de-synchronization attack several time during the time between two legitimate sessions.

Conti et al. proposed a hash chain based authentication protocol in [**?**], namely RIPP-FS.

This protocol is basically inspired by YA-TRAP [?] and YA-TRAP* [?] proposed by Tsudik. In this protocol, the tag updates its key in a chain of hash operations and responds with a value generated after applying the keyed hash operation over the time stamp information it received from the reader. The drawback in this scheme is that all the keys started with a seed value are reveled after finite number of authentication sessions. Moreover, if an adversary sends a time stamp information which is much bigger than the time stamp information stored in the tag memory, the tag has to compute a bigger chain of hash operations. In this case, the tag has to perform a lot of useless computations.

Tan et al. proposed a serverless authentication scheme in [?]. In this scheme, the reader initiates a session by broadcasting a request message. The tag responds with a random number. The reader then replies with its id along with another random number. The tag then generates a value applying a hash operation over the concatenated value of a function output and two random numbers. It then responds with the first few bits of this hash value along with a challenge. The reader verifies this response and responds with an answer corresponding to the challenge from the tag. It also sends a challenge to the tag. The tag verifies the answer from the reader and responds with an answer to the corresponding challenge from the reader. The reader verifies the answer from the tag and stops. In this scheme, the reader has to manage a list which can be large and hence the reader may suffer from the scalability problem. This scheme also consists of many communication between the reader and the tag due to many challenges and answers. The authors have proposed a second version of the protocol in the same paper [?]. In this protocol, the tag uses the XOR operation to generate the authentication information. It applies the XOR operation over its id and the hash value of the concatenated value of a function output and two random numbers. It then sends the resultant value along with first few bits of this hash value. The reader checks a valid entry in its list and determines the identification information. This protocol reduces a few communication steps. However, the reader in this scheme also suffer from scalability problem. In both the scheme described in this paper, since the tag responds with a few bits (instead of whole) of the hash value, this can conflict with the response from any other tag.

Song and Mitchell [?] have proposed an authentication scheme. In this scheme, the tag and the server both will keep two parameters. One of these two parameters is a unique string and the other is the hash value of this unique string. The hashed value acts as a pairwise secret between the tag and the server. The tag in this scheme responds with an encrypted random number and an integrity information on a request from the reader. The server decrypts the random number and verifies it for integrity. It then updates various parameters and sends an update information to the tag. The tag then updates its parameters. This scheme suffers from high computation

overhead due to hash and MAC operations. Some attacks against this scheme are also reported in [**?**].

In the authentication scheme proposed by Zhang et al.in [**?**], the reader collects identifier and secrete key from the backend server to prove the legitimacy to the tag. The tag attached to an object responds with a session random number. The reader proves its legitimacy using this random number along with its own identifier and the session key. The tag proves its legitimacy by providing its identifier and after the validation of the identifier, the reader generates a new session random number and sends this value to tag. The tag receives the new session random number and replies with a OK message. This scheme may suffer form synchronization problem since an adversary may block the original updated session random number and send another arbitrary value. However, the tag is unable to detect the validity of the new session random number and hence the tag and the backend server will be desynchronized. This scheme also suffers from the location privacy problem during the time between two successful authentication sessions. If there are more readers then the tag needs to keep pairwise secrets for all the reader and hence the tag may suffer from scalability problem.

2.5.1.2 Multiplication operation based authentication schemes

Hopper and Blum proposed an Human authentication protocol in [**?**] known as HB protocol. Their protocol uses multiplication operation to encrypt the secret key. In addition to this, it adds some noises to the encrypted information. Hence, the security of this protocol depends on the hardness of Learning Parity with Noise (LPN) problem. However this scheme cannot prevent the active attacks as mentioned in [**?**] [**?**]. Juels and Weis proposed an improvement over the HB protocol known as HB^+ in [**?**], which can prevent the active attacks. This protocol uses two secrets instead of one. However, this scheme is vulnerable to man-in-the-middle attack as described in [**?**] [**?**]. Bringer et al. improved the HB^+ protocol known as HB^{++} in [**?**]. In this protocol, they use two more secrets and a function. Actually, $HB^+ +$ is equivalent to running twice the HB^+ under independent secrets with correlated challenges. However, as per the reports in [**?**] and [**?**], HB^{++} is vulnerable to gain the knowledge of tag's private key. Moreover, HB, HB^+ and HB^{++} are vulnerable to traceability and side channel attack as mentioned in [**?**]. Therefore, these protocols are modified in [**?**] in order to remove the problems of the authentication schemes in HB family. In this modification, the crux of the HB^+ keeps intact. In addition to this, a parameter along with the relevant vector are removed to prevent the side channel attack. This modification keeps the protocol lightweight. It also updates the security parameter in each session to prevent the tracking attack.

In 2005, Karthikeyan and Nesterenko [**?**] proposed a RFID tag identification protocol that

incorporates the reader authentication. The algorithm is secure against the anticipated threats to RFID systems. It relies on simple matrix multiplication rather than any expensive cryptographic operation. However, this algorithm suffers from tracking attack during the time between two successful sessions. This scheme is also suffer from other attacks such as Denial of Services attack (DOS), replay attack as reported in [**?**]

2.5.1.3 Substring function based authentication schemes

Li et al. proposed a substring function based authentication scheme [**?**] in which the tag and the backend server keep an id of the tag. A substring function is used to extract a substring of a particular length from the id of the tag. In this scheme, the tag generates the authentication information by applying XOR operation over two random numbers and the substring obtained after applying the substring function over its id. The backend server finds a record which can match with the received authentication information. In order to obtain the final confirmation, the backend server sends another substring to the tag and the tag replies with OK or NO message. Chien et al. [**?**] show that this protocol is unable to authenticate a tag successfully. Moreover, this protocol suffers from the replay attack and the tracking attack.

Chien et al. improved the scheme proposed by Li et al. in [**?**]. In this scheme, the tag and the backend server share a secret key. The tag generates the authentication information by applying a substring operation over the XORed value of a random number and the another value which obtained after applying a rotation operation over the id of the tag. If the backend server finds a match, it further verifies the authentication of the tag by sending another substring to the tag. The tag finally confirms with OK or NO message. He Lei et al. mentioned in [**?**] that this protocol cannot provide forward secrecy requirement since the pairwise secrete does not get updated.

He Lei et al. [**?**] improved the protocol proposed in [**?**] which removes the vulnerabilities in the protocols proposed in [**?**] and [**?**]. In this protocol, the tag and the backend server maintain a counter and initially the value of this counter in backend server and the tag are same. In a request from the reader, the tag responds with a substring and the counter value it kept in its memory. The backend server verifies the validity of these information and compares the received counter value with the counter value kept in the corresponding record. If it finds that the received counter value is greater, it updates the pairwise secret until the two counters synchronized with each other. The backend server replies with another substring and the tag sends OK or NO message after verification of this substring. In the same step, the tag updates the pairwise secret key kept in its memory. This scheme also suffers from many communication steps. Though the authors claim that their scheme provides forward secrecy criteria. However, an adversary can compute the updated secret key by using a simple function. He can perform this operation by obtaining

the counter value and apply the function (since the function is not a secret information) over the compromised key and the counter value.

2.5.1.4 Fingerprint based authentication schemes

Khor et al. proposed an authentication scheme in [?] which uses Pseudo random number generator (PRNG), Cycle Redundancy Check (CRC), and XOR operations as the fundamental operations in their protocol. The encryption key is updated in each successful session using the PRNG function. The unique electronic fingerprint information of a tag is kept in its memory and in the backend server, and using this information, the protocol solves the problem of impersonation attack, eavesdropping attack, etc. However, in this scheme, the session key is updated only after the completion of a successful session. Therefore, during the time between two successful sessions, the session key remains same and the tag can be tracked during this period. Because the tag will respond with same information on multiple request from the adversary. In this scheme, the backend server updates its session key and sends a message to the tag. After successful verification of this message, the tag updates its session key. However, if the message sent by the reader is blocked by an adversary then the backend server has already updated the session key whereas the tag has the old session key. Therefore, the tag and the backend server are no more synchronized with each other. Hence the de-synchronization attack is another problem in this scheme.

Lars Kulseng et al. [?] presents a lightweight authentication scheme for RFID systems in which only the authenticated readers and tags can successfully communicate with each other. It uses Physically Unclonable Function (PUF) and Linear Feedback Shift Register (LFSR). The PUF function generates a unique id of the tag using the delay property of the hardware implementation of the tag. In this scheme, the tag uses an index to the id of the tag as a response to a request from the reader. This index is updated in each successful session. Though this scheme solves the problem of tracking attack, eavesdropping, etc., it suffers from the problem of tracking during the time between two successful sessions. Because the index value is updated only after the completion of a successful session. Therefore, during the time between two successful sessions, the tag will respond with same index value on multiple request from the adversary.

2.5.1.5 Psudo Random number generation based authentication schemes

Duc et al. proposed an authentication protocol in [?] using PRNG, Cycle Redundancy Check (CRC) and XOR operation. In their scheme, the tag and the backend server shared two keys and one pin. During the authentication process, the tag responds with the authentication information which were generated using CRC and XOR operation. The backend server verifies this response

and sends the object information to the reader and sends a confirmation information to the tag. The tag and the backend server then updates their parameters. This scheme cannot prevent the denial of service attack. It also unable to provide forward secrecy since the updation occurs using a known function only.

Chien and Chen [**?**] modified the scheme proposed by Duc et al. [**?**] which removes the denial of service attack by keeping the old information along with the new information for the tag in the backend server. However, according to the security analysis in [**?**], the denial service attack can be mounted to this scheme using two attempts. They modified the updation process by introducing PRNG function. However, the adversary can compute the updated key using the same PRNG function.

Tzu-Chang et al. has improved the scheme proposed by Chien and Chen [**?**] in [**?**]. They removed the denial of service threat and introduced the indexing mechanism in the database in the backend server to speedup the authentication process. In the protocol proposed by Tzu Chang et al., the tag replies with an authentication information along with the corresponding index value. The reader forwards these values to the backend server along with its own signature information and the random number it had sent to the tag. The backend server verifies the authentication of both the reader and the tag. The tag updates the information kept in its memory after obtaining the updates from the backend server. This scheme has the following problems. If an attacker changes the value of the index information to 0, then the index value will be useless. This scheme also suffers from tracking attack since the tag responds with same index value on multiple requests from an adversary during the time between two successful sessions. This scheme also does not guarantee the forward secrecy requirement since the adversary can generate the updated key using PRNG function.

Hoque et al. proposed a serverless authentication scheme in [**?**]. They assume that multiple readers can participate in the authentication process simultaneously. A certificate authority (CA) authorizes the readers and a set of tags are assigned to each reader. A reader contains a list of tags assigned to it. Each entry in the list contains an id and a seed value of a tag authorized to it. A tag contains the seed value of the corresponding reader and a secret key. During the authentication process, the reader initiates the communication by broadcasting the request message with a random nonce. The tag responds with an authentication information along with a random number. After receiving the response from the tag, the reader verifies the validity of this response and sends the update information to the tag. The tag verifies this update information and updates its memory on successful verification. This protocol suffers from the denial of service attack and the reader may suffer from the scalability problem if the number of tags assigned to it are go beyond a limit. This will happen due to the fact that the reader has to maintain the

database of all the tags assigned to it.

2.5.1.6 Simultaneous tag authentication scheme

Hyung-Joo Kim and Moon Seog Jun [?] proposed a lightweight mutual authentication protocol in 2010 for single tag, double tag and multiple tags. In the single tag authentication scheme, the reader initiates by sending a random number. The tag generates another random number and two session keys, and it uses these information to generate an intermediate information. It then sends this intermediate information to the reader along with the random number it had generated and a pointer value which acts as the index in the backend server. The reader forwards these information to the backend server along with the random number it had generated earlier. The backend server generates a third session key and uses it to generate another intermediate information and sends it to the tag via reader. The tag then verifies the validity of this intermediate information and generates fourth session key. The tag then sends the final authentication information to the backend server via the reader. The backend server finally verifies the validation of the final authentication information. In another scheme, two tags are authenticated simultaneously. Actually they extend the authentication process for single tag which is applicable to authenticate two tags in such a way that the overall computation due to the random number generation operation is less. They use the information generated by one tag as a random nonce for the other tag. In third scheme, more than two tags are authenticated in the similar manner. However, the tag which was accessed first is authenticated after completion of authentication of all the other tags. The reuse technique decreases the computation overhead. We found a few drawbacks in this scheme. For single tag authentication, any attack can be detected at the end of authentication process and hence there is some unnecessary computations. For double and and multiple authentication scheme, the tags are dependent on each other and the first tag will be authenticated at the end of the authentication process.

2.5.1.7 Elliptic curve cryptography(ECC) based authentication schemes

The authentication protocols such as the Schnorr protocol [?] and the Okamoto protocol [?] uses elliptic curve cryptography. The security of these schemes are based on the hardness of the Elliptic curve discrete logarithmic problem (ECDLP). However, both the protocols are vulnerable to tracking attack since the attacker can derive the prover's public key from the authentication information. Therefore, these conventional protocols are not applicable to RFID system [?] [?]. Batina et al. [?] presents an based Elliptic Curve Cryptography (ECC) authentication scheme. There are two algorithms for authentication. These are on-line authentication and off-line authentication. In the on-line authentication protocol, the tag replies with ID on request from the

reader. The ID is used to identify the tag and the reader replies with a challenge to tag. This protocol suffers from the problem of tracking attack during the time between two successful sessions.

Lee et al. proposed an ECDLP based Randomized Access Control (EC-RAC) authentication protocol in [**?**]. In this protocol, instead of publicly announcing the public key of the prover tag, it is kept secret in the server. During the authentication process, the prover tag generates a random number and an authentication information. It generates this authentication information using the public key and the private key kept in its memory. The server verifies the validity using the public key and the random numbers. This protocol is claimed to be a provably secured protocol and it is not vulnerable to tracking attack. However, a tracking attack is reported in [**?**] and [**?**] where an adversary can use the authentication information communicated during multiple sessions and can track the tag. Therefore, Lee et al. revised their scheme in [**?**] and proposed three different components, namely, ID-transfer, Password transfer and server authentication. They combined theses components to form the revised version of the EC-RAC protocol. However, this protocol also suffers from the tracking attack by a wide attacker [**?**]. Here wide implies that the attacker can access the result of the verification in the server i.e accept or reject. Therefore, the authors have revised their protocol in [**?**] to prevent the tracking attack from the wide attacker. However, this protocol needs one more ECC multiplication operation.

Gyozo et al,. proposed an elliptic curve cryptography based authentication protocol in [**?**]. In this protocol, the reader generates a random number and signs it using its private key to generate a signature. The tag verifies this signature using the public key of the reader and sends two pair of information to the backend server via the reader. One pair is generated using a function, called Tag_affording function and the other pair is generated using the simplified EC ELGamal encrypting scheme. The backend server verifies these information and authenticates the tag. This scheme suffers from the denial of service attack and does not fulfill the forward and backward secrecy requirement.

In the authentication scheme proposed by Guo-Rui Li et al. in [**?**], the reader initiates the communication by sending a random number. The tag responds with the authentication information along with the encrypted identifier (id). The id is encrypted using the public key of the server and then re-encrypted using the shared key, and two random numbers. The server verifies and sends the update information to the tag via the reader. The tag verifies the validity of the update information and then updates its memory after successful verification. This scheme suffers from excessive computations due to the use of public key cryptography.

2.5.2 Existing object searching schemes

Object searching is a very important problem since the desired object needs to be searched efficiently preserving the possible security and resource requirements. However, the literature has not focused this problem adequately. A few tag searching schemes have been suggested in the literature.

Tan et al. proposed four tag searching schemes in [**?**]. In their basic scheme, the reader broadcasts the desired tag information. The tag verifies this information and responds. This scheme is vulnerable to tracking attack since the adversary may use the encrypted response to track the object. In the first improvement, the tag keeps a list of previously used random numbers. The reader uses a new random number in the request message. If the tag receives any previously used random number, it ignores the request. Otherwise, the tag accepts the request and adds the new random number into the list. This solution is not scalable since the tag has to manage the list of random numbers which will increase in each session. The other problem is that the adversary obtains the idea about the existence of the desired object. This modification still suffers from the tracking attack [**?**]. In another improvement, the tags are allowed to reply revealing a few identifier bits which has a proper structure. The reader distinguishes the desired tag using the revealed bits. Since the tag identifier bits are disclosed to the adversary, he can track the corresponding object using this information [**?**]. Finally, they have another improvement where all the tags other than the desired one will reply with certain probability. In this improvement, they have used two hash functions in both the reader side and the tag side and this will make the authentication process inefficient. This protocol also suffers from ID disclosure attack [**?**].

Ahmed et al. proposed three server-less tag searching protocol in [**?**]. In their first protocol, the reader broadcasts a random number and the tags respond with authentication information. This authentication information has computed from the seed value kept in the tag memory. The reader verifies all the replies and if finds any such reply which is equal to the authentication information of the desired tag, it declares that the desired object is found. In this protocol, the reader has to process all the responses and hence it has to process $\frac{n}{2}$ number of responses in average, where n is the number of tags in the coverage area of the reader. The tag does not verify the query from the reader before replying and uses the same seed value to multiple query from the reader, and hence can be traced by the adversary. To avoid the tracking attack, the authors have modified their scheme and proposed the second protocol. In this protocol, the seed value is updated in both the tag and the reader. However, the reader has to process and update $\frac{n}{2}$ number of seeds which is a burden for the reader. Moreover, the adversary can desynchronize the reader and the tag by quering multiple times. Finally, they proposed their third protocol in which the

tag responds after verifying the authentication information of the reader. The desired tag replies with the authentication information and updates the seed value it kept in its memory and the undesired tags reply with certain probability using a random information to make the adversary fool about the existence of the desired tag. Finally, th reader verifies the desired tag information and declares about the search result. In this scheme, the reader does not need to process the all the tag information in its coverage area. It only processes the response from the desired tag and a few undesired tags. However, this scheme still suffers from de-synchronization attack since the attacker may block the response of the desired tag and the tag has already updated its seed value while the reader does not updated the corresponding seed value.

Kulseng et al. proposed three algorithms based on Linear Feedback Shift Register (LFSR) and Physically Unclonable Function (PUF) in [**?**]. In their first protocol, the reader broadcasts a search query with encrypted tag information. Though the information are encrypted, the adversary can know that the desired tag is present i.e Information leakage attack is present in this scheme. This scheme also suffers from the synchronization problem. In the first improvement, they try to remove the synchronization problem by keeping the old information. However, this solution also suffers from the vulnerability of Information leakage attack. In the third solution, they allow the undesired tags to reply with certain probability using fake information to avoid the vulnerability of Information leakage attack. The algorithms proposed in [**?**] suffer from high computation overhead due to the use of LFSR and PUF operations.

Hoque et al. [**?**] proposed the S-search protocol for finding a tag. In their scheme, the server sends a frame length and a random number to the reader. The reader generates the encrypted tag information and broadcasts it along with the slot information for the tags. Each tag in the coverage area of the reader computes the slot for itself and verifies the tag information. The desired tag generates the authentication information and responds along with the slot information to the reader. The undesired tags respond with a certain probability using fake information along with the slot assigned to them. The reader generates a bit record and verifies the presence of the desired tag and then sends this to the server. The server investigates the bit record and confirms about the existence of the desired tag. In this scheme, the computation is high due to the hash operations in both the reader and the tag which makes this scheme less efficient.

Lee et al. proposed an ECC based tag searching protocol in [**?**]. In this scheme, the tag and the server maintain a counter value to avoid any replay attack. The tag modifies this counter value after the successful verification of a query from the server. However, the authors recommend various modification strategy in the server side. During the searching process, the server broadcasts the desired tag information along with the counter value. The desired tag information is claimed to be the privacy preserved information and hence the tag cannot be tracked by a wide

attacker. This information is generated using two ECC multiplication operation. The tag first verifies the counter value and then verifies the tag information. The desired tag then responds with the authentication information. This part of the protocol is similar to the Pwd-transfer scheme described earlier in the same paper during the description of the authentication protocol. The security of this protocol is based on the hardness of the elliptic curve digital signature algorithm (ECDSA). However, this scheme suffers from the Information leakage attack since only the desired tag responds to a query from the server and hence the attacker can know the existence of the desired tag. In addition to this, this scheme does not satisfy the forward secrecy criteria since the secret key of the tag keep intact during the entire life of the tag. Due to the use of ECC multiplication, this scheme is inefficient.

Yoon et al. proposed a tag searching scheme in [**?**] where a trusted verifier and the tag maintain a counter value. The verifier sends the encrypted tag information to the reader along with the counter value. The reader broadcasts this information. The desired tag verifies the tag information and then checks the counter value. If all are verified successfully, the tag updates its counter value and generates an authentication information. Then the desired tag responds with authentication information to the verifier through the reader. The verifier verifies this response and reports about the search result. In this scheme, the adversary can know that the desired tag is present if he finds any response. Therefore, this protocol suffers from the Information leakage attack. Another problem is that they have used two hash operations in the tag which can make the protocol inefficient.

Zheng et al. proposed a two-phase Compact Approximator based Tag Searching protocol (CATS) using bloom filter in [**?**]. In this scheme, multiple number of tags can be searched in the same query. In the first phase of their scheme, the reader encodes the ids of the wanted tags into a bloom filter of a certain length which carries the membership information of the desired tags. The reader then broadcasts this filter along with a few parameters. A tag within the coverage area of the reader verifies this membership information. If the tag finds itself as a non-member tag, it keeps silent itself for the rest of the time. However, due to the false positive property of the bloom filter, a few undesired tags can be selected along with the desired tags. Thus, a set of tags selected in the first phase including the desired tags i.e. the candidate tags are ready to cooperate further. In the second phase, the candidate tags which were selected in the first phase respond to a query from the reader. The reader forwards these responses to the server. The server then constructs a virtual bloom filter and filters the desired tags. In this scheme, there is a probability that a number of undesired tags could be selected even after the second round of filtration due to the false positive property of the bloom filter. The computation in tag is also high due to multiple number of hash operations. Min et al. criticized CATS in [**?**]. According to Min et al., CATS

does not work when the size of the wanted tag set is much higher than the number of tags in the coverage area of the reader. This scheme also inefficient when the false positive ration is high.

Min et al. [**?**] proposed an iterative tag searching protocol (ITSP). They divide the bloom filter into a number of segments. The number of segments is equal to the number of hash function used. Instead of sending the whole bloom filter at a time, it sends each segment iteratively. The segment length in this scheme is dynamic and this is achieved due to the fact that the number of candidate tags in $(i+1)^{th}$ iteration is less than i^{th} iteration. Therefore, the length of the segment sent in $(i+1)^{th}$ iteration is less than $(i)^{th}$ iteration. The iteration continues until a certain condition satisfies. They showed that this scheme is efficient compared to CATS [**?**]. However, this scheme also suffers from false positive property. Moreover, the number of iteration is equal to the number of hash functions and hence the number of interactions between the tags and the reader is high.

2.5.3 Existing coexistence proof generation scheme

There are some coexistence proof generation protocols using RFID technology. We revisit them and try to find the implications they suffer during the proof generation process.

Juels had formulated the problem of yoking proof in [**?**]. In his protocol, two tags Tag A and Tag B contain their corresponding ids. They also contain a counter value. Tag A generates a parameter using a hash operation on its secret key and the counter value. It then sends this computed value along with the id and the counter value to the reader in response to the *Left proof* request. The reader forwards this to Tag B along with the request for right proof. Tag B generates a parameter using the keyed hash function (MAC) operation over the value it received from Tag A and the counter value it kept. It then sends this parameter along with its id and the counter value to Tag A via the reader, and then increases its counter. Tag A then generates another parameter using the MAC operation over the parameter it received from Tag B and the parameter it sent to Tag B earlier. It then sends this newly generated parameter to the reader and increases its counter value. The reader then generates the proof using the ids of two tags and the parameters it received from both the tags. It then submits this proof to the verifier. The verifier verifies the proof for its validity. This scheme is very simple. However, it is vulnerable to many attacks. The adversary may reuse the stored information to generate a valid proof [**?**]. He can block and capture the response from Tag B during a valid session. Therefore, Tag A cannot increase its counter value. Some other time, when Tag B is not present, the adversary can request for left proof to Tag A and after obtaining the response from the Tag A, it can wait for sometime and send the previously captured information to Tag A. Hence, Tag A can respond with a valid information to the adversary. Thus the adversary can generate a valid proof in absence of Tag B.

The adversary also can track the tags by observing the ids. The undesired tags may participate which can cause the denial of proof attack. In addition to this, the adversary may relay the request to the correct tag and hence a valid proof can be generated although the desired tag is not within the coexistence range.

Saito and Sakurai proposed a yoking proof protocol in [**?**] to prevent the replay attack which suffered by Juel's protocol [**?**]. In this protocol, the reader obtains a time stamp information from the server and broadcasts this information. Tag A then responds with a parameter which was generated by applying a keyed hash function (MAC) over the received time stamp information. Tag B received the response from Tag A through the reader and generates a parameter applying MAC operation over the received information and the time stamp information it received from the reader. Tag B then submits this parameter to the verifier through the reader. Though this scheme uses time stamp to prevent the replay attack, the adversary can predict the time stamp and broadcasts. The tags will respond and the adversary will collect these responses. Some other time when the original reader will request, the adversary can replay the stored information in absence of either tag A or Tag B or both. Therefore the adversary can generate a valid proof in absence of any of the tags [**?**]. The authors extended their protocol in the same paper to incorporate more than one tags. They introduce two kind of tags, namely product tag and pallet tag. The product tags usually attached to the objects and the pallet tag is usually attached to the container containing the objects. In this protocol, the reader obtains a time stamp information from the verifier and broadcasts. The product tag responds with a computed information after applying MAC operation over the time stamp information they received from the reader. The reader forwards these responses to the pallet tag. The pallet tag encrypts these responses and submits to the verifier through the reader. The extended scheme also suffers from the attack mentioned in [**?**]. This scheme also suffer from the denial of proof attack, relay attack, etc.

Piramuthu [**?**] had modified the scheme proposed by Saito and Sakurai [**?**] to ensure the dependency between the tags in order to generate a valid proof and hence the proof cannot be generated in absence of any of the desired tag. In this scheme, a random number is introduced instead of any time stamp. The reader authenticated itself to the verifier and obtains a random number. The tag generates a parameter after applying MAC operation over this random number. Another tag receives this computed information through the reader and computes a parameter applying the MAC operation over the received information and the random number it received earlier from the reader. It then submits this newly generated parameter to the reader and the reader prepares the proof and submits the proof to the verifier. Use of fresh random number can help to prevent the replay attack since this random number is embedded within the proof. Hence the replayed proof cannot be verified successfully due to this fresh random number. However,

if the adversary sends the same random number, the tags will respond with same information which will cause the tracking attack. The secret key of the tags are not updated and hence this scheme is unable to fulfill the forward secrecy requirement. The other attacks it suffers are denial of proof attack, relay attack etc. This scheme is also vulnerable to multi-proof session attack as mention in [**?**] and race condition as described in [**?**].

Bolotonny and Rubin have proposed an anonymous proof generation protocol for more than two tags in [**?**]. In their scheme each tag T_i generates a random number r_i and a value $a_i = f_{x_i}(r_i, a_{i-1})$. Here x_i is the secret key of tag T_i. The second parameter of function f is the response from the previous tag in the sequence. For tag T_1, it is set to 0. Thus all the tags compute a_i in a chain. This chain is terminated in tag T_1, when reader sends a_n to it as a final request. T_1 then responds to the reader with m, i.e. the MAC value of a_1, a_n, and x_1. The reader then generates and stores the proof $P_{1,2,3,...,n} = (r_1, r_2, ..., r_n, m)$. In this scheme, the undesired tags may participate in the proof generation process which can cause the denial of proof attack since the responses from the tags are not verified for the legitimacy. The adversary also can mount relay attack. In this protocol, the timer operation is assumed as the discharging rate of the underlying capacitor. However, this capacitor may recharge through other source of radio signal which can introduce noise in timer operation.

Peris-Lopez et al. proposed a coexistence proof generation protocol, namely, clumping proof in [**?**]. In this scheme, the reader obtains the encrypted time stamp information from the verifier and divides this information into two parts. It then sends the MSB part to Tag A as the request message. Tag A responds with an intermediate information after applying a MAC operation over the counter and the received information, and then increments its counter. The reader forwards this response along with the LSB part of the encrypted time stamp information to Tag B. After receiving, Tag B computes an intermediate information by applying MAC operation over the received information and the counter value it kept in its memory. It then sends the computed value to Tag A via the reader. Tag A then computes another parameter by applying MAC operation over the received information and another information it generated earlier in the same session. It then responds with this information. Finally the reader prepares and submits the proof to the verifier. This protocol suffers from de-synchronization attack. Because the adversary can send the same time stamp information multiple time and the tags will modify their counters which will not be reflected in the verifier. This scheme also suffers from the denial of proof attack.

Chih-Chung Lin et al. proposed a coexistence proof generation protocol [**?**] for tow tags, namely *sec-TS-proof* and another protocol for more than two tag, namely *chaining proof*. The later protocol assumes that the verifier can be offline during the proof generation process. In

sec-TS-proof, the authors actually modified the scheme proposed in [**?**] where the online verifier provides an encrypted timestamp information which includes a random nonce. They use this information to avoid the race condition where more than one reader are accessing the same tag simultaneously. This information also helps to avoid the relay attack. However, this protocol is vulnerable to tracking attack through the tag ids and it is unable to provide forward and backward secrecy requirement. The authors proposed *chaining proof* to incorporate more than two tags. In this protocol, the reader consists of an additional module, namely, timestamp database (TSD). The TSD provides a random number to the reader and combines this random number with a timestamp value. the reader sends a request information to tag T_1 which consists of this random number. T_1 responds with its id and a partial proof information. The reader combines this response with the request information it sent to T_1 and submits this value to TSD. It also computes a hash value out of the combined information and sends this hashed value to tag T_2. Meanwhile, the TSD keeps the combined information received from the reader along with a timestamp information. Tag T_2 responds with its id and a partial proof information. In similar fashion, the reader generates the request information for tag T_i using the response from tag T_{i-1}. Thus, TSD keeps the responses of all the tags along with the corresponding timestamp information. Finally, the reader generates the proof and submits this proof to the verifier when it comes to the coverage area of the reader. This protocol also vulnerable to the attacks similar to *sec-TS-proof*.

Burmester et al, proposed an anonymous grouping proof with forward secrecy for two tags in [**?**]. In this protocol, the tag keeps two sets of security parameters, namely, current (cur) and old to avoid the de-synchronization attack. During the proof generation process, the reader broadcasts a random value to the Tag A and Tag B. Each tag then computes two intermediate information using the information from cur and old. The tags then respond with some part of both the computed information. The reader verifies the responses and links the tags through the channel in the data link layer. After creation of this link, the Tag B computes another intermediate information using the information kept in either the set cur or old. The use of cur or old depends on which set match with the link request from the reader. Tag B then responds with this information and updates the parameters in cur and old, and stops. Tag A receives partial response of Tag B from the reader and computes an information in a similar manner to Tag B. It then submits this information to the reader. The reader generates the proof and submits the proof to the verifier. This scheme does not suffer from the denial of proof attack since the tag information are verified before the creation of the link through the channel in the data link layer. However, this protocol is vulnerable to tracking attack. Because the tag will reply with same information in response to the same random number as request message in multiple times from

an adversary and this can happen during the time between two successful sessions. Therefore, the tags are traceable during this time period. This protocol is also unable to provide forward secrecy requirement. Because if a tag is compromised for one session, the adversary can compute the group key and the random number of all the tags for the later sessions and he also can generate the secret key of the compromised tag in the later sessions.

Another scheme proposed by Yao et al. in [**?**]. In their scheme, a tag sends the pseudonym IDS_{tag} of its id to the reader in response to a request message. The reader then computes A_a, B_a, A_b, B_b and sends $A_a \| B_a, IDS_b$ to tag A and $A_b \| B_b, IDS_a$ to tag B. Tag A then computes $m_a (= [IDS_a + IDS_b + (ID_a + x_a)] \oplus r)$ and sends to tag B via reader. Tag B computes $m_b (= [(ID_b + x_b) + m_a] \oplus r)$ and sends it to reader. The reader then generates the proof $P_{AB} = (IDS_a, IDS_b, r, m_a, m_b)$. Here ID_{tag} is the id of a tag, x_{tag} is the secret key and r is the group random number. These parameters are updated in each successful session. This scheme suffers from traceability problem during the time between two successful sessions since the tag will respond with same *IDS* on consecutively fake "hello" message from the adversary. Denial of service attack is another problem in this scheme. This is because if the adversary blocks either m_a or m_b, tag A and/or tag B will update various parameters, whereas the verifier has not updated these parameters which will desynchronize the tags from the verifier. It also suffers from Relay attack and Denial of proof attack.

Duc and Kim [**?**] have proposed a (n, n) secret sharing scheme based proof generation protocol in 2010. In their scheme the verifier chooses a random number x and sends it to reader. The reader generates n shares of x such that if any share is missing, the original x cannot be formed. To generate these shares, the reader generates $n - 1$ random numbers and creates the n^{th} value by XORing all the random numbers with x. Thus reader generates the shares $y_i, i = 1, 2, ..., n$ and broadcasts these along with x. Each tag T_i generates m_i applying MAC on y_i, x, and its secret key K_i and then sends m_i to reader. The reader then generates and submits the proof $P = (\tau_1, y_1, m_1, \tau_2, y_2, m_2, ..., \tau_1, y_n, m_n)$. This scheme although very simple, suffers from many security flaws. The tags can be traceable since they will send same response on same x and y_i value. It also suffers from relay attack since the adversary may relay x and y_i to tag T_i which is beyond the coexistence range.

Nai-Wei Lo et al. proposed three protocols for co existence proof generation and verification in [**?**], namely, online verifier based protocol (OVBP), efficient online verifier based proocol (EOVBP) and offline timestamp server baed protocol (OTSBP). The OVBP is simplest protocol among these three protocols. In this protocol, every tag sends the authentication information to the reader in response to a query message from the reader. The online verifier uses this information to verify the legitimacy of the tag. This verification is performed to avoid the denial

of proof attack. The tag 1 is authenticated and then it receives an authentication information from the reader to verify the legitimacy of the reader. This authentication information consists of a timestamp information. After, this verification, it responds with a partial proof information. The reader forwards this information along with an authentication information to Tag 2. This communication is performed after the successful verification of the authentication information of Tag 2. Tag 2 then verifies the authentication information and responds with another partial proof information. The reader then prepares and submits the coexistence proof to the online verifier. This protocol uses two records for each tag to avoid denial of service attack. To reduce the time complexity in the online verifier, the authors have improved the scheme and proposed EOVP. In EOVP, instead of using two records for each tag, they incorporate a counter value which increases the efficiency and avoids the denial of service attack. However, this counter value can help the adversary to trace a tag. Though the authors claimed that their protocols guarantee the forward secrecy requirement, the adversary can compromise the secret key in one session and can compute the secrets in later sessions using the same PRNG function. The OVBP and EOVBP assume that the reader can communicate with online server. However, the server may not be reachable to the reader during the proof generation process. In remedy, the authors suggest an improved protocol OTSBP. In this protocol, a trusted timestamp module is introduced which is responsible to provide a timestamp value and stores the coexistence proof. This module is actually implemented in the reader and it submits the proof to the verifier when it is reachable to the verifier. All the protocols described in this paper use timestamp information to avoid the relay attack. This protocol is also suffer from the attacks similar to OVBP and EOVBP. All the protocols proposed in this paper uses three heavyweight operations (PRNG, MAC and Nun) and this will make the protocols inefficient. Moreover, if the adversary put many invalid tags, the proof generation process will be delayed for a long time.

2.5.4 Scope of further work

In this chapter, we provide a literature review on the direction of various protocols in RFID technology which adhere to various security requirements. We choose three applications based on RFID technology which need security protocols to accomplish the application requirements. The problems we emphasize are i) authentication of legitimate objects to a legitimate authority, ii) searching a legitimate object by a legitimate authority, and iii) generating and verifying a proof of coexistence of multiple number of relevant and legitimate objects by a legitimate authority. In order to perform the authentication requirement in problem i), the RFID reader (legitimate authority) broadcasts a request message and the RFID tags under coverage area of the reader respond. However, there can be invalid responses and these responses need to be

filtered. Therefore, the reader verifies the legitimacy of the valid responses with the help of a backend server. Thus the legitimate objects are authenticated. This is one way authentication. However, sometime the reader and the tags update various parameters after completion of the authentication process and the reader communicates the updated parameters to the tags. Therefore, the tags need to verify the legitimacy of the reader. This is because, an adversary may force the tags to modify their parameters which will affect the synchronization between the legitimate reader and the tags. This kind of authentication is two way authentication. In literature, we found many authentication protocols of different families which are classified according to the operations they used. In object searching problem (problem ii), the reader needs to broadcast a search information for a legitimate object. The tag attached to the desired object needs to verify the legitimacy of this request to avoid any non legitimate activity and responses to this request. The reader needs to verify this response with the help of the backend server. Authentication process alone could solve this problem. However, it will be inefficient. Therefore a few protocols have been proposed in the literature which employed separate approaches. Though the researchers use separate approaches to solve this problem, they keep the verification of the legitimacy of various entities in their protocols. In this problem, a non legitimate entity can conclude about the existence of the desired object by observing that whether there is any response at all. In order to avoid this kind of information leakage, most of the existing protocols suggest that the undesired tags also respond with certain probability using random information. Problem iii) i.e the proof generation and verification problem also needs to verify the legitimacy of various components. In this problem, the reader broadcasts a proof request and the relevant tags engaged themselves in order to generate a valid proof. This proof usually generated in a chain of relevant tags which needs a inter-tag communication. However, passive tags cannot communicate with each other and hence the reader acts as an inter-mediator in order to generate the proof. The reader submits this proof to the verifier and the verifier verifies this proof as when necessary. We found a few protocols in the literature which solve this problem considering the whole set of complications in this problem. In the literature, we found that the components involved in the existing protocols which solve the problem i) and ii) are mainly the objects tagged with RFID tag and the RFID reader. Another component involved in these protocols is backend server which is used to hold the information about the objects in concern. However, this component is optional since the reader itself can keep the information of the concerned objects. Therefore a few protocols assumed serverless environment. The protocols in problem iii) use the backend server as a verifier or they keep a separate verifier in addition the backend server.

In the literature, we found that many authentication schemes such as [**?**] [**?**] [**?**] [**?**] suffers from the tracking attack during the time between two successful sessions where the attacker

obtains same response from the tag in response to multiple request messages. Many other existing authentication protocols have designed in such a way that they do not have this problem. However, there are no such existing solution which can fix this problem and applicable over the suffered authentication schemes. Therefore, there is a need of a solution which can fix the tracking problem keeping the other parts of an authentication scheme intact. We propose a solution to this problem in chapter 3, which can be applicable over any authentication scheme that suffered from the tracking attack during the time between two successful sessions. The authentication scheme in [**?**] is also vulnerable to de-synchronization attack. We fix this problem in the same chapter.

The existing authentication schemes use single tag in each object. Therefore, the detection probability of the objects in these schemes is less due to the fact that the position at which the tag is attached to an object may not exists within the coverage area of the reader while a few other parts of the same object exist within the same coverage area. Attaching multiple number of tags in an object can improve the detection probability [**?**] and multiple tags can be utilized to enhance the security. If we use existing authentication scheme for multi-tag environment, the single set of security related information needs to be replicated to all the tags attached to the object. In this arrangement, if any tag is compromised by the adversary, all the tags will be compromised. Moreover, as we have seen in the literature, most of the authentication schemes are vulnerable to many attacks. Some authentication schemes uses heavy weight operations such as ECC multiplication, hash operation, etc. which are not applicable to RFID technology. Therefore, there is a need of separate lightweight authentication protocol in multi-tag arrangement which can utilize multiple number of RFID tags in order to enhance the security and detection probability of the objects. We propose a lightweight authentication protocol in chapter 4 which utilizes the multiple number tags to increase the security during the authentication process. Our proposed protocol uses very basic operation yet provides the possible security requirements. However, this protocol considers an increased traffic between the reader and the objects due to the fact that multiple number tags attached to every object responds to a request from the reader. Therefore, we propose another authentication protocol in the same chapter where the traffic congestion between the reader and the objects is similar to the traffic in existing authentication schemes. Moreover, this scheme provides the benefits of multi-tag arrangement. However, this scheme requires to use active tag as the tag entity.

The existing object searching schemes and coexistence proof generation schemes as we found in the literature use single tag for each object and these schemes also suffer from the detection probability problem. In similar argument, these existing schemes also cannot be applicable to multi-tag arrangement with enhanced security benefits. Therefore, the literature needs

the protocols to solve the object searching problem and coexistence proof generation problem in multi-tag arrangement. We propose a lightweight and secure object searching scheme in chapter 5 and a lightweight and secure coexistence proof generation and validation scheme in chapter 6.

Chapter 3

Approaches to Solve Location Privacy and Message Blocking Problems

The RFID tag can be embedded with the essential commodities such as passport, identity card etc and user can carry these items. Any adversary can access the RFID tag using a RFID reader without the knowledge of the user and gain private information of the user. If the information emanates from the RFID tag is encrypted and the adversary does not have the required secret key to decrypt, we can assume that he cannot be able to gain any private information of the user. However, the adversary can passively gain the location characteristics of the user. In order to achieve this, the adversary can guess a few locations and put RFID readers in these locations. Suppose, the user may carry the passport, identity card etc. and visit these locations regularly. The responses from the tag kept with the user will be read by the reader kept in various locations and adversary can obtain a location pattern of the user. Subsequently, the adversary can gain the location characteristics of the user. Therefore, location privacy is a great concern in any application which uses RFID technology.

In order to preserve the location privacy of the user, the RFID tag needs to prevent from any unauthorized reading and there are various techniques which accomplish this task. For example, the RFID tag can be kept in a Faraday cage which can prevent any kind of radiation to reach into the RFID tag. Hence the adversary is unable to obtain any response from the tag without the knowledge of the user. Blocker tag is another example where the tag will be destroyed after the initial reading of the tag. However, the blocker tag is useful only in those applications which requires to use the tag once and there is no need to use the tag in future. Jamming also can be used to prevent the penetration of the signal from any unwanted reader. These solutions perhaps can keep the privacy of the user. However, the user has to reactivate manually in order to activate a legitimate tag access.

Various authentication schemes using RFID technology have been proposed which tried to preserve the location privacy of the user. However, many protocols such as [**?**] [**?**] [**?**] [**?**] [**?**] cannot prevent attack against the location privacy during the time between two successful sessions. In these protocols, the security information are updated in a successful session in such a way that the responses in two successful sessions cannot be matched to conclude about a particular tag. However, during the time between two successful sessions, the tag can respond with same information in response to multiple requests from an adversary. Hence the tag can be traced during this time. If this duration is long then this attack is a serious problem. For example, a student may not visit the library frequently and he/she visit only once in every semester. However he/she always keeps the identity card with him/her which embedded with the RFID tag.

We have tried to solve the tracking problem between two successful sessions that exists in various existing authentication schemes. Our solution can be appended over an authentication scheme to overcome the tracking problem between two successful sessions.

We also found a de-synchronization attack in the authentication scheme proposed by Khor et al. and suggest a solution to that.

3.1 Proposed solution for solving the tracking problem during the time between two successful sessions

The solution explained in this section can be applied on the top of any existing authentication scheme that suffers from the tracking of RFID attached objects during the time between two successive and successful sessions. For the sake of understanding the effectiveness of the proposed solution, we explain the proposed solution by applying it over the fingerprint based authentication protocol [**?**] proposed by Khor et al. Therefore, we are first briefly introduction the fingerprint based authentication protocol and then the implementation of the proposed solution over it.

3.1.1 Fingerprint Based Authentication Protocol [?]

In the fingerprint based authentication scheme [**?**], the backend server initially stores five values $\text{CRC}(\text{EPC}_T \| FP) \oplus K_i$, $\text{CRC}(\text{EPC}_T \| FP) \oplus K_i$, EPC_T, FP, DATA of each tag in its database. Here $\text{CRC}(\text{EPC}_T \| FP) \oplus K_i$ is denoted as an index of the record corresponding to a tag. The session key of the tag is denoted by K_i and the electronic product code for the tag is denoted by EPC_T. FP denotes an unique electronic fingerprint of the tag. There is also an extra information related to the tag and that information is denoted as DATA. On the other hand, three values that are stored in the tag are K_i, EPC_T and FP. Session key of the current session is denoted

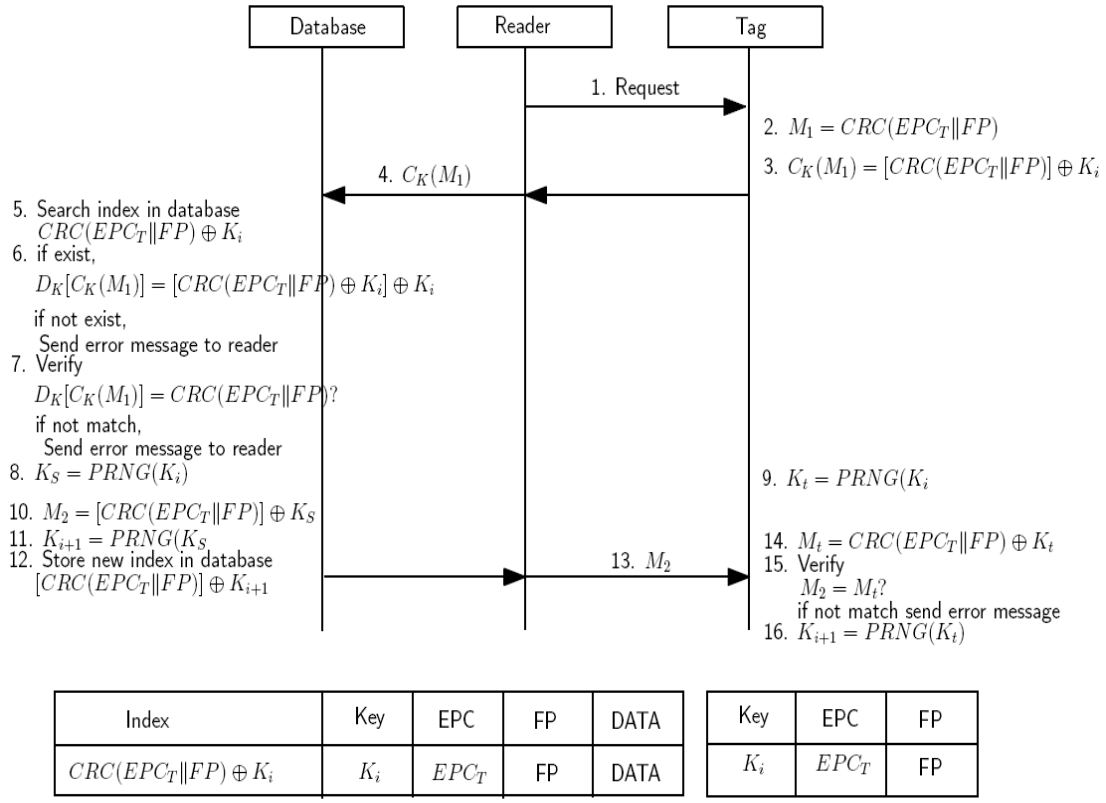

Figure 3.1: Fingerprint-based mutual authentication protocol [**?**]

as K_i. Figure 3.1 shows the steps in the authentication scheme proposed in [**?**]. The steps are summarized below.

Step 1: Reader broadcasts a **request** message.

Step 2: The tag receives the request message and computes

$$M_1 \leftarrow \text{CRC}(\text{EPC}_T \| \text{FP}).$$

Step 3: M_1 is encrypted as

$$C_k(M_1) \leftarrow [\text{CRC}(\text{EPC}_T \| \text{FP})] \oplus K_i.$$

Step 4: Tag sends $C_k(M_1)$ to the backend server via the reader.

Step 5: The backend server searches for an index, $\text{CRC}(\text{EPC}_T \| \text{FP}) \oplus K_i$ in its database.

Step 6: If matching index is found, then the backend server decrypts it using the session key, K_i that is in the same row as indicated by index for verification. Otherwise, the server sends an error message to the reader.

Step 7: If the decrypted message does not match, an error message is sent to the reader. Otherwise the tag is authenticated.

Step 8: Backend server computes

$$K_s \leftarrow \mathit{PRNG}(K_i).$$

Step 9: Meanwhile the tag has evaluated a temporary key

$$K_t \leftarrow \mathit{PRNG}(K_i)$$

Step 10: Backend server computes

$$M_2 \leftarrow \text{CRC}(\text{EPC}_T \| \text{FP}) \oplus K_S$$

Step 11: The backend server generates a new session key, K_{i+1}.

Step 12: The backend server stores a new index $[\text{CRC}(\text{EPC}_T \| \text{FP})] \oplus K_i$ in the database.

Step 13: The backend server forwards M_2 to tag through the reader.

Step 14: The tag computes

$$M_t \leftarrow \text{CRC}(\text{EPC}_T \| \text{FP}) \oplus K_t.$$

Step 15: The tag verifies the authentication of the reader by comparing M_2 and M_t. If it does not match then it sends an error message.

Step 16: Otherwise, the tag updates the session key, K_{i+1}, where $K_{i+1} \leftarrow \text{PRNG}(K_t)$.

Brief description: In the authentication process, the reader initiates the communication by broadcasting a request message. On the request message from the reader, tag computes the encrypted message $C_k(M_1)$ and sends it to the backend server via the reader. After receiving,the

Backend server searches for an such an index in its database which is equals to $C_k(M_1)$. If it does not find any match it sends an error message to the reader and stops. Otherwise, it decrypts CRC(EPC$_T$‖FP) using the session key K_i kept in the same row as indicated by the searched index for verification and checks the validation of CRC(EPC$_T$‖FP). It sends an error message and stops if CRC(EPC$_T$‖FP) is invalid. Otherwise, the tag is authenticated and the backend server computes K_S by applying PRNG operation on K_i and then computes M_2 using this K_S. It then sends M_2 to tag via the reader. Meanwhile, the tag has evaluated a temporary key K_t and computed M_t. It then compares this M_t with the received M_2. If there is no match, it sends an error message. Otherwise, it authenticates the backend server. After successful completion of this session, both tag and the backend server updates their session key K_i. Therefore, it ensures that the next response from the tag will not be the same. However, if the session is made after a long time, then tracking can be made on the basis of the message $C_k(M_1)$. This is because the session key will not be updated till any successful session takes place.

3.1.2 Proposed solution to preserve the location privacy

After a thorough study of a number of authentication schemes [**?**] [**?**] [**?**] [**?**] [**?**], we found a common problem that exists in most of the existing authentication schemes and the problem is that an object attached with RFID tag is traceable. Some of the schemes [**?**] [**?**] have resolved this problem. However, the object is still traceable during the time between two successive and successful sessions. Our solution can help these authentication schemes to prevent the tracking attack. Figure 3.2 is a pictorial representation of our proposed solution.

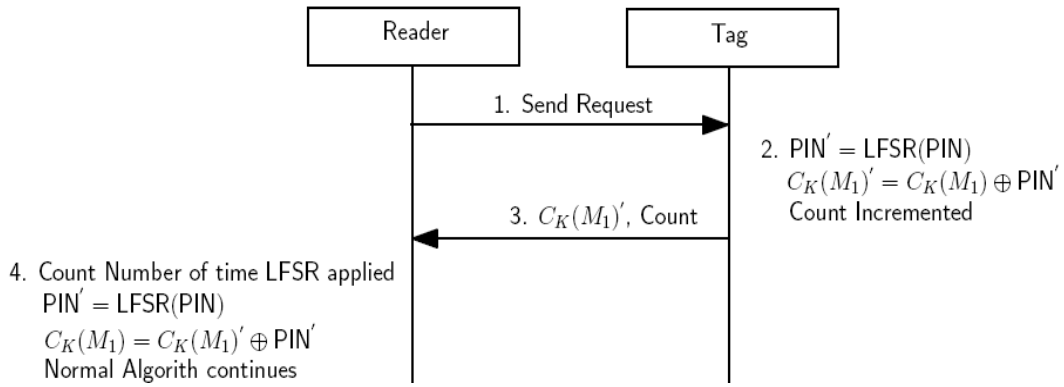

Figure 3.2: Proposed solution applied over the explained scheme in section 3.1.1

We have a few assumptions for our proposed solution as follows:

1) We use a publicly known LFSR (Linear Feedback Shift Register) function.

2) The environment consists of many tags.

3) PIN is a secret which is same for all the tags.

A few parameters need to be added and initialized in order to execute the proposed solution over an existing authentication schemes. The following are the initialization steps requires in the proposed solution:

a) The tags and the reader are pre-installed with a secret value known as PIN.

b) The tag maintains a variable COUNT for counting the number of LFSR operation applied to PIN and it is initialized to zero.

c) The tag and the reader maintains another variable PIN$'$ for storing the resultant value after applying the LFSR operation.

d) Initially the value of PIN$'$ is equals to the value of PIN.

During the authentication process of the existing authentication protocol, a few additional steps need to be executed in order to adopt the proposed solution. The following are the steps of the proposed solution which can be applied over the authentication scheme in [**?**]:

Step 1: The reader sends a request message to tag.

Step 2: The tag updates the PIN$^{'}$ and increments the counter value as

$$\text{PIN}' \leftarrow \text{LFSR}(\text{PIN}')$$
$$\text{COUNT} \leftarrow \text{COUNT} + 1$$

Step 3: The tag computes

$$C_k(M_1)^{'} \leftarrow C_k(M_1) \oplus \text{PIN}^{'}$$

Step 4: The tag sends $C_k(M_1)^{'}$ and COUNT to the reader.

Step 5: The reader obtains PIN$^{'}$ as

$$\text{PIN}^{'} \leftarrow \text{PIN}$$
For i=1 to COUNT
$$\quad \text{PIN}^{'} \leftarrow \text{LSFR}(\text{PIN}^{'})$$

Step 6: The reader extracts the message

$$C_k(M_1) \leftarrow C_k(M_1)' \oplus \text{PIN}'$$

and checks the authentication of tag as described in section 3.1.1.

Brief description: In the above solution, the reader first sends a request message to tag. Then the tag applies the LFSR operation on PIN$'$ and increments the COUNT. It then XORs the original message of the existing authentication protocol with PIN$'$ and sends the resultant information with COUNT value to the reader. The reader uses the COUNT value to find the number of time it has to apply the LFSR operation over the PIN to obtain the desired PIN$'$ value. After that the reader XORs the message obtained from the tag with its PIN$'$ value to obtain the original message. Then the reader verifies the authentication of the tag as described in section 3.1.1. When COUNT reaches to its maximum value, the COUNT is reset to 0.

The proposed modification solves the tracking problem between two successful sessions. Each time the reader sends the request message to tag, the tag updates its counter value and finds a new PIN$'$ to be used for XORing with the original message. Therefore, each response from the tag will be unique. Again, as per our assumption, the environment consists of many tags. Therefore, the tracking based on the COUNT value will be difficult. However, if any PIN$'$ is compromised in a particular session, the successive PIN$'$ can be computed using the PRNG function by the adversary. Therefore, there is no forward secrecy guarantee of the PIN$'$ in the proposed solution.

The process of applying the LFSR operation by the reader can be optimized by using a lookup table to store some of the most frequent COUNT values and corresponding PIN$'$ values. Thus instead of applying LFSR operation repeatedly COUNT number of times, the reader can directly obtain the PIN$'$ value from this lookup table. Therefore, the delay due to the computation of the PIN$'$ can be reduced and consequently the reader can be efficient.

3.2 Implementation details

The proposed solution for solving the tracking problem during the time between two successful sessions uses the LFSR function and a variable COUNT to store the counter value. Therefore, a LFSR function needs to be implemented on the RFID tag and the COUNT variable needs to be stored in the non-volatile memory of the tag. Passive tags (non-battery) typically have 64 bits to 1 kilobyte of non-volatile memory while the active tags may have memories as high as 128 kilobytes. The counter can be implemented using flip-flop circuits. If we use 16 flip-flops, we can get value of the COUNT variable ranging from 0 to $2^{16} - 1$. Therefore, in order to store the

COUNT variable in the tag memory, the required memory is 16 bits. This can be easily adopted in passive tags and active tags.

Again the counter can be implemented using D flip-flop, where a D flip-flop requires 4 NAND gates and one NOT gate. Therefore, the implementation of the single D flip-flop requires 9 gates. For 16-bit counter, 16 D flip flops are needed. Therefore, overall 144 gates are needed for its implementation. Passive RFID tags with no battery have between 200-2000 gates available for security measures [**?**]. Therefore, it can be easily implemented on RFID tags.

If the PIN is of 16 bit, then we need 16-bit LFSR. It means 16 flip-flops. So, the number of gates required for its implementation is 144. Therefore, the existing authentication schemes require 288 additional gates to adopt the proposed solution.

3.3 Message blocking problem suffered by the authentication protocol proposed by Khor et al. [?]

The fingerprint based authentication protocol [**?**] suffers from the message blocking problem. In section 3.1.1, we have already described the working of this protocol. We have seen that in step 10, the backend server computes the message M_2 and updates its index value and the session key. On receiving the message M_2 (in step 15), the tag verifies M_2 and updates its session key. However, if the message M_2 is blocked by an attacker or somehow this message is unable to reach to the tag, there can be a synchronization problem which can occur due to inconsistency between the tag and the backend server. This is because the backend server has already updated its key but the tag has not updated the key yet. We have proposed a solution to mitigate this problem.

3.3.1 Remove the message blocking problem

The proposed solution makes the following modification in the database of the backend server:

According to the protocol in [**?**], the backend server stores five attributes for every objects in its database. The index value denoted as CRC(EPC$_T$‖FP) $\oplus$ K_i, where the session key is denoted by K_i, the electronic product code for the tag is denoted as EPC$_T$, the unique electronic fingerprint is denoted by FP and any extra information related to the tag is denoted by DATA. We propose the addition of two more attributes in the record of each tag: INDEX$_{OLD}$, and $K_{i,OLD}$. Here $INDEX_{OLD}$ stores the previous index value and $K_{i,OLD}$ stores the previous session key for the tag.

A few steps of the authentication protocol in [**?**] also needs be modified in order to adopt our proposed solution. The modified steps are as follows:

1) The reader receives $C_k(M_1)$ from the tag and forwards it to the backend server.

2) The backend server searches for the index value $C_k(M_1)$ in its database.

3) If the backend server finds a match, it continues the authentication process as described in [**?**] and replaces the attributes $INDEX_{OLD}$, and $K_{i,OLD}$ with the received index value and the current session key K_i. It then updates the session key K_i and the index value by the newly generated index value and the session key.

4) Otherwise, the backend server searches the $INDEX_{OLD}$ entries in order to find a match with the received index value. If it does not find a match, the backend server sends an error message to the reader and stops.

5) If it finds a match using $INDEX_{OLD}$ entries, it means the tag has not updated its session key K_i due to the message blocking problem in previous session and the backend server has updated its session key. Therefore, the backend server continues the authentication process using the session key $K_{i,OLD}$ (i.e. previous session key) and follows the steps as described in [**?**]. It then updates the session key K_i and the index value using the newly generated index value and the session key. However, this time the backend server does not replaces $INDEX_{OLD}$, and $K_{i,OLD}$.

In the proposed solution, we keep the old index value and the old session key information of each tag along with the new index value and session key information. If the updated authentication information sent by the reader to tag is blocked by the adversary, it seems that there can be synchronization problem. However, according to the proposed solution, the old index value and old session key can help to identify the tag. Hence there is no synchronization problem. The old entries keep intact until the backend server finds that the tag is identified using the new entries. Thus the proposed solution has removed the message blocking problem in [**?**]. The additional cost to adopt our solution is to keep two additional information for each tag in the database of the backend server. Therefore, if there are n number of tags, the modified authentication protocol requires to keep $2n$ number of additional information. We assume that the backend server does not have storage limitation problem and hence the proposed solution can be adopted in the authentication protocol proposed by Khor et al. [**?**].

3.4 Conclusion

A number of RFID authentication schemes suffer from the tracking attack between two successive and successful sessions. In this chapter we proposed a solution to this problem and ensures that there can be no incident which can track an object attached with RFID tag. We have analyzed the proposed solution and found that if any PIN$'$ is compromised, the successive PIN$^{'}$ will also be compromised and hence we are not ensuring any forward secrecy of PIN$'$. We assume that the adversary is unable to compromise any PIN$^{'}$. Therefore, this chapter raises an important problem as a future work i.e. how to ensure the forward secrecy of the PIN$'$ value. In another problem, we have found that the authentication scheme proposed in [**?**] suffers from the message blocking problem and consequently there can be a synchronization problem. We have removed the synchronization problem by suggesting the use of old copy of index value and the session key of each tag in the backend server. In this chapter, we address two kind of attacks that can occur during an authentication process. There are many other attacks which can disrupt the authentication pocess. In the next chapter (Chapter 4), we address these attacks and describe our proposed authentication schemes.

Chapter 4

Managing authentication and detection probability in Multi-tag RFID system

Authentication is a necessary requirement for Radio Frequency Identification (RFID) technology so that only legitimate entities are allowed access of information. For example an adversary may send a fake id to the RFID reader and the reader may allow it to access further information about an object. Thus the adversary can misuse the information about an object. During the authentication task, the adversaries may implement many attacks as we have mentioned in earlier chapters and hence the authentication scheme needs to be secure against the possible attacks. The classical cryptography techniques cannot be used due to various resource limitations in RFID chip. Therefore, we need to use lightweight cryptography techniques which can be applicable to this kind of resource limited environment and also can provide sufficient security benefits during the authentication process.

Attachment of single tag in an object has less detection probability since the attached tag may not be within the communication range. In this situation, the object staying within the communication range cannot be identified. Use of multiple tags in an object solves this problem which increases the detection probability of an object enormously [**?**]. The tags are attached in such a way that if any part of the object is within the communication range of the reader then there is at least a tag attached to it which is within the communication range [**?**] of the reader. Thus, the use of multiple numbers of tags increases the detection probability of the object. The challenges we have encountered in this problem are i) whether multiple resources can be utilized to enhance the security, ii) whether it is possible to prevent the possible attacks during the authentication process. iii) whether it is possible to implement the authentication task amid various resource limitations of RFID technology. In this chapter, we propose two lightweight authentication schemes assuming the objects are attached with multiple numbers

of RFID tags. We conduct study on how the proposed schemes handle the possible security implications and how much resources the proposed schemes require to use.

4.1 Object authentication using RFID technology: A Multi-tag approach

In this section, we describe the first authentication scheme which assumes that every object is attached with multiple number of RFID tags. Before describing the proposed scheme we describe the *Communication Model*. Table 4.1 lists the meaning of symbols used in the following discussions.

Table 4.1: Symbols used to describe the authentication protocol 1

Symbol	Meaning
G_j	An object
m	Number of tags attached to an object
T_i	i^{th} tag in an object
IN_i	Index value
S_i	Secret key
TID_i	Tag id in tag memory
N_i	Session key in tag memory
$TID_{i,old}$	Old tag id in backend server
$TID_{i,new}$	New tag id in backend server
$N_{i,old}$	Old session key in backend server
$N_{i,new}$	New session key in backend server
U_i	Update status
v, g_i, g_i'	Random numbers
$Valid_j$	Validity information for G_j
$\oplus$	XOR operation
$+/-$	Addition/Subtraction operation

4.1.1 Communication Model

The components involved in the communication model are a set of objects, RFID reader and a trusted server called backend server. Every object is attached with m number of RFID tags in

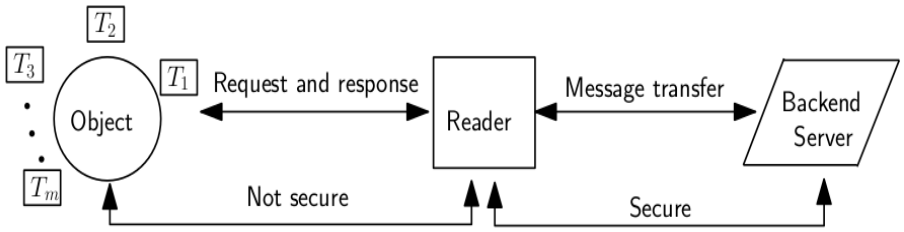

Figure 4.1: Communication model for the object authentication scheme

a process similar to [**?**]. A workstation acts as the backend server which has relatively higher storage capacity. It keeps the information of the objects. A RFID reader acts as an intermediary between the tags attached to the objects and the backend server. The communication between the reader and the backend server is wired or wireless and is assumed to be secured. On the other hand, the communication between the reader and the tags attached to the objects are wireless and is not secure. Figure 4.1 illustrates this communication model.

4.1.2 Proposed Protocol

The proposed authentication scheme has two phases, namely, Setup phase and Authentication phase. Figure 4.3 illustrates these phases. Before introducing these phases, we describe the information maintained by the components mentioned in the communication model.

4.1.2.1 Information in backend Server and tags

The tag attached to an object contains the information about the object. It contains the index value IN_i, secret key S_i, tag identifier TID_i and session key N_i.

IN_i	S_i	TID_i	N_i

The backend server contains a database to keep the information for all the objects. One record in the database contains the information about one object. This record contains a validity information $valid_j$ and the information for m number of tags attached to the corresponding object.

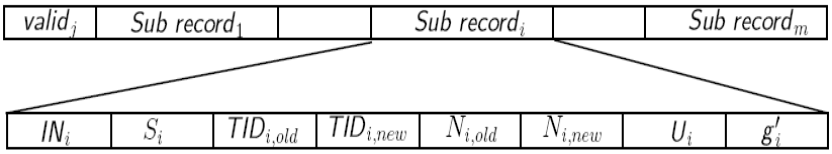

Figure 4.2: Information in backend server

The record for an object is divided into m number of sub-records. Each sub-record contains index value IN_i, secret key S_i, two tag identifiers $TID_{i,old}$, $TID_{i,new}$, two session keys $N_{i,old}$, $N_{i,new}$, update status U_i and random number information g_i'. Figure 4.2 illustrates the record for an object kept in the backend server.

Object	Reader	Backend Server
Setup phase		
IN_i, S_i, TID_i, N_i $i = 1, 2, ..., m$		$IN_i, S_i, TID_{i,old}, TID_{i,new}, N_{i,old}, N_{i,new}$ $U_i, g_i', i = 1, 2, ..., m, Valid_j \leftarrow 0$
Authentication phase		
	2. v ←	1. v ← Generates v
$K_i = (TID_i - g_i) \oplus ((v \oplus g_i) - N_i)$	3. IN_i, K_i, g_{i_1} →	4. IN_i, K_i, g_{i_1} → Verifies K_i with j^{th} record under index IN_i If verified on $TID_{i,new}, N_{i,new}$ $Valid_j \leftarrow Valid_j + 1, U_i \leftarrow 1, g_i' \leftarrow g_i$ Else if verified on $TID_{i,old}, N_{i,old}$ $Valid_j \leftarrow Valid_j + 1, U_i \leftarrow 2, g_i' \leftarrow g_i$ If $Valid_j \geq l, j = 1, 2, ..., n$ If $U_i = 1, i - 1, 2, ..., m$ Generates TID_i', N_i' randomly Generates P_{1i}, P_{2i}, P_{3i} using $TID_{i,new}$, $N_{i,new}$, TID_i', N_i', g_i', v $TID_{i,old} \leftarrow TID_{i,new}, N_{i,old} \leftarrow N_{i,new}$ $TID_{i,new} \leftarrow TID_i', N_{i,new} \leftarrow N_i'$ Else If $U_i = 2$ Generates TID_i', N_i' randomly Generates P_{1i}, P_{2i}, P_{3i} using $TID_{i,old}$, $N_{i,old}$, TID_i', N_i', g_i, v $TID_{i,new} \leftarrow TID_i', N_{i,new} \leftarrow N_i'$ Else ignores and stops.
Extracts TID_i', N_i' from P_{1i}, P_{2i} If P_{3i} is valid Updates $TID_i \leftarrow TID_i', N_i \leftarrow N_i'$	6. $IN_i, P_{1i}, P_{2i}, P_{3i}$ ←	5. $IN_i, P_{1i}, P_{2i}, P_{3i}$ ←

Figure 4.3: Proposed authentication protocol

4.1.2.2 Setup Phase

In Setup phase, the tags and the backend server are initialized and deployed for authentication process to be performed in future. We consider n objects. In the illustration, we describe the initialization process for the object $G_j, 1 \leq j \leq n$.

- G_j is assigned the tags $T_i, i = 1, 2, ..., m$. The memory of each tag T_i is loaded with an index value IN_i (Index values are unique corresponding to the tags attached to an object. However, any two or more objects have the tags with same index value.), a secret key S_i, a tag id TID_i and a session key N_i.

- The $Valid_j$ field in the record for G_j is initialized to zero. The sub-records for the tags attached to G_j are initialized as follows. The index value IN_i and secret key S_i which were loaded to the memory of the tag T_i are also loaded to the corresponding fields. The tag id TID_i which was loaded to the memory of tag T_i are also loaded to both the fields $TID_{i,old}$ and $TID_{i,new}$. Similarly, the session key N_i which was loaded to the memory of the tag T_i are also loaded to both the fields $N_{i,old}$ and $TID_{i,new}$.

Thus, after initialization process, the assigned tags are attached to the corresponding objects appropriately similar to the proces described in [**?**] and the objects are deployed.

4.1.2.3 Authentication Phase

In authentication phase, the objects are authenticated as and when required. We use separate algorithms for the components mentioned in the communication model (described in Section 4.1.1). Algorithms 1, 2, 3 are performed by the reader, the tags attached to an object and the backend server respectively.

Algorithm 1 executed by reader

1: Receives v from backend server and broadcasts.
2: Receives computed information IN_i, K_i, g_i from tags.
3: Forwards IN_i, K_i, g_i to backend server.
4: Receives computed information $IN_i, P_{1i}, P_{2i}, P_{3i}$ from backend server.
5: Forwards $IN_i, P_{1i}, P_{2i}, P_{3i}$ to tags.

Brief description: During authentication phase, the backend server generates a random number v and sends it to the reader. The reader broadcasts this v. The tags within the communication range of the reader receives this v and replies with authentication information K_i along with random number g_i and index value IN_i. Reader receives the responses from the tags and forwards these to the backend server. The backend server receives each set of response IN_i, K_i, g_i and verifies the validity. It starts with the first record kept in the database and uses the sub-record under this record having index IN_i. It verifies K_i using the new identifier and session key and on successful verification, it makes the corresponding update flag U_i as 1. If

Algorithm 2 executed by a tag attached to an object

1: Receives random number v.
2: Generates g_i randomly and then computes $K_i \leftarrow (TID_i - g_i) \oplus ((v \oplus g_i) - N_i)$
3: Sends IN_i, K_i, g_i to reader.
4: Receives $IN_i, P_{1i}, P_{2i}, P_{3i}$ from reader.
5: **If** $IN_i =$ own index **then**
6: $TID_i' \leftarrow ((P_{1i} \oplus ((TID_i \oplus v) - S_i)) - S_i) \oplus g_i$,
7: $N_i' \leftarrow ((P_{2i} \oplus ((N_i \oplus g_i) - S_i)) - S_i) \oplus v$.
8: **If** $P_{3i} = ((S_i \oplus v) - (P_{1i} \oplus TID_i' \oplus N_i)) \oplus ((P_{2i} \oplus TID_i \oplus N_i') - (S_i \oplus v))$ **then**
9: Updates $TID_i \leftarrow TID_i', N_i = N_i'$

verification is not successful using new information, it uses old identifier and session key to verify K_i. This time it makes the update flag U_i as 2. It also increases $valid_j$ by 1 on successful verification using either new or old information. If the verification fails using both new and old information, it selects the next record and continues this until it finds a valid record or finishes with all records in the database.

Thus the backend server verifies all the responses. It then identifies the valid object. To do this, it searches the records with $valid_j$ greater than or equal to the threshold value l. If it finds any such record, it identifies the corresponding object and generates the update information P_{1i}, P_{2i}, P_{3i} for the tags T_i attached to the same object for which the U_i is a nonzero value. If the value of U_i is 1, it uses the new identifier TID_{inew} and session key N_{inew} to generate the update information. Otherwise, it generates the update information using the old identifier TID_{iold} and session key N_{iold}. After completing the updation process, it resets the valid flags $valid_j$ of all the objects and update flags U_i of all the tags kept in the database. The backend server sends IN_i, P_{1i}, P_{2i} and P_{3i} to tags via reader. The tag receives these information and updates its memory after verifying the received information.

4.1.3 Analysis of the Proposed Scheme

We analyze the proposed scheme to evaluate its applicability in practical scenarios. We choose four parameters, namely, security, computation, communication and storage requirements.

4.1.3.1 Security Analysis

The communication between tag and the reader can be misused by the adversaries who may try to mount various attacks. Therefore the proposed scheme needs to be secure against these

Algorithm 3 executed by backend server

Generates and sends a random number v to reader.
Receives K_i, IN_i, g_i from reader.
For all $K_i, \boldsymbol{IN}_i, g_i$
  $j \leftarrow 1$.
  Satisfy $\leftarrow 0$.
  Repeat
    Selects Information kept under index IN_i in j^{th} record
    If $K_i = [(TID_{i,new} - g_i)) \oplus ((v \oplus g_i) - N_{i,new})]$ **then**
      $valid_j \leftarrow valid_j + 1$, $\mathrm{U}_i \leftarrow 1, g_i' \leftarrow g_i$, *Satisfy* $\leftarrow 1$.
    Else If $K_i = [(TID_{i,old} - g_i)) \oplus ((v \oplus g_i) - N_{i,old})]$ **then**
      $valid_j \leftarrow valid_j + 1$, $\mathrm{U}_i \leftarrow 2, g_i' \leftarrow g_i$, *Satisfy* $\leftarrow 1$.
    $j \leftarrow j + 1$
  Until $j > n$ **or** *Satisfy* $= 1$
For $j = 1$ to n
  If $valid_j \geq l$ **then**
    This object is authenticated.
    For $i = 1$ to m
      If $\mathrm{U}_i = 1$ **then**
        Randomly generates TID_i', N_i'.
        Computes $P_{1i} \leftarrow (S_i + (TID_i' \oplus g_i')) \oplus ((TID_{inew} \oplus v) - S_i), P_{2i} \leftarrow (S_i + (N_i' \oplus v)) \oplus ((N_{inew} \oplus g_i') - S_i), P_{3i} \leftarrow ((S_i \oplus v) - (P_{1i} \oplus TID_i' \oplus N_{inew})) \oplus ((P_{2i} \oplus TID_{inew} \oplus N_i') - (S_i \oplus v))$.
        Sends $IN_i, P_{1i}, P_{2i}, P_{3i}$ to reader.
        Updates $TID_{iold} \leftarrow TID_{inew}, N_{iold} \leftarrow N_{inew}, TID_{inew} \leftarrow TID_i', N_{inew} \leftarrow N_i'$.
      Else If $\mathrm{U}_i = 2$ **then**
        Randomly generates TID', N'.
        Computes $P_{1i} \leftarrow (S_i + (TID_i' \oplus g_i')) \oplus ((TID_{iold} \oplus v) - S_i), P_{2i} \leftarrow (S_i + (N_i' \oplus v)) \oplus ((N_{iold} \oplus g_i') - S_i), P_{3i} \leftarrow ((S_i \oplus v) - (P_{1i} \oplus TID_i' \oplus N_{iold})) \oplus ((P_{2i} \oplus TID_{iold} \oplus N_i') - (S_i \oplus v))$.
        Sends $IN_i, P_{1i}, P_{2i}, P_{3i}$ to reader.
        Updates $TID_{inew} \leftarrow TID_i', N_{inew} \leftarrow N_i'$.
For $j = 1$ to n
  $valid_j \leftarrow 0$
  For $i = 1$ to m
    $U_i \leftarrow 0$

attacks. We analyze the security of the proposed scheme in this section.

Informal security analysis

We informally analyze how an adversary $\mathcal{A}$ can mount various attacks mentioned in the threat model and how the proposed scheme can prevent these attacks.

- *Eavesdropping:* $\mathcal{A}$ can intercept $g_i, v, K_i, P_{1i}, P_{2i}, P_{3i}$ and try to find out the secret information such as secret key S_i, session key N_i, etc. For example, he may try to compute TID_i from K_i. He needs to separate $(TID_i - g_i)$ and $((v \oplus g_i) - N_i))$ from K_i and then can compute TID_i from $(TID_i - g_i)$. However, Shannon has proved in [**?**] that it is not possible to separate A and B from $A \oplus B$ as long as the bit size of A and B are same and any of A or B does not contain a value which it had contained in any other session completed earlier.[1] Since the size of $(TID_i - g_i)$ and $((v \oplus g_i) - N_i))$ are same and they are not same in multiple communications, $\mathcal{A}$ is unable to separate these from K_i. Similarly, the other equations are secure from eavesdropping.

- *Location privacy:* $\mathcal{A}$ can try to find out a pattern using the information $K_i, P_{1i}, P_{2i}, P_{3i}, g_i, v$ in multiple sessions. The proposed scheme uses new random numbers in each sessions to generate K_i, P_{1i} etc. For example, K_i consists of randomly generated v, g_i which were not used in the previous sessions. Therefore the adversary cannot relate the K_i of one session with the K_i of other sessions. Similarly, he cannot use other information transmitted through the insecure medium to find a pattern. He can try to use the index information to trace an object. However, there can be the responses with same index information from more that one objects.

- *Location privacy between two successful sessions:* During the time between two successful sessions, $\mathcal{A}$ can try to replay same v and can expect same response K_i from the tag. However, the tag randomly generates g_i and includes it into K_i. Therefore, the responses are not same and $\mathcal{A}$ cannot trace the object. In similar argument, he cannot trace the object intercepting the information P_{1i}, P_{2i}, P_{3i}.

- *Man-in-the-middle attack:* $\mathcal{A}$ can modify K_i and expect that the modified information will be validated in the backend server. However, since he does not know the secret infor-

[1] For i^{th} bit, $C_i = A_i \oplus B_i$. Let B_i is a random bit. Hence for all $i, P(B_i = 0) = P(B_i = 1) = \frac{1}{2}$. Let $P(A_i = 0) = p_i$. Therefore, $P(A_i = 1) = 1 - p_i$. Now, $P(C_i = 0) = P(A_i = 0) \times P(C_i = 0 | A_i = 0) + P(A_i = 1) \times P(C_i = 0 | A_i = 1) = P(A_i = 0) \times P(B_i = 0) + P(A_i = 1) \times P(B_i = 1) = p_i \times \frac{1}{2} + (1 - p_i) \times \frac{1}{2} = \frac{1}{2}$. Therefore, $P(C_i = 0)$ does not depend on p_i. Conversely, we can say that the probability of obtaining correct A_i from the given C_i is $\frac{1}{2}$, where B_i is random.

mation, his modified information cannot be validated successfully. $\mathcal{A}$ can modify P_{1i}, P_{2i} and expect that the tag will extract the wrong information from P_{1i}, P_{2i} and update. However, since he does not know the secret information, he cannot generate a valid P_{3i} which can validate the modified P_{1i} and P_{3i}. The tag will use this P_{3i} to verify the authentication and integrity of P_{1i} and P_{2i}, and will ignore the modified information.

- *Replay attack:* $\mathcal{A}$ can replay v used in the previous session and can expect that the tag will send the same response K_i. Since the tag uses a random number g_i as we have mentioned in the argument of location privacy between two successful sessions, $\mathcal{A}$ cannot trace the object. He can replay the K_i. However, the v used in this K_i is not equal to the v generated in this session by the backend server. Therefore, the replayed information cannot be validated in the backend server. Similarly, the replayed P_{1i}, P_{2i} and P_{3i} cannot be validated in the tag due to new g_i and v.

- *Forward secrecy:* Suppose $\mathcal{A}$ captures TID_i and try to compute TID'_i. Since he does not know S_i, he is unable to compute $((\mathit{TID}_i \oplus v) - S_i)$ from P_{1i} and hence cannot compute TID'_i. In similar argument, he cannot compute N'_i using N_i. If he is able to capture S_i, then also he cannot compute TID'_i or N'_i since he does not know TID_i. If he is able to capture all S_i, N_i and TID_i, then only he can compute TID'_i or N'_i.

- *Backward secrecy:* Similar to forward secrecy, the proposed scheme prevents the backward secrecy, i.e $\mathcal{A}$ can only be able to intercept S_i, N_i and TID_i if he is able to capture all the secrets $S_i, N'_i, \mathit{TID}'_i$.

- *De-synchronization attack:* $\mathcal{A}$ tries to mount this attack as following:

  a) Blocks the update information P_{1i}, P_{2i}, P_{3i} and expects that the backend server has modified the session key and tag id, however, the tag has not updated the corresponding secrets and they cannot communicate in future. The proposed scheme keeps the old copy of the tag id and session key. Therefore, the response from the tag can be validated in backend server using the old information and this information is unchanged until the backend server finds that the tag has updated its information, i.e the response from the tag has verified using new information.

  b) Modifies P_{1i}, P_{2i} and expects that the tag retrieves the wrong information and hence there will be a mismatch between the information in tag and the backend server. As we have explained in the Man-in-the-middle attack, the tag will update only if it verifies P_{3i} successfully which is the integrity information of P_{1i}, P_{2i}. Hence tampering of P_{1i}, P_{2i} will be detected and the tag will not update the secrets.

Therefore, the proposed scheme prevents the De-synchronization attack and the tag and the backend server can still be able to communicate further after this attack.

- *Impersonation attack:* $\mathcal{A}$ may physically clone a legitimate tag and use the cloned tag to impersonate the corresponding object. The proposed scheme uses a threshold value (l) to validate an object, i.e $\mathcal{A}$ have to clone at least the threshold number of tags in order to impersonate an object. This will increase the difficulty for $\mathcal{A}$ to mount this attack. Thus the existence of multiple number tags in an object helps to increase the difficulty for the adversary. The proposed scheme has taken this advantage to increase the security during authentication.

Formal Security Analysis

In this section, we provide formal proofs which can assure the security of the proposed scheme. Firstly, we show that the adversary is unable to mount any attack by intercepting information transmitted through insecure medium during a particular session. Secondly, the adversary may try to intercept information transmitted during multiple sessions and try to mount attacks after manipulation of these information. We show that the proposed scheme is safe from this operation. Finally, we show that the adversary may try to approximate the addition or subtraction operation used in the equations for the information transmitted through insecure medium into XOR operation and try to mount attacks described in the threat model. We show that the probability of such attack in the proposed scheme is negligible.

Definition 1: *(Security of the Object Authentication Scheme (OAS))*. The OAS is secure if, any efficient adversary, given any one interaction (not necessarily complete) and a history of earlier interactions, cannot derive (with probability greater than $0.5 + \theta$, for a non-negligible θ) any secret.

Problem 1: Find p and q from a given number n, where p, q are unknown random numbers of same length (bit size) and $n = p \oplus q$.

Hardness of Problem 1: Let $Adv_{\mathcal{A}}^{XOR}$ denotes an adversary $\mathcal{A}$'s advantage in finding p and q from the given n, we have $Adv_{\mathcal{A}}^{XOR} = Pr[(p, q) \Leftarrow_R \mathcal{A} : p, q$ being random numbers of same length and $n = p \oplus q]$ [2]. $\mathcal{A}$ is allowed to be probabilistic and the probability in the advantage is computed over the random choices made by $\mathcal{A}$. We call the Problem 1 as computationally infeasible, if $Adv_{\mathcal{A}}^{XOR} \leq \epsilon$, for any sufficiently small $\epsilon > 0$.

[2] $(x, y) \Leftarrow_R \mathcal{A}$ denotes pair (x, y) is selected randomly by the adversary $\mathcal{A}$.

We define a random oracle Disclose.

Disclose: This random oracle unconditionally outputs p, q from the input n, where $n = p \oplus q$.

Theorem 4.1.1. *The proposed object authentication scheme (OAS) is secure from intercepting the secret information by $\mathcal{A}$ under the experiment depicted in Algorithm 4.*

Algorithm 4 : $EXP_A^{OAS}1$

1: Intercepts $g_i, v, K_i, P_{1i}, P_{2i}, P_{3i}$
2: Calls Disclose on input K_i and obtains $(TID_i - g_i), ((v \oplus g_i) - N_i) \leftarrow$ Disclose(K_i)
3: Computes $TID_i \leftarrow (TID_i - g_i) + g_i$, $N_i \leftarrow -(((v \oplus g_i) - N_i) - (v \oplus g_i))$
4: Calls Disclose on input P_{1i} and obtains $(S_i + (TID_i' \oplus g_i)), ((TID_i \oplus v) - S_i) \leftarrow$ Disclose(P_{1i})
5: Computes $S_i \leftarrow -(((TID_i \oplus v) - S_i) - (TID_i \oplus v))$, $TID_i' \leftarrow ((S_i + (TID_i' \oplus g_i)) - S_i) \oplus g_i$
6: Calls Disclose on input P_{2i} and obtains $(S_i + (N_i' \oplus v)), ((N_i \oplus g_i) - S_i) \leftarrow$ Disclose(P_{2i})
7: Computes $N_i' \leftarrow ((S_i + (N_i' \oplus v)) - S_i) \oplus v$
8: **If** $P_{3i} = ((S_i \oplus v) - (P_{1i} \oplus TID_i' \oplus N_i)) \oplus ((P_{2i} \oplus TID_i \oplus N_i') - (S_i \oplus v))$ **then**
9: Successfully eavesdrop the secrets
10: **Else**
11: Return 0 (Failure)

Proof. $\mathcal{A}$ intercepts $g_i, v, K_i, P_{1i}, P_{2i}, P_{3i}$ and tries to intercept the secret like session key, secret key, etc. using the experiment depicted in Algorithm 4. He calls a random oracle Disclose and finds the components tied with XOR operation. He then computes the secrets. However, the probability that he can separate the components tied with XOR operation depends on probability that he can solve the Problem 1. According to the hardness of the Problem 1, the probability of separating the components tied by XOR operation is sufficiently small. Therefore, the success probability of the experiment depicted in Algorithm 4 is sufficiently small and the proposed scheme is secure under this experiment. □

Corollary 4.1.2 (Theorem 4.1.4)**.** *The proposed object authentication scheme is secure from the attacks described in the Threat model.*

Proof. Suppose, $\mathcal{A}$ intercepts the secret information such as session key, id, etc. using the experiment depicted in Algorithm 4. Since he has the secret information, he can mount the attacks like replay attack, man-in-the-middle attack, de-synchronization attack. He further intercepts

Algorithm 5 $EXP_{A}^{OAS}2$

1: Intercepts $g_i^1, v^1, K_i^1, P_{1i}^1, P_{2i}^1, P_{3i}^1$
2: **If** $K_i^1 = (TID_i' - g_i^1) \oplus ((v^1 \oplus g_i^1) - N_i')$
3: Tracing is successful
4: Computes $TID_i'' \leftarrow (P_{1i}^1 \oplus (((TID_i' \oplus v^1) - S_i) - S_i) \oplus g_i^1$
5: Computes $N_i'' \leftarrow (P_{2i}^1 \oplus ((N_i' \oplus g_i^1) - S_i) - S_i) \oplus v^1$
6: **If** $P_{3i}^1 = ((S_i \oplus v^1) - (P_{1i}^1 \oplus TID_i'' \oplus N_i')) \oplus ((P_{2i}^1 \oplus TID_i' \oplus N_i'') - (S_i \oplus v^1))$ **then**
7: Breaking forward secrecy is successful

the information communicated in the next session and tries to mount the attack against the location privacy using the experiment depicted in Algorithm 5. He can also get the confirmation about the attack against forward and backward secrecy from this experiment. However, the success probability of this experiment depends on the probability of intercepting the various secrets. Therefore the success probability of this experiment depends on the probability of success in the experiment depicted in Algorithm 4 which is sufficiently small. □

Theorem 4.1.3. *The proposed object authentication scheme is secure from $\mathcal{A}$ under the experiment depicted in Algorithm 6.*

$$\mathcal{L} = \left\{ \begin{array}{l} \text{Equations in unsuccessful session } \text{Ses}_i \\ \hline K_i = (TID_i - g_i) \oplus ((v \oplus g_i) - N_i) \\ P_{1i} = (S_i + (TID_i' \oplus g_i)) \oplus ((TID_i \oplus v) - S_i) \\ P_{2i} = (S_i + (N_i' \oplus v)) \oplus ((N_i \oplus g_i) - S_i) \\ P_{3i} = ((S_i \oplus v) - (P_{1i} \oplus TID_i' \oplus N_i)) \oplus ((P_{2i} \oplus TID_i \oplus N_i') - (S_i \oplus v)) \\ \hline \text{Equations in successful session } \text{Ses}_{i+1} \\ \hline K_i^1 = (TID_i - g_i^1) \oplus ((v^1 \oplus g_i^1) - N_i) \\ P_{1i}^1 = (S_i + (TID_i' \oplus g_i^1)) \oplus ((TID_i \oplus v^1) - S_i) \\ P_{2i}^1 = (S_i + (N_i' \oplus v^1)) \oplus ((N_i \oplus g_i^1) - S_i) \\ P_{3i}^1 = ((S_i \oplus v^1) - (P_{1i}^1 \oplus TID_i' \oplus N_i)) \oplus ((P_{2i}^1 \oplus TID_i \oplus N_i') - (S_i \oplus v^1)) \\ \hline \text{Equations in successful session } \text{Ses}_{i+2} \\ \hline K_i^2 = (TID_i' - g_i^2) \oplus ((v^2 \oplus g_i^2) - N_i') \\ P_{1i}^2 = (S_i + (TID_i'' \oplus g_i^2)) \oplus ((TID_i' \oplus v^2) - S_i) \\ P_{2i}^2 = (S_i + (N_i'' \oplus v^2)) \oplus ((N_i' \oplus g_i^2) - S_i) \\ P_{3i}^2 = ((S_i \oplus v^2) - (P_{1i}^2 \oplus TID_i'' \oplus N_i')) \oplus ((P_{2i}^2 \oplus TID_i' \oplus N_i'') - (S_i \oplus v^2)) \end{array} \right\}$$

Proof. $\mathcal{A}$ can intercept the information transmitted in multiple sessions and perform XOR operation over the corresponding equations of the intercepted information. Thus he can obtain secret information and mount various attacks. In order to verify whether this attack is present or not, we perform the experiment depicted in Algorithm 6. We prepared a list of equations $\mathcal{L}$ which consists of the equations for the information transmitted in an unsuccessful session Ses_i, a successful session Ses_{i+1} and another successful session Ses_{i+2}. These sessions are three consecutive sessions. We select the sessions in such a way that any other session cannot provide any extra benefit. The algorithm takes the list $\mathcal{L}$ as an input where each equation in the list has two components that are tied with XOR operation. □

Algorithm 6 $EXP_A^{OAS}3$

1: **For** each pair of equations in $\mathcal{L}$
2: **If** the pair has common component **then**
3: Apply XOR operation on the pair and obtain a new equation $\mathcal{E}'$.
4: **If** $\mathcal{E}' \notin \mathcal{L}$ **then**
5: Add $\mathcal{E}'$ into $\mathcal{L}$
6: **End For**

For example, the equation for K_i consists of two components $(TID_i - g_i)$ and $((v \oplus g_i) - N_i)$ that are tied with XOR operation. If it finds any pair of equations which has a common component tied by XOR operation, it applies XOR operation over these two equations and outputs a new equation which is added to $\mathcal{L}$. It selects this pair to apply the XOR operation because the XOR operation will suppress the common component and hence the resultant equation may become vulnerable. However, if any pair does not have any common component, the XOR operation cannot help. The XOR operation will increase the components in the resultant equation and this cannot be benefited to $\mathcal{A}$. Thus it continues till it obtains a new equation in $\mathcal{L}$. According to our experiment, there is no new equation produced by the Algorithm 6. Therefore the proposed scheme is secure.

XOR-approximation: Approximate a given equation $A = (B + C) \oplus (D - E)$ into another equation $A' = (B \oplus C) \oplus (D \oplus E)$. The probability that $A = A'$ is $(\frac{3}{2})^{d-1}$, where d is the length (bit size) of A, A', B, C, D, E[3].

[3]Replace +/- operation in $A = (B + C) \oplus (D - E)$ with XOR operation to obtain a new equation $A' = (B \oplus C) \oplus (D \oplus E)$. The LSB of A' is same as LSB of A since there is no carry or borrow input bit in LSB. However there can be carry/borrow input bit in other bits and maximum probability that i^{th} bit of A is equals to the i^{th} bit of A' is $\frac{3}{4}$ [**?**]. Therefore, the probability of $A = A'$ is $(\frac{3}{4})^{d-1}$, (d is the bit size of A and A').

Theorem 4.1.4. *The proposed object authentication scheme is secure from the attacks in the threat model under the XOR-approximation assumption.*

Table 4.2: Success probability on various d values

d	1	2	32	64	96	128
ζ	1	7.5×2^{-3}	1.339366×2^{-13}	1.345425×2^{-27}	1.351512×2^{-40}	1.357627×2^{-53}

Proof. The equations used in the proposed scheme consists of +/- operation and $\mathcal{A}$ can convert these equations using XOR approximation and then try to mount the attacks mentioned in the threat model. However, the success probability depends on the successful approximation. The probability of successful approximation is $(\frac{3}{4})^{d-1}$, where d is the length (bit size) of each secure information. Table 4.2 shows this probability on various values of d. Clearly, the probability decreases with increase in d. However, a large d value is computationally infeasible. Therefore, an appropriate value of d needs to be chosen which can be computationally feasible and the success probability to mount various attack is sufficiently small. □

4.1.3.2 Comparison

We compare the proposed authentication scheme with a selected set of existing authentication schemes using a few parameters and these parameters are such as security, computation and communication.

Security comparison:
Table 4.3 shows that the proposed scheme satisfies all the security requirements mentioned in the threat model except the Impersonation attack.

Table 4.3: Security assurance

	a	b	c	d	e	f	g	h	i
Weis et al. [?]	N	Y	N	N	N	Y	Y	Y	N
Randomized hash [?]	N	Y	N	Y	Y	Y	Y	Y	N
Song et al. [?]	Y	Y	Y	Y	Y	N	Y	Y	N
Hyung-Joo et al. [?]	Y	Y	N	Y	Y	Y	Y	Y	N
Guo-Rui Li et al. [?]	Y	N	Y	Y	Y	N	N	N	N
Proposed scheme	Y	Y	Y	Y	Y	Y	Y	Y	P

a: Eavesdropping, b: Man-in-the-middle attack, c: Replay attack, d: Traceability, e: Traceability between two successful sessions, f: Forward security, g: Backward security, h: De-synchronization attack, i: Impersonation attack, Y: Satisfy, N: Not satisfy, P: Partially satisfy,

However, the use of multiple number of tags in each object helps to increase the difficulty for the adversary to mount this attack. The existing schemes are [?] [?] [?] [?] unable to prevent two or more attacks.

Computational Overhead:

We analyze the computational overhead of the proposed scheme and compare the scheme with the existing schemes. Table 4.4 illustrates the computation requirements in various scheme.

Table 4.4: Number of operations performed in various scheme

	Tag					Reader					Backend Server				
	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
Weis et al. [?]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Randomized hash [?]	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0
Song et al. [?]	6	0	6	3	1	0	0	0	0	1	$4n+4$	0	6	$2n+1$	0
Hyung-Joo et al. [?]	7	1	5	0	5	0	0	2	0	1	$4n+4$	n	$n+7$	0	$2n+2$
Guo-Rui Li et al. [?]	2	0	7	5	1	0	0	0	0	1	$n+1$	0	$4n+3$	$4n+2$	1
Proposed scheme	14	8	0	0	1	0	0	0	0	0	$4n^2m+13m$	$6n^2m+6m$	0	0	$2m+1$

a: XOR, b: Addition/Subtraction, c: Attachment/Detachment/Multiplication/Division/Shift, d: Hash, e: Random number generation, q: Number of readers, n: Number of objects, m: Number of tags attached to an object

In our analysis, we consider the operations used in the proposed scheme such as XOR, addition, subtraction, random number generation, and the other operations used in the existing schemes such as hash functions, attachment/detachment operation, etc. Table 4.4 shows that

the tag in the proposed scheme uses most number of XOR and addition, subtraction operation. However, these are elementary operations. It uses only one heavyweight operation, i.e. random number generation. However, the tags in the existing schemes [**?**] [**?**] [**?**] use many heavyweight operations. The schemes proposed in [**?**] use minimum operations. However, these schemes are unable to prevent most of the attacks. Similarly the backend server uses many elementary operations in the proposed scheme whereas in the existing schemes [**?**] [**?**] [**?**], it uses many heavyweight operations. The reader uses no operation in the proposed scheme whereas the schemes proposed in [**?**] [**?**] [**?**] use one or more heavyweight operations. Therefore, the proposed scheme is lightweight in respect to the computation overhead in tag and can be deployable in real life environment.

Communication Overhead:

We compute the overhead due to communication between the components mentioned in the communication model.

Table 4.5: Communication overhead of various scheme

	Tag	Reader	Backend Server
Weis et al. [**?**]	4	6	2
Randomized hash [**?**]	4	6	2
Song et al. [**?**]	4	9	5
Hyung-Joo et al. [**?**]	5	9	4
Guo-Rui Li et al. [**?**]	6	$5 + 6n$	$3n + 2$
Proposed scheme	$4m + 4$	$8m + 6mn + 2$	$4m + 3nm + 1$

n: Number of objects, m: Number of tags attached to an object

Table 4.5 shows that the communication requirements for the existing schemes [**?**] [**?**] [**?**] [**?**] are less. However, the proposed scheme requires to communicate more information due to the fact that multiple number of tags are present in each object. However, multi-tag arrangement helps to increase the difficulty for the adversary to mount attacks.

Storage Requirement:

RFID tags have limited storage capacity. Therefore, we analyze the existing schemes and the proposed scheme in terms of storage requirements. Table 4.6 illustrates this analysis.

Table 4.6: Storage requirement

	Tag	Reader	Backend Server
Weis et al. [**?**]	3	0	$3n$
Randomized hash [**?**]	1	0	n
Song et al. [**?**]	1	0	5
Hyung-Joo et al. [**?**]	2	0	$2n$
Guo-Rui Li et al. [**?**]	3	0	$5n$
Proposed scheme	4	0	$8mn + n$

q: Number of reader, n: Number of objects, m: Number of tags attached to an object

Table 4.6 shows that the scheme proposed in this paper requires to store 4 parameters. If we consider the maximum size of each parameter as 128 bits then the tag requires to store only 512 bits information. The tags in the existing schemes also require to store almost equal number of information bits. Though the backend server does not suffer from storage limitations, we analyze the storage requirement for these components as well. According to Table 4.6, the proposed scheme requires higher storage overhead in backend server. This is again due to the fact that each object is attached with multiple number of tags.

4.2 Traffic congestion problem due to multi-tag environment

In the first authentication protocol, all the tags attached to an object need to be allowed to respond which can help to prove the authentication of the object. Allowing all the tags attached to the object will increase the traffic in the communication path between the reader and object. Therefore, the first protocol suffers from the traffic congestion problem and it requires a modification.

We propose another authentication scheme for multiple objects in multi-tag environment. In this authentication scheme we assume that each object is assigned m number of tags and one of them is selected as master. This tag is responsible to carry out the authentication task for the relevant object. On successful authentication in a particular session, the backend server randomly selects another tag attached to the same object randomly to act as master in the next session and sends a search information via reader. The desired tag verifies the search information and becomes the new master tag. It replies with a confirmation information to the backend server via reader. The backend server then sends a request information to previous master tag to release the master responsibility. The previous master tag verifies the request and releases

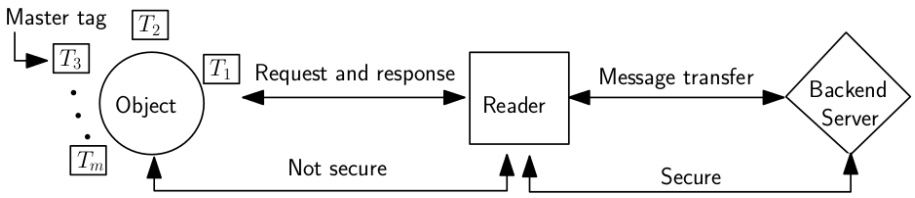

Figure 4.4: Communication model for the authentication scheme 2

master responsibility. The previous master tag or newly selected master tag may not exist within the coverage area of the reader. However they can obtain information through the other tags attached to the same object. In this system, the detection probability of the object does not decrease although the destination tag does not exist within the coverage area of the reader. In order to achieve this, there is a need of inter-tag communication which may help to reach the information to the appropriate destination. Also the traffic congestion does not increase as a result of single response from each object. Before describing the second authentication scheme, we describe the *Communication Model* we have used in this protocol.

4.2.1 Communication model

In the proposed second authentication scheme, the components involve are RFID tag, RFID reader, and backend server. In some works, the backend server has not been used and the RFID reader stores necessary information. However, the reader may not have sufficient memory to keep information about all the objects in a set and hence may suffer from scalability problem. Due to this limitation, we keep backend server in our communication model. Figure. 4.4 illustrates the communication model.

We consider m number of tags are attached to each object. At least one tag is reachable to the reader if any part of the object is within the communication range. In each session, a tag among these tags acts as master. This master tag will generate the authentication information for the object. This tag may not be reachable from the reader and hence it accesses any information for the object through other tags. Each tag contains a static routing table and it is created just after the attachment of all tags to the object. A tag uses its routing table for sending any information if it knows the destination. Otherwise, it broadcasts.

We assume that the communication between reader and backend server is secure while the communication between reader and object is insecure.

The backend server is a workstation. We assume that it is scalable in terms of memory and computation. A database in it contains information about each object.

4.2.2 Proposed authentication scheme

We propose an authentication scheme where m number of tags are attached with an object and one of them is selected as master. This tag is responsible to carry out the authentication task of the relevant object. On successful verification of authentication in a particular session, the back-end server randomly selects another tag attached to the same object randomly to act as master in the next session and sends a search information via reader. The desired tag verifies the search information and becomes the master tag. It replies with a confirmation information to backend server via reader. The backend server then sends a request information to previous master tag to release the master responsibility. The previous master tag verifies the request and releases master responsibility. The previous master tag or newly selected master tag may not be within the communication range of the reader and hence there is a need of inter-tag communication which may help to reach the information to the appropriate destination. Table 4.7 shows the symbols we have used in following discussion.

Table 4.7: Symbols used to describe the authentication protocol 2

Symbols	Meaning	Size (bits)
m	Number of tags attached to G	–
TID_{old}	Old Tag id	d
TID_{new}	New Tag id	d
TID_{nm}	Tag id of new master tag	d
TID'	Randomly generated Tag id	d
N_{old}	Old session key	d
N_{new}	New session key	d
N_{nm}	Session key of new master tag	d
N'	Randomly generated session key	d
L	Secret key	d
L_{nm}	Secret key of new master tag	d
ML	Master secret key	d
$MTID$	Master tag id	d
MN	Master session key	d
IN_i	Index of i^{th} tag in G	d
g, v	Random numbers	d

4.2.2.1 Information in tag and backend server

In this section, we describe various databases used in the proposed scheme. The information for a tag are kept in the tag memory and the information of all the tags attached to an object are kept in a record in the database of beckend server.

Contents in tag memory: In the memory of a tag, the first field contains the pairwise secret key L. The second and third fields contain new and old tag id respectively. The fourth and fifth fields contain new and old session key respectively. The sixth field contains the routing table which is used to send any information in shortest path to another tag attached to the same object. This is a static table and it is created immediately after the attachment of all tags in the object.

L	TID_{new}	TID_{old}	N_{new}	N_{old}	*Routing table*

Information in tag memory

We describe the routing table of a tag having index IN_2 attached to object G. This routing table contains the indices of nearest neighbors through which there are shortest paths from this tag to other tags attached to the same object. For example, if this tag wants to send any information to the tag having index IN_3, it consults the routing table and obtains IN_4 as the index of nearest neighbor to the destination. Hence, it sends the information to tag having index IN_4. The tag having index IN_4 then sends it to its nearest neighbor and thus the information can reach to its destination.

Table 4.8: Routing table of tag having index IN_2

IN_1	IN_3	...	IN_m
IN_1	IN_4	....	IN_1

Contents in the backend server: The database in the backend server has a table which contains the records for each object. In the record for object G, the first m fields contain the information for the tags attached to G. Each of these fields contains a secret key, a tag id, and a session key for the corresponding tag attached to G. The next three fields contains the latest values of master secret key, master tag id, and master session key. Table 4.9 shows the content of a record kept in the backend server.

Table 4.9: Information for an object in backend server

IN_1	IN_2	$\cdots$	IN_m	*Master secret*	*Master TID*	*Master session key*
L_1, TID_1, N_1	L_2, TID_2, N_2	$\cdots$	L_m, TID_m, N_m	*ML*	*MTID*	*MN*

4.2.2.2 Description of the protocol

The proposed scheme has two phases − Setup phase and Authentication phase. The schematic diagram of the proposed scheme is shown in Table 4.10

Setup phase: During Setup phase, the tags and the database in the backend server are initialized. There is a set of objects in this RFID system. Each object is assigned m number of tags. We consider the initialization process of a tag attached to an object G and the same process is followed to initialize all the tags in all objects. The steps are described below.

i) A tag id *TID*, a session key N, and a pairwise secret key L is generated randomly for each tag attached to an object G.

ii) The parameter L is loaded into the secret key field L in the tag memory, *TID* is loaded into both the fields TID_{new} and TID_{old} in the tag memory. Similarly N is loaded into both the fields N_{new} and N_{old} in the tag memory. There are three fields for secret key, tag id, and session key in the record for a tag attached to G in the database of the backend server. The parameters L, TID, N are loaded into the corresponding fields of the same tag in backend server.

iii) For object G in the backend server, a tag is selected randomly to be the master tag and the secret key, tag id, and session key of that tag are loaded into the fields $ML, MTID, MN$ respectively.

Authentication phase: The entities involved during the authentication of an object are reader, interface tag, current master tag, backend server, and new master tag. First we show the steps followed by these entities and then briefly describe the authentication process.

i) **Reader:** The reader in RFID communication scatters electromagnetic signal to identify any tag within its communication range and performs the tasks specified in Algorithm 7.

Table 4.10: Second authentication protocol

Setup phase:			
Tag		**Reader**	**Backend server**
$L, TID_{new}, TID_{old}, N_{new}, N_{old}$			$[(L_1, TID_1, N_1), (L_2, TID_2, N_2), ...,$ $(L_m, TID_m, N_m)], ML, MTID, MN$
Authentication phase:			
Current master			
2. Generates authentication information *g, K, T*	1. *Request with v* $\longleftarrow$ 3. *g, K, T* $\longrightarrow$	4. *v, g, K, T* $\longrightarrow$	5. If K and T are valid Selects another tag as new master tag. Generates K_1, M_1, M_2, M_3
New master			
8. If M_3 is valid on L, M_1, M_2 If K_1 is valid on L, TID_{new}, N_{new} Extracts TID', N' from M_1, M_2 using L, TID_{new}, N_{new}. Generates Q using L, TID', N'. Updates $TID_{old} \leftarrow TID_{new}$, $N_{old} \leftarrow N_{new}, TID_{new} \leftarrow TID'$, $N_{new} \leftarrow N'$. Else if K_1 is valid on L, TID_{old}, N_{old} Extracts TID', N' from M_1, M_2 using L, TID_{old}, N_{old}. If $TID'! = TID_{new}$ and $N'! = N_{new}$ Generates Q using L, TID', N'. Updates $TID_{new} \leftarrow TID'$, $N_{new} \leftarrow N'$.	7. *g, v,* K_1, M_1, M_2, M_3 $\longleftarrow$ 9. Q $\longrightarrow$	6. K_1, M_1 M_2, M_3 $\longleftarrow$ 10. Q $\longrightarrow$	11. If Q is valid Authentication is successful Generates K_2.
Current master			
14. If K_2 is valid Releases master responsibility.	13. K_2 $\longleftarrow$	12. K_2 $\longleftarrow$	

Algorithm 7 executed by reader

1: Broadcasts a random number v.
2: Receives random number g and computed information K, T from current master tag and forwards the information *g, K, T* along with v to backend server.
3: Receives computed information K_1, M_1, M_2, M_3 from the backend server and forwards these to new master tag along with g, v.
4: Receives computed information Q from new master tag and forwards it to the backend server.
5: Receives computed information K_2 from the backend server and forwards it to current master tag.

ii) **Interface tag(λ):** The tag which is within the communication range of the reader is helpful in the situation when the current master tag or newly selected master tag are not within the communication range of the reader. It usually receives information from the reader and forwards them to current master tag or newly selected master tag and vice versa. We describe the tasks performed by the interface tag (λ) in Algorithm 8.

Algorithm 8 executed by interface tag(λ)

1: Receives v from reader.
2: Broadcasts own index IN_λ and v to current master tag ignoring any further v, IN_λ.
3: Receives g, K, T from current master tag and forwards these to reader.
4: Receives g, v, K_1, M_1, M_2, M_3 from reader and broadcasts these along with own index IN_λ to new master tag ignoring further g, v, K_1, M_1, M_2, M_3.
5: Receives Q from new master tag and forwards this to the reader.
6: Receives K_2 from the reader and forwards this to current master tag.

iii) **Current master tag:** This is a tag in object G which is responsible to prove authentication in current session. This may or may not be within the communication range of the reader. We describe the tasks performed by it in Algorithm 9.

Algorithm 9 executed by a current master tag (cm)

1: Receives either v directly from the reader or IN_λ, v from interface tag.
2: Calculates $K \leftarrow [(TID_{new} - (N_{new} \oplus g)) \oplus (N_{new} - (v \oplus g))], T \leftarrow (L - (K \oplus g)) \oplus ((v \oplus g) - L)$.
3: Sends g, K, T to reader.
4: Receives K_2 from reader.
5: **If** $(TID_{new} - (g \oplus L)) \oplus ((v \oplus g) - TID_{new}) = K_2$ **then**
6: Releases master responsibility.

iv) **Backend server:** The workstation which contains the entire information about all the objects in its database and checks the validity of information from an object. This also generates new information about an object. We describe the tasks performed by it in Algorithm 10.

v) **New master tag:** In each session a new tag is selected randomly to do the authentication task. The tag which is selected in current session after successful authentication of G is the new master tag. The responsibilities it has assigned is described in Algorithm 11.

Description of the authentication phase: We briefly describe the authentication phase of the proposed scheme. The reader initiates by broadcasting a request message with a random number v.

If current master tag attached to G is within the communication range of the reader, it receives v and then replies with authentication information g, K, T. Otherwise, another tag attached to G (interface tag λ) which is within the communication range of the reader receives v and broadcasts this along with its own index (IN_λ) ignoring any further v in the same session. Any intermediate tag simply receives and broadcasts these ignoring any further message of same type in the same session. Thus, the current master tag obtains v and IN_λ. It then replies with the authentication information g, K, T to interface tag through shortest path along with its own index IN_{cm}. The index value IN_λ and routing table help the current master tag to obtain the shortest path. The interface tag receives g, K, T and IN_{cm}, and forwards g, K, T to reader.

The reader receives g, K, T and forwards these to the backend server along with v. The backend server, after receiving g, K, T and v, verifies the authentication using master tag id, master session key and master secret key kept in each record. If a valid record is found, the response is treated as legitimate. Otherwise, the session is terminated.

Another tag in the valid record is selected randomly to transfer master responsibility. The backend server now randomly generates a new tag id TID' and a new session key N' and then

Algorithm 10 executed by backend server

1: Receives g, K, T, v from reader.
2: valid $\leftarrow 0$.
3: $j \leftarrow 1$.
4: **Repeat**
5: Selects *MN*, *MTID*, *ML* from j^{th} record.
6: **If** $K = [(MTID - (MN \oplus g)) \oplus (MN - (v \oplus g))]$ and $T = (ML - (K \oplus g)) \oplus ((v \oplus g) - ML)$ **then**
7: Randomly generates TID', N'.
8: Selects a tag randomly from the valid record and use id, session key and secret key TID_{nm}, N_{nm}, L_{nm} of this tag.
9: Calculates $K_1 \leftarrow [(TID_{nm} - (v \oplus N_{nm})) \oplus ((v \oplus g) - N_{nm})]$,
10: $M_1 \leftarrow (L_{nm} + TID') \oplus ((N_{nm} \oplus v) - L_{nm})$,
11: $M_2 \leftarrow (L_{nm} + N') \oplus ((TID_{nm} \oplus g) - L_{nm})$,
12: $M_3 \leftarrow (L_{nm} - (M_1 \oplus g)) \oplus ((M_2 \oplus v) - L_{nm})$.
13: Sends K_1, M_1, M_2, M_3 to reader.
14: Receives Q from reader.
15: **If** $Q = (L_{nm} - (TID' \oplus g)) \oplus ((N' \oplus v) - L_{nm})$ **then**
16: j^{th} record has been satisfied for authentication.
17: $valid \leftarrow 1$.
18: Sends $K_2 \leftarrow [(MTID - (g \oplus ML)) \oplus ((v \oplus g) - MTID)]$ to reader.
19: Updates $MN \leftarrow N'_{nm}, MTID \leftarrow TID', ML \leftarrow L_{nm}, TID_{nm} \leftarrow TID', N_{nm} \leftarrow N'$.
20: $j \leftarrow j + 1$.
21: **Until** $valid = 1$ or $j > n$
22: **If** $valid = 0$ **then**
23: Authentication is not successful.

generates update information M_1, M_2 and M_3. It also generates authentication information K_1 and sends these to the reader. The reader broadcasts these along with v, g to newly selected master tag.

The tag within the communication range of the reader checks integrity of these information using its own secret key L and then verifies authentication of K_1 using its tag id TID_{new} and session key N_{new} kept in its memory. If verification is successful, it retrieves new tag id and session key TID' and N'. It replaces TID_{old} and N_{old} by TID_{new} and N_{new} respectively, and then replaces TID_{new} and N_{new} by TID' and N' respectively. If verification is not successful, it verifies K_1 using TID_{old} and N_{old}. On successful verification, it retrieves TID' and N', and

Algorithm 11 executed by a new master tag (nm)

1: Receives g, v, K_1, M_1, M_2, M_3 from reader.
2: **If** $M_3 = (L - (M_1 \oplus g)) \oplus ((M_2 \oplus v) - L)$ **then**
3: **If** $K_1 = [(TID_{new} - (v \oplus N_{new})) \oplus ((v \oplus g) - N_{new})]$ **then**
4: Becomes new master tag and calculates
5: $TID' \leftarrow (M_1 \oplus ((N_{new} \oplus v) - L)) - L$,
6: $N' \leftarrow (M_2 \oplus ((TID_{new} \oplus g) - L)) - L$,
7: $Q \leftarrow (L - (TID' \oplus g)) \oplus ((N' \oplus v) - L)$.
8: Sends Q to reader.
9: Updates $TID_{old} \leftarrow TID_{new}, N_{old} \leftarrow N_{new}$ and then $TID_{new} \leftarrow TID', N_{new} \leftarrow N'$.
10: **Else if** $K_1 = [(TID_{old} - (v \oplus N_{old})) \oplus ((v \oplus g) - N_{old})]$ **then**
11: Calculates $TID' \leftarrow (M_1 \oplus ((N_{old} \oplus v) - L)) - L$, $N' \leftarrow (M_2 \oplus ((TID_{old} \oplus g) - L)) - L$.
12: **If** $TID' != TID_{new}$ and $N' != N_{new}$ **then**
13: Becomes new master tag and calculates
14: $Q \leftarrow (L - (TID' \oplus g)) \oplus ((N' \oplus v) - L)$.
15: Sends Q to reader.
16: Updates $TID_{new} \leftarrow TID', N_{new} \leftarrow N'$.

replaces TID_{new} and N_{new} by TID' and N' respectively. If verification is successful, it becomes master tag and sends confirmation Q to reader directly or via interface tag. Otherwise, it checks whether it receives any IN_λ along with v, g, K_1, M_1, M_2 and M_3. If it does not receive such IN_λ, it becomes interface tag and broadcasts v, g, K_1, M_1, M_2 and M_3 along with its own index as IN_λ. Otherwise, being an intermediate tag, broadcasts the received information.

The backend server obtains Q through reader and checks validity. If the verification is successful, it declares that the object is authenticated in j^{th} record and it updates $MTID, MN$ and ML using TID', N' and secret key L of newly selected master tag. It also updates the id and session key of newly selected master tag using TID' and N' respectively. It then sends request message K_2 for releasing the master responsibility to previous master tag. The previous master tag releases the master responsibility after successful verification of K_2. Thus authentication phase is completed.

4.2.3 Analysis of the scheme

It is desirable for any authentication scheme to be efficient and secure against possible attacks, and hence we analyze the proposed scheme. We also compare the proposed scheme with the existing schemes [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] based on various parameters such as security, computations

etc.

4.2.3.1 Security analysis

An adversary $\mathcal{A}$ may try to misuse the information transmitted through insecure medium between reader and object during the authentication process

.

Informal security analysis:

We analyze how an adversary $\mathcal{A}$ can mount various attacks specified in the Threat model and how those attacks are prevented in the proposed scheme.

a) **Eavesdropping:** $\mathcal{A}$ may intercept $K, T, K_1, M_1, M_2, M_3, K_2, Q$ and try to eavesdrop secure information such as session key, identifier etc. For example, $\mathcal{A}$ intercepts $K(= (TID_{new} - (N_{new} \oplus g)) \oplus (N_{new} - (v \oplus g)))$ and tries to eavesdrop N_{new} from it. If $\mathcal{A}$ is able to compute $(N_{new} - (v \oplus g))$ then only he can be able to eavesdrop N_{new}. However the XOR operation is secure when the operands used in this operation are one time pad and the operands bit lengths are same. Shannon [**?**] has proved this, i.e. for a given ciphertext $C(= A \oplus B)$, $\mathcal{A}$ is able to obtain absolutely no additional information[4] about A or B when A and B are of same length (bit size) and are randomly chosen. Here the bit size of $(TID_{new} - (N_{new} \oplus g))$ and $(N_{new} - (v \oplus g)))$ are same and due to randomly chosen new v and g in every response, the components $(TID_{new} - (N_{new} \oplus g))$,$(N_{new} - (v \oplus g)))$ will not remain same. Therefore, the conditions for one time pad are satisfied. Similarly, all the other information transmitted through insecure medium are secure.

b) **Man-in-the-middle attack:** $\mathcal{A}$ may try to modify any information transmitted through insecure medium. $\mathcal{A}$ may send modified K, T to reader. According to the proposed scheme, the backend server checks the authentication of K and then checks the integrity using T. Therefore, the modified K will not be verified in the backend server. The modified K may be accidentally verified for another record kept in the backend server. However $\mathcal{A}$ does not know any secret information and cannot be able to generate a valid T which can confirm the integrity of modified K. $\mathcal{A}$ can guess a valid T, however, the probability of success is 2^{-d}, where d is the bit size of T. Similarly, $\mathcal{A}$ cannot be able to successfully modify K_1, M_1, M_2, since M_3 is used as the integrity information. $\mathcal{A}$ may modify Q and

[4]For i^{th} bit, $C_i = A_i \oplus B_i$. Let B_i is a random bit. Hence for all $i, P(B_i = 0) = P(B_i = 1) = \frac{1}{2}$. Let $P(A_i = 0) = p_i$. Therefore, $P(A_i = 1) = 1 - p_i$. Now, $P(C_i = 0) = P(A_i = 0) \times P(C_i = 0 | A_i = 0) + P(A_i = 1) \times P(C_i = 0 | A_i = 1) = P(A_i = 0) \times P(B_i = 0) + P(A_i = 1) \times P(B_i = 1) = p_i \times \frac{1}{2} + (1 - p_i) \times \frac{1}{2} = \frac{1}{2}$. Therefore, $P(C_i = 0)$ does not depend on p_i. Conversely, we can say that the probability of obtaining correct A_i from the given C_i is $\frac{1}{2}$, where B_i is random.

the backend server cannot assign a newly selected tag as master and the current master tag will continue as master. However, the newly selected tag has already accepted the master responsibility. Since backend server did not assign it as master, the authentication request from this tag will not be verified. If $\mathcal{A}$ modifies K_1, current master tag cannot verify it successfully and continues as master. On the other hand, the backend server have already transfered the master responsibility to newly selected tag. Therefore, any authentication request from the current master tag will not be verified successfully in the next session and any modification by $\mathcal{A}$ will not damage the authentication process.

c) **Replay attack:** $\mathcal{A}$ may store information transmitted in a legitimate session and replay these information later. In the proposed scheme, the information $K, T, K_1, M_1, M_2, M_3, K_2, Q$ consist of random numbers v and g. $\mathcal{A}$ may store K, T and replay to prove the legitimacy. However, the reader will request with v which is not same with the v involved in replayed K, T. Since $\mathcal{A}$ does not know any secret information, he cannot inject new v into the stored K, T. Therefore the replayed K, T cannot be verified. $\mathcal{A}$ may replay K_1, M_1, M_2, M_3 and tag will successfully verify K_1 using old id and session key. However, it will immediately detect the replay attack by comparing the extracted id and session key with new id and session key stored in its memory. If $\mathcal{A}$ modifies M_1, M_2 without modifying K_1, M_3, tag cannot detect replay attack. However modified M_1, M_2 cannot pass the integrity check. Since $\mathcal{A}$ does not know any secrets, he cannot generate a valid M_3 which can help to pass the integrity check. $\mathcal{A}$ may replay Q, K_2. These information will not be verified since the v, g included in replayed information are not same with v, g used in request, response in current session and also $\mathcal{A}$ cannot inject these v, g into the replayed information.

d) **Location tracing:** $\mathcal{A}$ may trace an object based on the information transmitted during legitimate sessions. In the proposed scheme, master tag responds with K, T which involves fresh random numbers v, g. Also the session key and id involved in this response are neither same nor related to the session key and id used in the last successful response from the same tag. Therefore, $\mathcal{A}$ cannot relate this response with the response in last successful session and hence cannot trace the object. $\mathcal{A}$ may try to trace an object by observing K involved in current response and Q, M_1, M_2 involved in the last successful session. However, $\mathcal{A}$ cannot compute id and session key from K and Q, M_1, M_2 and hence cannot trace the object. Therefore due to the use of fresh random numbers v and g, $\mathcal{A}$ cannot trace an object.

e) **Location tracing between two successful sessions:** $\mathcal{A}$ may try to replay or block infor-

mation and hence can trace an object during the time between two successful sessions. $\mathcal{A}$ may use the following strategies to trace the object between two successful sessions.

- $\mathcal{A}$ may replay same v used in last successful session and expects that the tag will reply with same K. In the proposed scheme, the tag uses fresh random number g to generate K, T and hence the responses for the replayed K are not same. $\mathcal{A}$ also cannot relate the responses.
- $\mathcal{A}$ may replay same v, g, K_1, M_1, M_2, M_3 and expect that the tag will respond with same Q. However the tag will detect the replay attack as we have mentioned during the discussion of replay attack and hence will not reply.
- $\mathcal{A}$ may block Q or K_2 and expect that old master will respond with same authentication information K, T on same v. However, use of fresh random number g will mislead him.

From the above cases, we can conclude that there is no location tracing threat during the time between two successful sessions.

h) **De-synchronization attack:** The session key and id of newly selected master tag are updated. $\mathcal{A}$ may block some information and expect that the tag will update its id and session key kept in its memory while the backend server does not. In the proposed scheme, $\mathcal{A}$ may block g, v, K_1, M_1, M_2, M_3. In this situation, since tag does not obtain g, v, K_1, M_1, M_2, M_3, it does not update its id and session key. Also it does not respond. Hence the backend server does not update id and session key for this tag. Therefore there is no de-synchronization threat. However, if $\mathcal{A}$ blocks Q, the tag has already updated its information while the backed server does not update. Therefore, it seems that there can be synchronization problem. In the proposed scheme, there can be two situations.

- Tag validates K_1 using new session key and id stored in its memory and keeps these as old id and session key.
- Tag validates K_1 using old session key and id. However, it does not replace the old information.

In both the situations, the newly selected master tag will not be desynchronized and can recognize any further request using the old information kept in its memory. The authentication process will be failure and the master responsibility will not be transferred. Hence the previously assigned master tag can continue as the current master tag.

f) **Forward secrecy:** Let $\mathcal{A}$ somehow captures session key N_{nm} and id TID_{nm} and tries to compute N' and TID'. Since he does not know L_{nm}, he cannot calculate N' and TID' from

M_1 and M_2, and since he does not know N' and TID' cannot calculate L_{nm}. If he has either N_m or ID_{nm} along with L_{nm} then only he can calculate N' and TID'. Therefore, to disrupt forward secrecy, he has to capture L_{nm} and either N_{nm} or TID_{nm} or both.

g) **Backward secrecy**: Similar to forward secrecy, the adversary $\mathcal{A}$ may try to disrupt the backward secrecy of the proposed scheme. However since he does not have L_{nm} he is unable to compute N_{nm} and TID_{nm} from M_{1i}, M_{2i} using captured N' and TID'.

i) **Impersonation attack:** $\mathcal{A}$ may try to clone the current master tag and can act as a legitimate entity. In the proposed scheme, the master responsibility is not fixed to a particular tag and in each session a new tag in the same object is selected as the master tag for the next session. Since there is multiple number of tags, $\mathcal{A}$ can guess a tag that will be the master in the next session. The probability of success of his guess is $\frac{1}{p}$, where p is the number of tags attached to an object. If $p = 8$, the probability that $\mathcal{A}$ will be successful is 0.125. Therefore, the use of multiple tags increases the difficulty for the adversary to physically capture the appropriate tag.

Formal security analysis:

We provide formal proofs which can confirm the security of the proposed scheme. First, we show that there is sufficiently small advantage for the adversary to mount any attack defined in the Threat model. Secondly, the adversary can use information transmitted in multiple sessions and hence he can mount the attacks. We provide a formal proof that he cannot utilize the information used in multiple sessions. Finally, we show that the adversary has a very small probability to mount any attack after approximating the addition/subtraction operation into XOR operation.

Definition 1: *Security of the Second Object Authentication Scheme (SOAS).* The SOAS is secure if, any efficient adversary, given any one interaction (not necessarily complete) and a history of earlier interactions, cannot derive (with probability greater than $0.5 + \theta$, for a non-negligible θ) any secret.

Problem 1: Find p and q from a given number n, where p, q are unknown random numbers of same length (bit size) and $n = p \oplus q$.

Hardness of Problem 1: Let $Adv_{\mathcal{A}}^{XOR}$ denotes an adversary $\mathcal{A}$'s advantage in finding p and q from the given n, we have $Adv_{\mathcal{A}}^{XOR} = Pr[(p, q) \Leftarrow_R \mathcal{A} : p, q$ being random numbers of same

length and $n = p \oplus q$] [Note: $(x, y) \Leftarrow_R \mathcal{A}$ denotes pair (x, y) is selected randomly by the adversary $\mathcal{A}$]. $\mathcal{A}$ is allowed to be probabilistic and the probability in the advantage is computed over the random choices made by the adversary $\mathcal{A}$. We call the Problem 1 is computationally infeasible, if $Adv_{\mathcal{A}}^{XOR} \leq \epsilon$, for any sufficiently small $\epsilon > 0$.

We define a random oracle Disclose.

Disclose: This random oracle unconditionally outputs p, q from the input n, where $n = p \oplus q$.

Theorem 4.2.1. *The proposed Second object authentication scheme (SOAS) is secure under the experiment depicted in Algorithm 12 against an efficient adversary for deriving any secret of a tag after intercepting the information transmitted through insecure medium during a particular session.*

Algorithm 12 *EXP* $1_{\mathcal{A},SOAS}^{XOR}$ (T_2 is newly selected master tag during a successful session Ses_i)

1: Intercepts v, g, K_1, M_1, M_2, M_3 in successful session Ses_i, where $K_1 = [(\textit{TID}_2 - (v \oplus N_2)) \oplus ((v \oplus g) - N_2)], M_1 = (L_2 + \textit{TID}_2') \oplus ((N_2 \oplus v) - L_2), M_2 = (L_2 + N_2') \oplus ((\textit{TID}_2 \oplus g) - L_2), M_3 = (L_2 - (M_1 \oplus g)) \oplus ((M_2 \oplus v) - L_2)$
2: Call Disclose on input K_1 and obtains $(\textit{TID}_2 - (v \oplus N_2)), ((v \oplus g) - N_2) \leftarrow Disclose(K_1)$
3: Calculates $N_2 \leftarrow (-1) \times (((v \oplus g) - N_2) - (v \oplus g)), \textit{TID}_2 \leftarrow (\textit{TID}_2 - (v \oplus N_2)) + (v \oplus N_2)$
4: Call Disclose on input M_1 and obtains $(L_2 + \textit{TID}_2'), ((N_2 \oplus v) - L_2) \leftarrow Disclose(M_1)$
5: Calculates $L_2 \leftarrow (-1) \times (((N_2 \oplus v) - L_2) - (N_2 \oplus v)), \textit{TID}_2' \leftarrow (L_2 + \textit{TID}_2') - L_2$
6: Call Disclose on input M_2 and obtains $(L_2 + N_2'), ((\textit{TID}_2 \oplus g) - L_2) \leftarrow Disclose(M_2)$
7: Calculates $N_2' \leftarrow (L_2 + N_2') - L_2$
8: Call Disclose on input M_3 and obtains $(L_2 - (M_1 \oplus g)), ((M_2 \oplus v) - L_2) \leftarrow Disclose(M_3)$
9: Calculates $L_2' \leftarrow (L_2 - (M_1 \oplus g)) + (M_1 \oplus g)$
10: **If** $L_2' \,!= L_2$ **then**
11: return 0 (Failure)
12: **else**
13: Successful to intercept $\textit{TID}_2, \textit{TID}_2', N_2, N_2', L_2$

Proof. We construct an adversary $\mathcal{A}$ such that $\mathcal{A}$ has the ability to derive various secrets. $\mathcal{A}$ plays a game. $\mathcal{A}$ is allowed to access the communication medium between tags and the reader, and he is given the information transmitted through insecure medium in a particular session Ses_i as a challenge. He can win the game if he is successful to compute the secrets and hence mount

Algorithm 13 *EXP* $2^{Attacks}_{\mathcal{A},SOAS}$ (T_2 is current master tag during session Ses_{i+1})

1: Intercepts v_1, g_1, K, T in session Ses_{i+1}, where $K = [(TID'_2 - (N'_2 \oplus g_1)) \oplus (N'_2 - (v_1 \oplus g_1))], T = (L_1 - (K \oplus g_1)) \oplus ((v_1 \oplus g_1) - L_1)$.
2: Calculates $TID''_2 \leftarrow (K \oplus (N'_2 - (v_1 \oplus g_1))) + (N'_2 \oplus g_1)$.
3: **If** $TID'_2 != TID''_2$
4: Return 0 (Failure).
5: **else**
6: Forward and Backward secrecy violated and Object attached with T_2 has traced.

various attacks using these secrets. He does an experiment depicted in Algorithm 12. We define the success probability for $EXP1^{XOR}_{\mathcal{A},SOAS}$ as $Succ1^{XOR}_{\mathcal{A},SOAS} = Pr[EXP1^{XOR}_{\mathcal{A},SOAS} = 1]$. Then the advantage of experiment $EXP1^{XOR}_{\mathcal{A},SOAS}$ is given by $Adv^{XOR}_{\mathcal{A},SOAS} = max_{\mathcal{A}}[Succ1^{XOR}_{\mathcal{A},SOAS}]$, where the maximum is taken over all adversary. According to the experiment depicted in Algorithm 12, if an adversary is able to solve the Problem 1, he can be able to calculate all the secrets and win the game. However, according to the hardness of problem 1, we have $Adv^{XOR}_{\mathcal{A},SOAS} \leq \epsilon$ for sufficiently small $\epsilon > 0$. Hence the proposed scheme is secure.

□

Corollary 4.2.2 (Theorem 4.2.1)**.** *The proposed object authentication scheme (SOAS) is secure against any attack in the Threat model based on the search information for a new master tag in a successful session* Ses_i *and the authentication response by the same tag during the next session* Ses_{i+1}.

Proof. We use the same $\mathcal{A}$ which obtains various secrets using the experiment depicted in Algorithm 12. He is further permitted to access the communication medium between reader and tags during the next session when the same tag responds with authentication information K, T, g. $\mathcal{A}$ accesses these information as challenge and tries to mount various attacks specified in the threat model. Since $\mathcal{A}$ has the secrets, i.e. tag id, session key and pairwise secret key, $\mathcal{A}$ has already successfully eavesdropped. $\mathcal{A}$ can also mount man-in-the-middle attack, replay attack and desynchronization attack. To get confirmation of the attacks against forward secrecy, backward secrecy and traceability, $\mathcal{A}$ further conducts an experiment depicted in Algorithm 13. Clearly $\mathcal{A}$ is successful to mount these attacks. However, the probability of success depends on the probability of accessing the various secrets. Therefore, success probability of the experiment depicted in Algorithm 12 entails the success probability of the adversary $\mathcal{A}$ to mount various attacks in the Threat model. Hence the proposed scheme is secure against the attacks in threat model.

□

Theorem 4.2.3. *Proposed object authentication scheme (SOAS) is secure under the experiment depicted in Algorithm 14.*

Proof. We construct an adversary $\mathcal{A}$ who is given a list of equations $\mathcal{L}$ as a challenge which consists of the equations for the parameters transmitted through insecure medium during the following sessions:

- An unsuccessful session Ses_i when the tag T_1 acts as the current master tag and another tag T_2 attached to the same object is picked as new master tag. Random numbers used in this session are v, g.
- Successful session Ses_{i+1} when the tag T_1 acts as the current master tag and another tag T_3 attached to the same object is picked as new master tag. Random numbers used in this session are v_1, g_1.
- Successful session Ses_{i+2} when a tag T_3 acts as the current master tag and another tag T_2 attached to the same object is picked as new master tag. Random numbers used in this session are v_2, g_2.
- Successful or unsuccessful session Ses_{i+3} when a tag T_2 acts as the current master tag and another tag T_3 attached to the same object is picked as new master tag. Random numbers used in this session are v_3, g_3.

$$\mathcal{L} = \left\{ \begin{array}{l} \text{Equations in unsuccessful session } \text{Ses}_i \\ \hline K = [(\mathit{TID}_1 - (N_1 \oplus g)) \oplus (N_1 - (v \oplus g))] \\ T = (L_1 - (K \oplus g)) \oplus ((v \oplus g) - L_1) \\ K_1 = [(\mathit{TID}_2 - (v \oplus N_2)) \oplus ((v \oplus g) - N_2)] \\ M_1 = (L_2 + \mathit{TID}'_2) \oplus ((N_2 \oplus v) - L_2) \\ M_2 = (L_2 + N'_2) \oplus ((\mathit{TID}_2 \oplus g) - L_2) \\ M_3 = (L_2 - (M_1 \oplus g)) \oplus ((M_2 \oplus v) - L_2) \\ Q = (L_2 - (\mathit{TID}'_2 \oplus g)) \oplus ((N'_2 \oplus v) - L_2) \\ K_2 = (\mathit{TID}_1 - (g \oplus L_1)) \oplus ((v \oplus g) - \mathit{TID}_2) \\ \hline \text{Equations in successful session } \text{Ses}_{i+1} \\ \hline K' = [(\mathit{TID}_1 - (N_1 \oplus g_1)) \oplus (N_1 - (v_1 \oplus g_1))] \\ T' = (L_1 - (K' \oplus g_1)) \oplus ((v_1 \oplus g_1) - L_1) \\ K'_1 = [(\mathit{TID}_3 - (v_1 \oplus N_3)) \oplus ((v_1 \oplus g_1) - N_3)] \\ M'_1 = (L_3 + \mathit{TID}'_3) \oplus ((N_3 \oplus v_1) - L_3) \\ M'_2 = (L_3 + N'_3) \oplus ((\mathit{TID}_3 \oplus g_1) - L_3) \\ M'_3 = (L_3 - (M'_1 \oplus g_1)) \oplus ((M'_2 \oplus v_1) - L_3) \\ Q' = (L_3 - (\mathit{TID}'_3 \oplus g_1)) \oplus ((N'_3 \oplus v_1) - L_3) \\ K'_2 = (\mathit{TID}_1 - (g_1 \oplus L_1)) \oplus ((v_1 \oplus g_1) - \mathit{TID}_2) \\ \hline \text{Equations in successful session } \text{Ses}_{i+2} \\ \hline K'' = [(\mathit{TID}'_3 - (N'_3 \oplus g_2)) \oplus (N'_3 - (v_2 \oplus g_2))] \\ T'' = (L_3 - (K'' \oplus g_2)) \oplus ((v_2 \oplus g_2) - L_3) \\ K''_1 = [(\mathit{TID}_2 - (v_2 \oplus N_2)) \oplus ((v_2 \oplus g_2) - N_2)] \\ M''_1 = (L_2 + \mathit{TID}''_2) \oplus ((N_2 \oplus v_2) - L_2) \\ M''_2 = (L_2 + N''_2) \oplus ((\mathit{TID}_2 \oplus g_2) - L_2) \\ M''_3 = (L_2 - (M''_1 \oplus g_2)) \oplus ((M''_2 \oplus v_2) - L_2) \\ Q'' = (L_2 - (\mathit{TID}''_2 \oplus g_2)) \oplus ((N''_2 \oplus v_2) - L_2) \\ K''_2 = (\mathit{TID}'_3 - (g_2 \oplus L_3)) \oplus ((v_2 \oplus g_2) - \mathit{TID}'_3) \\ \hline \text{Equations in successful session } \text{Ses}_{i+3} \\ \hline K''' = [(\mathit{TID}''_2 - (N''_2 \oplus g_3)) \oplus (N''_2 - (v_3 \oplus g_3))] \\ T''' = (L_2 - (K''' \oplus g_3)) \oplus ((v_3 \oplus g_3) - L_2) \\ K'''_1 = [(\mathit{TID}'_3 - (v_3 \oplus N'_3)) \oplus ((v_3 \oplus g_3) - N'_3)] \\ M'''_1 = (L_3 + \mathit{TID}''_3) \oplus ((N'_3 \oplus v_3) - L_3) \\ M'''_2 = (L_3 + N''_3) \oplus ((\mathit{TID}'_3 \oplus g_3) - L_3) \\ M'''_3 = (L_3 - (M'''_1 \oplus g_3)) \oplus ((M'''_2 \oplus v_3) - L_3) \\ Q''' = (L_3 - (\mathit{TID}''_3 \oplus g_3)) \oplus ((N''_3 \oplus v_3) - L_3) \\ K'''_2 = (\mathit{TID}''_2 - (g_3 \oplus L_2)) \oplus ((v_3 \oplus g_3) - \mathit{TID}''_2) \end{array} \right\}$$

Algorithm 14 *EXP3*$_{\mathcal{A}}^{SOAS}$

1: **for** each pair of equations in $\mathcal{L}$ **do**
2: **If** the pair has common component **then**
3: Apply XOR operation on the pair and obtain a new equation $\mathcal{E}'$.
4: **If** $\mathcal{E}' \notin \mathcal{L}$ **then**
5: Add $\mathcal{E}'$ into $\mathcal{L}$
6: **end for**

The sessions are selected carefully which may be helpful for $\mathcal{A}$ to mount various attacks and no more sessions can provide any additional advantage to $\mathcal{A}$. $\mathcal{A}$ can win the game if he is able to mount various attacks utilizing $\mathcal{L}$. $\mathcal{A}$ performs the experiment *EXP* $3_{\mathcal{A}}^{SOAS}$ depicted in Algorithm 14. It takes a list of equations $\mathcal{L}$ and finds a pair of equation which have one or more common components which are bound by XOR operation. It performs a XOR operation on this pair and derives a new equation to include into $\mathcal{L}$, if there is no such equation in $\mathcal{L}$ which is similar to the derived equation. It selects this pair because existence of common components can help to have less number of components in the resultant equation and that may help to reveal one or more secret information. However, if the pair does not have any common component, the XOR operation will increase the difficulty to extract the secret information. The list $\mathcal{L}$ is the input of Algorithm 14. After performing the experiment depicted in Algorithm 14, clearly $\mathcal{A}$ cannot obtain any new equation and hence he is unable to obtain any secret or mount any attacks described in Section 4.2.3.1.

□

XOR approximation: For a given equation $A = B + C - D$, (B, C, D are three random numbers of size d bits) if we replace +/- operation by XOR operation we can obtain another equation $A' = B \oplus C \oplus D$. Then the probability that $A = A'$ is $(\frac{3}{4})^{d-1}$. [**?**][5]

Theorem 4.2.4. *Probability that an adversary $\mathcal{A}$ can mount various attacks using XOR approximation is bound above by* $(\frac{3}{4})^{d-1}$

[5]We replace +/- operation in $A = B + C - D$ with XOR operation to obtain a new equation $A' = B \oplus C \oplus D$. The LSB of A' is same as LSB of A since there is no carry or borrow input bit in LSB. However there can be carry/borrow input bit in other bits and maximum probability that i^{th} bit of A is equals to the i^{th} bit of A' is $\frac{3}{4}$ [**?**]. Therefore, the probability of $A = A'$ is $(\frac{3}{4})^{d-1}$, (d is the bit size of A and A').

Table 4.11: Success probability on various d vales

d	1	2	32	64	96	128
ζ	1	7.5×2^{-3}	1.339366×2^{-13}	1.345425×2^{-27}	1.351512×2^{-40}	1.357627×2^{-53}

Proof. The equations used in the proposed scheme has +/- operation and the adversary $\mathcal{A}$ can mount various attacks using XOR-approximation. The probability that he will be successful is $\zeta = (\frac{3}{4})^{d-1}$, where d is the bit size of each parameter. The proposed scheme is secure for sufficiently small ζ and large d. Table 4.11 shows various ζ values. Therefore depending on the security requirement, an appropriate d can be chosen for various parameters to implement the proposed scheme. □

4.2.3.2 Comparison with other schemes:

We compare the proposed scheme with the existing schemes [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] in accordance to the following parameters: security, computation, communication, and storage overhead.

Security comparison:
Table 4.12 illustrates the comparative study in respect to various security requirements specified in threat model.

Table 4.12: security assurance

	a	b	c	d	e	f	g	h	i
Weis et al. [**?**]	N	Y	N	N	N	Y	Y	Y	N
Randomized hash [**?**]	N	Y	N	Y	Y	Y	Y	Y	N
Zhang et al. [**?**]	Y	N	P	Y	N	Y	N	N	N
Song et al. [**?**]	Y	Y	Y	Y	Y	N	Y	Y	N
Khor et al. [**?**]	Y	N	Y	Y	N	N	Y	N	Y
Tzu Chang et al. [**?**]	Y	Y	Y	Y	N	N	Y	Y	N
Hyung-Joo et al. [**?**]	Y	Y	N	Y	Y	Y	Y	Y	N
Proposed scheme	Y	Y	Y	Y	Y	Y	Y	Y	P

a: Eavesdropping, b: Man-in-the-middle attack, c: Replay attack, d: Traceability, e: Traceability between two successful sessions, f: Forward security, g: Backward security, h: De-synchronization attack, i: Impersonation attack, Y: Satisfy, N: Not satisfy, P: Partially satisfy

It can be seen that the proposed scheme satisfies all the security requirements in the Threat model except the impersonation attack. The use of variable tags for authentication task creates the impersonation attack more difficult and hence it is partially satisfied in the proposed scheme. However, all the schemes other than the proposed scheme have one or more security flaws. Therefore, the proposed scheme is more secure as compared to the existing schemes.

Computation overhead:

We estimate the operations used in the proposed scheme and the existing schemes. Since the proposed scheme considers that an object is attached with multiple number of tags, we consider maximum computation overhead for a tag in terms of number of operations in one session. Table 4.13 shows that the tag in existing schemes [?] [?] [?] [?] [?] uses computationally intensive operation like hash whereas the tag in the proposed scheme do not use any such operation. However it uses mostly lightweight operations such as XOR and addition/subtraction and only one random number generation operation which is much less in comparison to [?] [?] [?] [?]. Therefore we can conclude that computation requirement for tag in the proposed scheme is less. The reader in the proposed scheme uses only one random number generation operation. However in the schemes [?] [?] [?] [?], reader requires to compute many operations. The backend server on the other hand, needs to compute more number of XOR and addition/subtraction operations in the proposed scheme. However, in the schemes [?] [?] [?] [?], the backend server also uses almost equal number of such operations.

Table 4.13: Number of operations performed in various scheme

	Tag					Reader					Backend Server				
	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
Weis et al. [?]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Randomized hash [?]	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0
Zhang et al. [?]	1	0	0	$q+2$	0	2	0	0	3	1	0	0	0	0	0
Song et al. [?]	6	0	6	3	1	0	0	0	0	1	$4n+4$	0	6	$2n+1$	0
Khor et al. [?]	2	0	1	1	2	0	0	0	0	0	4	0	1	1	2
Tzu Chang et al. [?]	8	0	0	0	7	1	0	0	1	1	$q+3n+7$	0	0	q	$n+5$
Hyung-Joo et al. [?]	7	1	5	0	5	0	0	2	0	1	$4n+4$	n	$n+7$	0	$2n+2$
Proposed scheme	16	12	0	0	1	0	0	0	0	1	$6n+16$	$4n+12$	0	0	3

a: XOR, b: Addition/Subtraction, c: Attachment/Detachment/Multiplication/Division/Shift, d: Hash, e: Random number generation, q: Number of readers, n: Number of objects

Communication overhead:

We have compared the proposed scheme with the existing schemes in respect to the number of messages which have been communicated by various entities during the authentication process. Table 4.14 shows the comparative study. We consider the maximum communication requirement for a tag in one session.

Table 4.14: Communication overhead of various scheme

	Tag	Reader	Backend Server	*BRO*
Weis et al. [**?**]	4	6	2	4
Randomized hash [**?**]	4	6	2	4
Zhang et al. [**?**]	6	14	8	6
Song et al. [**?**]	4	9	5	4
Khor et al. [**?**]	3	5	2	3
Tzu Chang et al. [**?**]	6	14	8	6
Hyung-Joo et al. [**?**]	5	9	4	5
Proposed scheme	27	22	10	12

n: Number of objects, *BRO*: Between Reader and object

According to Table 4.14, the communication overhead is maximum in the proposed scheme in comparison to the existing schemes [**?**] [**?**] [**?**] [**?**] [**?**] [**?**]. The tags which play intermediate or interface role have to receive and forward many messages to actual destination. This happens when the desired tag is beyond the communication range of reader, and another tag attached to the same object is within the communication range. However, the traffic between reader and object is almost same in comparison to other schemes. Therefore, although the proposed scheme suffers from increased overall communication overhead in some cases due to the use of multiple number of tags in each object, it does not suffer from too much communication overhead between reader and object.

Storage overhead:

Table 4.15 shows the storage overhead for the existing schemes and the proposed scheme. According to Table 4.15, the proposed scheme requires to keep more information than the schemes in [**?**] [**?**] [**?**] [**?**] [**?**].

Table 4.15: Storage requirement

	Tag	Reader	Backend Server
Weis et al. [**?**]	3	0	$3n$
Randomized hash [**?**]	1	0	n
Zhang et al. [**?**]	$2q+2$	4	$4q+n$
Song et al. [**?**]	1	0	5
Khor et al. [**?**]	3	0	$5n$
Tzu Chang et al. [**?**]	4	1	$q+8n$
Hyung-Joo et al. [**?**]	2	0	$2n$
Proposed scheme	$m+5$	0	$3n(3m+1)$

q: Number of reader, n: Number of objects, m: Number of tags attached to an object

However, it has less storage overhead than the scheme in [**?**]. Storage overhead in the proposed scheme is higher due to the following reasons:

- Needs to keep static routing table to perform efficient inter-tag communication.
- Needs to keep more secure information due to lightweight operations used in the proposed scheme.

4.2.4 Conclusion

The information transmitted during the authentication process in RFID technology can be misused. Therefore, the authentication task needs to be secured. Existing authentication schemes assume that the objects are attached with single tag. Use of multiple number of tags in an object can increase its detection probability. Therefore, there is a need of secure authentication scheme in multi-tag environment. This authentication scheme also needs to be lightweight since the RFID tag suffers from various resource constraints. In this chapter, we proposed two such authentication protocols which addresses the authentication process in multi-tag environment. The first authentication scheme we proposed in this chapter assumes that the tags within the coverage area of the reader which are attached to an object are responsible to prove the authentication of the object. The analysis shows that this authentication scheme is secure from the possible attacks and can enhance the difficulty for the adversary to mount attacks. It also helps to increases the detection probability of the objects yet the the computation in tag is low. However, due to the multi-tag environment, the computation in the backend server is high. Due to the similar reason, the communication and storage requirement are comparatively high. Since

multiple tags are allowed to prove the authentication for the object, the traffic in the communication between the reader and object is high and hence we propose a second authentication protocol in multi-tag environment which does not suffer from this problem. This scheme also lightweight and secure which enhances the difficulty for the adversary by taking the advantage of having multiple number of tags. However, this scheme requires to use active tag in order to inter-tag communication. passive tag also can be used in this scheme. However, due to lack of inter-tag communication, detection probability of the object will be less. However, in both the scheme, although the multi-tag concept increases the coverage area, the adversary cannot not obtain any additional information. Therefore, the increased coverage area cannot not help the adversary. The impersonation attack through physical attack are prevented partially. However, use of suitable Physical Unclonable Function (PUF) can help to prevent this attack. The proposed authentication schemes can be extended in order to search an object from a large set of objects. However, the efficiency of this process is low. Therefore, this problem requires a separate solution. In the next chapter Chapter 5), we address this problem.

Chapter 5

A New Object Searching Protocol for Multi-tag RFID

In the RFID technology [**?**], a set of objects are attached with small chips called RFID tag, that contains the identification information (ID) about an object. To find a particular object, a RFID reader requests the tag attached to the desired object to respond. This tag responds with its ID if it exists within the communication range. In this way, the desired object is detected. Any authentication scheme similar to can be used for this purpose. However, the response from each tag within the communication range needs to be verified until there is a response from the tag in the desired object [**?**]. This requires to verify $\frac{n}{2}$ number of tags on the average for each search, where n is the total number of tags. Therefore, a separate protocol needs to be designed in order to solve the object searching problem.

The security and privacy risks involve in searching process are eavesdropping, information leakage, traceability, man-in-the-middle attack, forward secrecy, backward secrecy, replay attack, de-synchronization attack and impersonation attack. Therefore, the searching process needs to be secured. Traditional cryptography techniques are not applicable in RFID communication to secure the searching process due to the resource limitations of RFID tag. The focus of the existing schemes [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] was to design lightweight and secure protocols for object searching problem. These schemes assume that the objects are attached with single tag. However, there can be a situation where the attached tag in the desired object stays outside the communication range of the reader while a few locations of the same object are within the communication range. In this situation, if the reader tries to search this object, it will be undetected. Therefore, the detection probability of the objects in these schemes is less. Attachment of multiple number of tags to an object can improve the detection probability [**?**]. If we apply existing object searching schemes (based on single tag environment) to multi-tag environment,

we have to keep one set of security related information for an object to all the tags attached to it. If an adversary somehow manages to compromise any tag attached to an object, then he can compromise the other tags attached to the object. Therefore the effort requires for the adversary to compromise an object is less. However, we can keep multiple set of security related information to the tags attached to an object and the searching process requires at least the threshold number of valid responses in order to search the desired object. Therefore, the adversary needs to compromise threshold number of tags in order to impersonate an object. Similarly, the adversary has to face more difficulty to mount the attacks such as replay attack, de-synchronization attack, etc. Depending on the application environment, an appropriate threshold value needs to be chosen. Since responses from multiple number of tags for an object need to be processed by the reader and the backend server, the object searching scheme needs to be lightweight.

In this chapter, we propose a lightweight and secure object searching scheme. In the proposed scheme, each object is attached with multiple number of RFID tags and the search process requires at least the threshold number of legitimate responses in order to search an object. In this scheme, the tags attached to the undesired objects respond to a search query using fake information with a certain probability which helps to prevent the information leakage attack. Probability of the useless computation due to these fake responses in the proposed scheme is negligible.

5.1 Communication model

The components involved in the communication process are *object, RFID tag, RFID reader* and *backend server*. Figure 5.1 illustrates the communication model. There are multiple number of

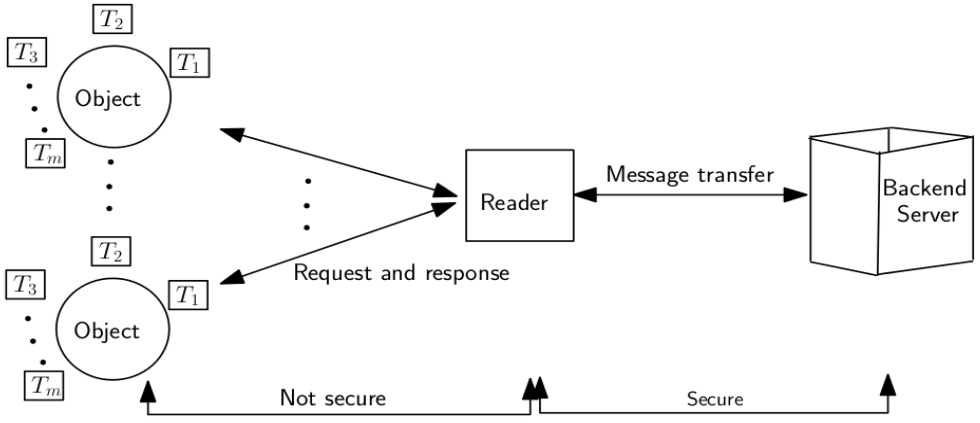

Figure 5.1: Communication model for the proposed object searching protocol

objects and each object is attached with multiple number of RFID tags. RFID tag contains the information about the object with which it is attached. It is responsible to verify any search request and provides appropriate response. The backend server is responsible to generate search information and verify the authentication of any information it receives. It also maintains a database which contains the information about the objects. The RFID reader acts as an intermediator between the tags attached to the objects and the backend server. The communication medium between reader and backend server is secure while the communication medium between reader and tag is insecure.

5.2 Proposed object searching protocol

In the proposed scheme, every objects are attached with multiple number of RFID tags. The RFID reader broadcasts the search information of a desired object. The tags in the desired object verifies the search information and response with authentication information. On the other hand, the tags in the undesired object respond (with a certain probability) with fake information. The reader partially verifies the responses and forwards the partially valid responses to the backend server. The backend server then verifies the responses and thus the desired object is found. Table 5.1 describes the symbols used in the proposed scheme.

Table 5.1: Notations

Symbol	Meaning
G	An object
T_i	i^{th} Tag in G
n	Number of tags attached to G
ID_i	ID of i^{th} tag in G
N_i	Session key of i^{th} tag in G
S_i	Pairwise secret between backend server and i^{th} tag in G
A	A set of random numbers
r_{1i}, r_{2i}	Random numbers for i^{th} tag in G
ID_{inew}	New ID of i^{th} tag in G
N_{inew}	New session key of i^{th} tag in G
$\oplus$	XOR operation
l	Threshold value

5.2.1 Information in the tag memory and backend server

Each tag contains a secret key S_i, two session keys N_{inew}, N_{iold}, and two identifiers ID_{inew}, ID_{iold}. The backend server contains the records for each object. The record for object G contains the

S_i	N_{inew}	N_{iold}	ID_{inew}	ID_{iold}

Information in tag memory

T_1	T_2	...	T_n
S_1, N_1, ID_1	S_2, N_2, ID_2	...	S_n, N_n, ID_n

Record for object G in backend server

information for the tags attached to G. For each tag $T_i, i = 1, 2, ..., n$, the first field contains the secret key S_i, the second and third fields contain the session key N_i and the identifier ID_i, respectively.

5.2.2 Description of the protocol

The proposed scheme has two phases, namely, setup phase and object searching phase. Figure 5.2 shows the schematic diagram of the proposed scheme.

5.2.2.1 Setup phase

In setup phase, the tags and the backend server are initialized with necessary information.

Setup tag: The $i^{th}(i = 1, 2, ..., n)$ tag of object G is assigned a secret key S_i, a session key N_i, and an identifier ID_i and these parameters are loaded into the memory of the same tag where S_i is kept in the secret key field, N_i is kept in both N_{inew} and N_{iold} fields and the ID_i is kept in both ID_{inew} and ID_{iold} fields.

Setup backend server: The database in the backend server is loaded with the records of all objects. The information S_i, N_i, and ID_i for each tag attached to G are kept under the record of G.

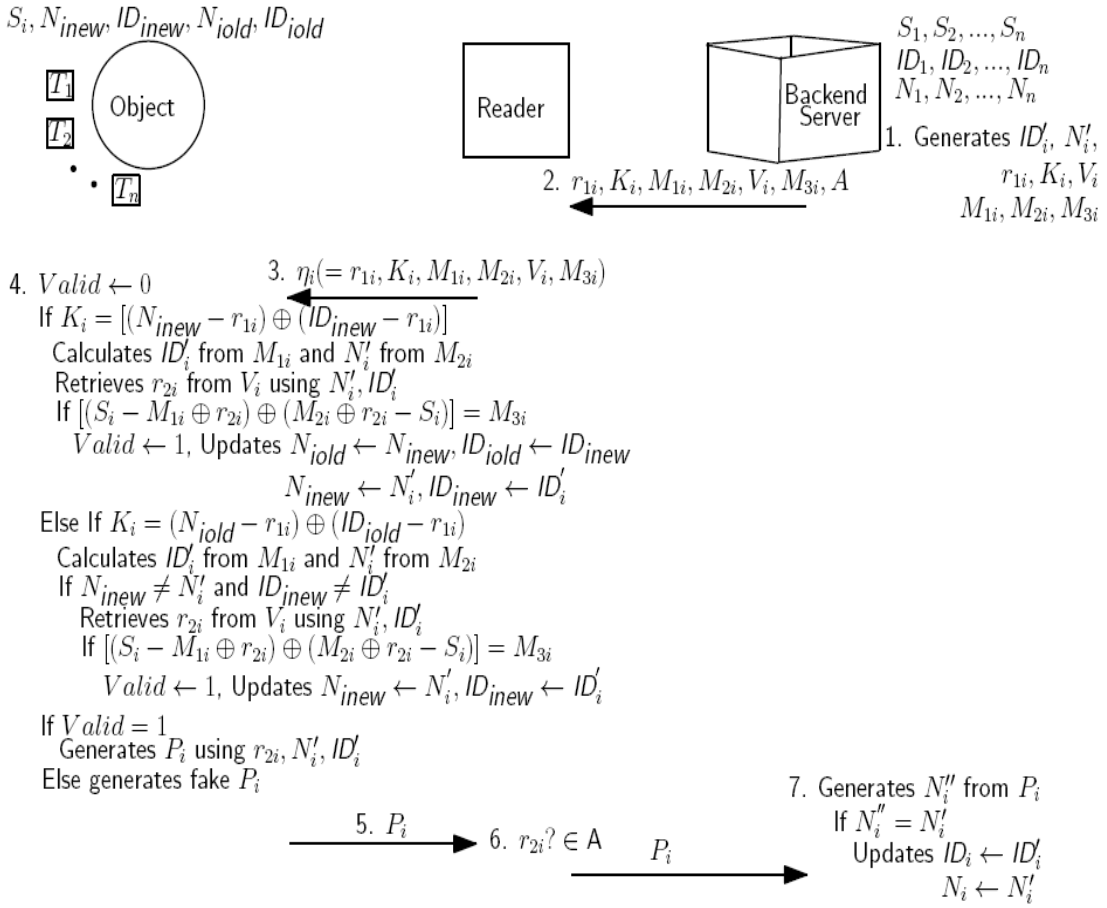

Figure 5.2: Protocol to search an object

5.2.2.2 Object searching phase

The steps require to search an object G attached with n number of tags are depicted in Algorithm 15. We briefly describe the search process in this section.

Description of the object searching phase: To search an object G, the backend server generates a set A having n number of random numbers $(r_{21}, r_{22}, ..., r_{2n})$ in step 1 and then assigns the random number r_{2i} $(1 \leq i \leq n)$ as index for the tag T_i in G. Therefore, a tag T_i in G has two indices $\langle G, r_{2i} \rangle$. It also generates a random number r_{1i} for T_i and then encrypts the current id and session key to generate K_i using this random number. It then generates the new tag id ID'_i and the new session key N'_i for tag T_i. Using these newly generated information and pairwise secret key S_i, it generates the encrypted update information M_{1i} and M_{2i}, and then encrypts r_{2i} using ID'_i and N'_i, and thus generates V_i. Finally, it generates the integrity information M_{3i}

Algorithm 15 A session to search an object G

Step 1: Operation in backend server

Generates a set A of n random numbers $r_{2i}, i = 1, 2, ..., n$

For each tag $T_i, i = 1, 2, ..., n$ in G, assigns r_{2i} as index and generates new session key N_i', identifier ID_i' and a random number r_{1i}

Generates $K_i \leftarrow (N_i - r_{1i}) \oplus (ID_i - r_{1i})$, $M_{1i} \leftarrow (S_i - N_i) \oplus (ID_i' - S_i)$, $M_{2i} \leftarrow (S_i - ID_i) \oplus (N_i' - S_i)$, $V_i \leftarrow (r_{2i} - N_i') \oplus ID_i'$, $M_{3i} \leftarrow (S_i - (M_{1i} \oplus r_{2i})) \oplus ((M_{2i} \oplus r_{2i}) - S_i)$

Step 2: Communication from backend server to reader

Sends $r_{1i}, K_i, M_{1i}, M_{2i}, V_i$ and M_{3i} along with A

Step 3: Communication from reader to tags

Broadcasts **request** message with $\eta_1\eta_2...\eta_n$. Here $\eta_i = r_{1i}, K_i, M_{1i}, M_{2i}, V_i, M_{3i}$

Step 4: Operation in reachable tag T_i

Retrieves $\eta_i (= r_{1i}, K_i, M_{1i}, M_{2i}, V_i, M_{3i})$ from $\eta_1\eta_2...\eta_n$

$Valid \leftarrow 0$

if $K_i = [(N_{inew} - r_{1i}) \oplus (ID_{inew} - r_{1i})]$ **then**

Calculates $ID_i' \leftarrow (M_{1i} \oplus (S_i - N_{inew})) + S_i$, $N_i' \leftarrow (M_{2i} \oplus (S_i - ID_{inew})) + S_i$, $r_{2i} \leftarrow (V_i \oplus ID_i') + N_i'$

if $M_{3i} = [(S_i - (M_{1i} \oplus r_{2i})) \oplus ((M_{2i} \oplus r_{2i}) - S_i)]$ **then**

$Valid \leftarrow 1$, Updates $N_{iold} \leftarrow N_{inew}, ID_{iold} \leftarrow ID_{inew}$ and then $N_{inew} \leftarrow N_i', ID_{inew} \leftarrow ID_i'$

else if $K_i = [(N_{iold} - r_{1i}) \oplus (ID_{iold} - r_{1i})]$ **then**

Calculates $ID_i' \leftarrow (M_{1i} \oplus (S_i - N_{iold})) + S_i$, $N_i' \leftarrow (M_{2i} \oplus (S_i - ID_{iold})) + S_i$

if $N_{inew} \neq N_i'$ **and** $ID_{inew} \neq ID_i'$ **then**

Calculates $r_{2i} \leftarrow (V_i \oplus ID_i') + N_i'$

if $M_{3i} = [(S_i - (M_{1i} \oplus r_{2i})) \oplus ((M_{2i} \oplus r_{2i}) - S_i)]$ **then**

$Valid \leftarrow 1$

Updates $N_{inew} \leftarrow N_i^{'}, ID_{inew} \leftarrow ID_i^{'}$

Step 5: Communication from tag T_i ***to reader***

if $Valid = 1$ for T_i **then**

Generates $q_i \leftarrow [(ID_i' - r_{2i}) \oplus (N_i' + r_{1i})]$ and sends $P_i \leftarrow r_{2i} \| q_i$ to reader

else

Sends a fake P_i to reader with probability λ

Step 6: Communication from reader to backend server

if r_{2i} (extracted from P_i) $\notin A$ **then**

Ignores the message

if reader finds at least threshold number (l) of valid responses **then**

For a valid tag T_i, it forwards P_i to backend server

else

Stops and reports object not found

Step 7: Operation in backend server

For tag T_i, separates q_i and r_{2i} from P_i. Locates P_i in the database using index $\langle G, r_{2i} \rangle$

if $N_i' = [q_i \oplus (ID_i' - r_{2i})] + r_{1i}$ **then**

Updates $ID_i \leftarrow ID_i', N_i \leftarrow N_i'$ and declares that the entry for T_i in record for G is valid

if at least threshold number (l) of valid entries found for G **then**

Reports search is successful

using M_{1i}, M_{2i}, r_{2i} and S_i. The backend server then sends $r_{1i}, K_i, M_{1i}, M_{2i}, V_i, M_{3i}$ for each tag in G along with A to reader in step 2. The reader then broadcasts a search request $\eta_1\eta_2...,\eta_n$, where $\eta_i = r_{1i}, K_i, M_{1i}, M_{2i}, V_i, M_{3i}$ $(i = 1, 2, ..., n)$ in step 3. The tag T_i within the communication range of reader, retrieves η_i from the broadcast information and checks the validity in step 4. To check the validity, it uses the information in the fields ID_{inew} and N_{inew} kept in its memory and after the successful verification, it extracts new id and session key ID_i' and N_i' from M_{1i} and M_{2i} respectively. It also extracts r_{2i} from V_i using the extracted ID_i' and N_i' and then verifies the correctness of extracted values using M_{3i}. If this verification is successful, it copies the information in ID_{inew} and N_{inew} to ID_{iold} and N_{iold} respectively, and updates ID_{inew} and N_{inew} using the id and session key extracted from M_{1i} and M_{2i}. Otherwise, it checks the validity again using the information in the fields ID_{iold} and N_{iold}. On successful verification, it extracts the update information and checks whether the updated information have already stored in its memory or not. If there is no match, it extracts r_{2i} from V_i and then verifies the integrity of all the extracted values. If this verification is successful, it updates the information in the fields ID_{inew} and N_{inew} using the id and session key extracted from M_{1i} and M_{2i}. If the tag T_i verifies the search information successfully either using old information or new information, it generates the authentication information P_i using r_{1i}, r_{2i}, ID_i' and N_i'. It then sends this P_i to reader in step 5 after waiting a random time interval. The tag is waiting before responding due to the fact that if all the tags respond simultaneously, the response from various tags may collide with each other. The waiting time can reduce the collision probability. The tag replies with probability λ using a fake P_i which contains a random r_{2i} in the case when it fails to successfully verify the search request. This fake response helps to fool the adversary about the existence of the desired object, i.e Information leakage attack. Therefore, greater λ value will decrease the probability of Information leakage attack. However, it will increase the useless comparison in reader and useless computation in the backend server. Therefore, 0.5 can be a balance value. However, it can be selected according to the requirement.

The reader receives P_i in step 6. If the received P_i contains a r_{2i} such that $r_{2i} \in A$ then this P_i is partially valid. The reader forwards these partially valid P_i in the same step to backend server if it obtains at least threshold number (l) of such partially valid responses.

In step 7, the backend server maps the responses using the indices $\langle G, r_{2i} \rangle$ and checks the authentication of each response. It updates the session key and id for a valid response using the id and session key generated earlier in step 1. The backend server then reports that the search is successful if it finds threshold number (l) of valid responses. Otherwise it reports that the desired object is not found.

5.3 Analysis of the proposed scheme

We analyze the proposed scheme to evaluate its applicability. In the analysis, we evaluate the security of the proposed scheme and then compare with the existing schemes. We also estimate and compare the computation, communication and storage overhead in the proposed scheme and existing schemes.

5.3.1 Security analysis

The adversary may attempt to mount the attacks utilizing the insecure communication medium between reader and object. In this section, we analyze the security of the proposed scheme.

5.3.1.1 Informal analysis

We analyze how an adversary $\mathcal{A}$ can mount the attacks specified in the threat model and how the proposed scheme handles those attacks.

Eavesdropping: During the search process, $\mathcal{A}$ listens r_{1i}, $K_i(= (N_i - r_{1i}) \oplus (ID_i - r_{1i}))$, $M_{1i}(= (S_i - N_i) \oplus (ID'_i - S_i))$, $M_{2i}(= (S_i - ID_i) \oplus (N'_i - S_i))$, $V_i(= (r_{2i} - N'_i) \oplus ID'_i)$, $M_{3i}(= (S_i - (r_{2i} \oplus M_{1i})) \oplus ((r_{2i} \oplus M_{2i}) - S_i))$ and $P_i(= r_{2i} \| (ID'_i - r_{2i}) \oplus (N'_i + r_{1i}))$, and may try to learn the secrets N_i, ID_i etc. Therefore, $\mathcal{A}$ has to separate the components in an equation which are binded using XOR operation. For example, $\mathcal{A}$ has to compute $(ID_i - r_{1i}))$ from K to obtain ID_i. However, XOR operation is secure if it is used as one time pad and the operands bit lengths are same. Shannon has proved in [**?**] that for a given ciphertext $C(= A \oplus B)$, $\mathcal{A}$ is able to obtain absolutely no additional information about A or B when A and B are of same length (bit size) and are randomly chosen[1]. The bit length of $(N_i - r_{1i})$ and $(ID_i - r_{1i}))$ are same and due to new random number r_{1i} in every search request, the components $(N_i - r_{1i})$ and $(ID_i - r_{1i}))$ are not remain same. Therefore it satisfies the one time pad condition. Similarly, the other information transmitted through insecure medium are secure. Moreover, in the proposed scheme, the threshold value concept enhances the difficulty for $\mathcal{A}$ to compromise an object which requires to eavesdrop at least one security information of each of the threshold number of tags.

[1]For i^{th} bit, $C_i = A_i \oplus B_i$. Let B_i is a random bit. Hence $\forall i, P(B_i = 0) = P(B_i = 1) = \frac{1}{2}$. Let $P(A_i = 0) = p_i$. $\therefore P(A_i = 1) = 1 - p_i$. Now, $P(C_i = 0) = P(A_i = 0) \times P(C_i = 0 | A_i = 0) + P(A_i = 1) \times P(C_i = 0 | A_i = 1) = P(A_i = 0) \times P(B_i = 0) + P(A_i = 1) \times P(B_i = 1) = p_i \times \frac{1}{2} + (1 - p_i) \times \frac{1}{2} = \frac{1}{2}$. $\therefore P(C_i = 0)$ does not depend on p_i. Conversely, we can say that the probability of obtaining correct A_i from the given C_i is $\frac{1}{2}$, where B_i is random.

Location privacy: The security related information N_i, ID_i etc. are generated randomly and updated in each successful session. Therefore, there is no relation between the $P_i(= r_{2i}\|(ID_i - r_{2i}) \oplus (N_i + r_{1i}))$ in one session and the $P_i'(= r_{2i}'\|(ID_i' - r_{2i}') \oplus (N_i' + r_{1i}'))$ in another session and hence $\mathcal{A}$ is unable to obtain any pattern from the responses. Thus, the objects are not traceable from the information communicated through insecure medium in successful sessions.

Location privacy between two successful sessions: $\mathcal{A}$ may replay the same search query r_{1i}, $K_i, M_{1i}, M_{2i}, V_i, M_{3i}$ (used in the last successful session) multiple times before the next legitimate session, and the tags are expected to respond with same P_i and thus $\mathcal{A}$ may trace an object during this period. According to the proposed scheme, tag will match the replayed information with the old information stored in its memory and then further match the updated information with the new information stored in its memory. It will find a match and the tag will respond with fake P_i using which $\mathcal{A}$ will not be able to relate the responses during the period between two successful sessions and hence cannot trace the object.

Information leakage: In traditional search process, only the tag attached to the desired object replies in response to a search query. According to this process, $\mathcal{A}$ may not be able to obtain any secure information, however, he can be able to obtain the information about its presence. In some situations, this information may be valuable to $\mathcal{A}$. In the proposed scheme, the undesired tags respond[2] with fake P_i. Since $\mathcal{A}$ does not have any secret key, he cannot distinguish between the fake response and legitimate response. Therefore $\mathcal{A}$ cannot conclude about the existence of the desired object. Also, multiple responses from the same object cannot guarantee the Information leakage attack since there can be responses from non-member tags.

Replay attack: $\mathcal{A}$ may store r_{1i}, $K_i(= (N_i - r_{1i}) \oplus (ID_i - r_{1i}))$, $M_{1i}(= (S_i - N_i) \oplus (ID_i' - S_i))$, $M_{2i}(= (S_i - ID_i) \oplus (N_i' - S_i))$, $V_i(= (r_{2i} - N_i') \oplus ID_i')$, $M_{3i}(= (S_i - (r_{2i} \oplus M_{1i})) \oplus ((r_{2i} \oplus M_{2i}) - S_i))$ and send later to tag. If the original session has completed successfully, the replayed information will be verified using N_{iold} and ID_{iold}. In another case, if $\mathcal{A}$ blocks the search request in original session and replayed the blocked request to tag, the tag will verify this request using N_{inew} and ID_{inew}. In both cases, the tag in the desired object will respond with P_i after successful verification which includes r_{1i} of original legitimate session. On a new request from the reader, $\mathcal{A}$ may try to send this P_i. However, this P_i will not be verified since the backend server has used the new r_{1i} for the new request. Also $\mathcal{A}$ cannot inject new r_{1i} since

[2] An undesired tag responds with probability λ

he does not have any secret. Moreover, the threshold value concept increases the difficulty for $\mathcal{A}$ which requires to replay the threshold number of information in order to prove the existence of the desired object.

Man-in-the-middle attack: Blocking the original $r_{1i}, K_i, M_{1i}, M_{2i}, V_i, M_{3i}$ and P_i, $\mathcal{A}$ may send fake $r_{1i}, K_i, M_{1i}, M_{2i}, V_i, M_{3i}$ and P_i through insecure medium. However, the fake information will be discarded since $\mathcal{A}$ does not have any secret i.e. N_i, ID_i and S_i, and hence he cannot send any valid information. Moreover, $\mathcal{A}$ can modify $M_{1i}, M_{2i}, V_i, M_{3i}$ without changing K_i, and try to convince the tag to update the wrong information. The tag will successfully verify K_i. However, since $\mathcal{A}$ does not have the secret key S_i, he is unable to generate a correct M_{3i}. Therefore, the tag will reject the query and respond with a fake information.

De-synchronization attack: If $\mathcal{A}$ blocks P_i, the tag contains new session key and identifier which were updated after successful verification of $r_{1i}, K_i, M_{1i}, M_{2i}, V_i, M_{3i}$. However the backend server cannot update the session key and identifier for the same tag in its database and will contain the old copies. Thus, it seems that there can be a synchronization problem. The proposed scheme avoids this problem by keeping the old session key and identifier in the tag memory. Therefore, the backend server can send a search request using the old session key and identifier and the tag can successfully verify it using the old session key and identifier stored in its memory, and update accordingly. In another case, $\mathcal{A}$ can modify M_{1i}, M_{2i}, M_{3i} and expect that the tag will retrieve and update the wrong session key N_{inew} and identifier ID_{inew}, which may cause the synchronization problem. This is not possible in the proposed scheme, because the tag will validate the retrieved secrets using M_{3i} and secret key S_i. Therefore, the modified update will not be verified successfully since $\mathcal{A}$ without knowing the secret key cannot provide a valid M_{3i}. Therefore, the tag will not update the wrong key. Hence, there is no De-synchronization threat in the proposed scheme. Even if he somehow succeeds in mounting the desynchronize attack, he requires to desynchronize at least $m - l$ number of tags in order to desynchronize an object.

Forward security: $\mathcal{A}$ may somehow be able to capture N_i and ID_i, and try to compute N_i' and ID_i'. Since he does not know S_i, he cannot compute N_i' and ID_i' from M_{1i} and M_{2i}, and since he does not know N_i' and ID_i', he cannot compute S_i. If he has either N_i or ID_i along with S_i then only he is able to compute ID_i' and N_i'. Therefore, to disrupt the forward secrecy, he has to capture S_i and either N_i or ID_i or both. To damage the forward security of an object it has to compromise at least the threshold number of tags in one session.

Backward security: In similar to forward secrecy requirement, $\mathcal{A}$ may try to disrupt the backward secrecy of the proposed scheme. However since he does not have S_i he is unable to compute N_i or ID_i or both from M_{1i}, M_{2i}. To damage the backward security of an object it has to compromise at least the threshold number of tags in one session.

Impersonation attack: $\mathcal{A}$ may physically compromise the tags for impersonation. In the proposed scheme, he has to compromise at least threshold number (l) of tags. Therefore, we do not claim that the proposed scheme is fully secure from this kind of attack. However, it increases the difficulty for $\mathcal{A}$. Hence, the proposed scheme partially fulfills this requirement.

5.3.1.2 Formal security analysis

In this section, we provide formal proofs which evaluate the security of the proposed scheme. First, we show that the proposed scheme is secure under the assumption that the adversary intercepts the information transmitted through the insecure medium in a particular session (refer as interaction). Using these information, he can try to obtain the secure information such as identifier, secret key etc. and try to implement an attack. Secondly, the adversary may intercept information transmitted in more than one sessions and try to represent these information using XOR operation to determine the secure information. We show that the proposed scheme is secure under this operation. Finally, we show that the proposed scheme is secure under the assumption that the adversary may approximate the addition/subtraction operation into XOR operation and hence can mount any attack.

Problem 1: Suppose n is computed using p and q such that $n = p \oplus q$, where length (bit size) of p, q are same. Determine the correct p, q (which are secrets) from the given n.

Hardness of problem 1: Formally, if $Adv_{\mathcal{A}}^{XOR}$ denotes an adversary $\mathcal{A}$'s advantage in finding the correct p and q from the given n, we have $Adv_{\mathcal{A}}^{XOR} = Pr[(p, q) \Leftarrow_R \mathcal{A} : p, q$ being random numbers of same length and $n = p \oplus q]$. [Note: $(x, y) \Leftarrow_R \mathcal{A}$ denotes pair (x, y) is selected randomly by the adversary $\mathcal{A}$].

$\mathcal{A}$ is allowed to be probabilistic and the probability in the advantage is computed over the random choices made by the adversary $\mathcal{A}$. If $\mathcal{A}$ can chose a correct p, he can compute a correct q from n or vice versa. The probability that $\mathcal{A}$ can choose a correct p or q is 2^{-d}, where d is the length (bit size) of p, q, n.

Definition 1: An object searching scheme (OSS) is secure if, any efficient adversary, given any one interaction (not necessarily complete) and a history of earlier interactions, cannot derive (with probability greater than $0.5 + \theta$, for a non-negligible θ) any secret.
We define a random oracle Disclose:

Disclose: This random oracle unconditionally output p, q from the input $n(= p \oplus q)$.

Theorem 5.3.1. *The proposed object searching scheme (OSS) is secure under the experiment in Algorithm 16 for deriving the secrets like session key, identifier etc. intercepting the information transmitted through an insecure medium during a particular session.*

Proof. Let the adversary $\mathcal{A}$ can be able to derive various secrets. $\mathcal{A}$ is asked to play a game and he is allowed to access the communication medium between the tags and the reader. He is given the information transmitted through insecure medium in a particular session as a challenge. He can be able to win the game if he is successful to compute the secrets. He does an experiment depicted in Algorithm 16. We define the success probability for *EXP* $1^{XOR}_{\mathcal{A},OSS}$ as

Algorithm 16 *EXP* $1^{XOR}_{\mathcal{A},OSS}$

1: Intercepts search request $< K_i, r_{1i}, M_{1i}, M_{2i}, V_i, M_{3i} >$ where, $K_i \leftarrow (N_i - r_{1i}) \oplus (ID_i - r_{1i}), M_{1i} \leftarrow (S_i - N_i) \oplus (ID'_i - S_i), M_{2i} \leftarrow (S_i - ID_i) \oplus (N'_i - S_i), V_i \leftarrow (r_{2i} - N'_i) \oplus ID'_i, M_{3i} \leftarrow (S_i - (M_{1i} \oplus r_{2i})) \oplus ((M_{2i} \oplus r_{2i}) - S_i)$
2: Call Disclose on input K_i and obtains $(N_i - r_{1i}), (ID_i - r_{1i}) \leftarrow Disclose(K_i)$
3: Calculates $ID_i \leftarrow (ID_i - r_{1i}) + r_{1i}, N_i \leftarrow (N_i - r_{1i}) + r_{1i}$
4: Call Disclose on input M_{1i} and obtains $(S_i - N_i), (ID'_i - S_i) \leftarrow Disclose(M_{1i})$
5: Calculates $S_i \leftarrow (S_i - N_i) + N_i, ID'_i \leftarrow (ID'_i - S_i) + S_i$
6: Call Disclose on input M_{2i} and obtains $(S_i - ID_i), (N'_i - S_i) \leftarrow Disclose(M_{2i})$
7: Calculates $N'_i \leftarrow (N'_i - S_i) + S_i$
8: Calculates $(r_{2i} - N'_i) \leftarrow V_i \oplus ID'_i$ and then $r_{2i} \leftarrow (r_{2i} - N'_i) + r_{2i}$
9: Calculates $(S'_i - (M_{1i} \oplus r_{2i})) \leftarrow M_{3i} \oplus ((M_{2i} \oplus r_{2i}) - S_i)$ and then $S'_i \leftarrow (S'_i - (M_{1i} \oplus r_{2i})) + M_{1i} \oplus r_{2i}$
10: **if** $S_i ! = S'_i$ **then**
11: return 0 (Failure)
12: **else**
13: Successful to intercept $ID_i, N_i, r_{2i}, S_i, ID'_i, N'_i$

Succ $1^{XOR}_{\mathcal{A},OSS} = Pr[$*EXP* $1^{XOR}_{\mathcal{A},OSS} = 1]$. Then the advantage of the experiment *EXP* $1^{XOR}_{\mathcal{A},OSS}$ is given by $Adv^{XOR}_{\mathcal{A},OSS} = max_{\mathcal{A}}[$*Succ* $1^{XOR}_{\mathcal{A},OSS}]$, where the maximum taken over all adversaries.

According to the experiment depicted in Algorithm 16, if an adversary is able to solve the XOR operation (problem 1), he can be able to calculate all the secrets and win the game. However, from the hardness of problem 1, we have $Adv^{XOR}_{\mathcal{A},OSS} \leq \epsilon$ where $0 < \epsilon \leq 2^{-d}$. Hence the proposed scheme is secure from intercepting the secure information. □

Corollary 5.3.2 (Theorem 5.3.1)**.** *The proposed scheme is secure against the adversary to mount any attack described in the threat model based on the information communicated in a particular session and the information communicated in the next session.*

Proof. The adversary $\mathcal{A}$ which obtains various secrets using the experiment depicted in Algorithm 16 is further permitted to access the communication medium between reader and tags during the next session when the reader searches for the same object. $\mathcal{A}$ accesses these information as challenge and tries to mount various attacks. Since $\mathcal{A}$ has the information ID_i, N_i, r_{2i}, S_i,

Algorithm 17 *EXP* $2^{attacks}_{\mathcal{A},OSS}$

1: Adversary $\mathcal{A}$ has the information $ID_i, N_i, r_{2i}, S_i, ID'_i, N'_i$
2: Intercepts $< P_i >$ during the same session depicted in Algorithm 16 and detaches $r'_{2i}, (ID'_i - r_{2i}) \oplus (N'_i + r_{1i})$
3: **if** $r_{2i} = r'_{2i}$ **then**
4: Information leakage is successful
5: Intercepts $< K^1_i, r^1_{1i}, M^1_{1i}, M^1_{2i}, V^1_i, M^1_{3i} >$ in the next session, where, $K^1_i \leftarrow (N'_i - r^1_{1i}) \oplus (ID'_i - r^1_{1i}), M^1_{1i} \leftarrow (S_i - N'_i) \oplus (ID''_i - S_i), M^1_{2i} \leftarrow (S_i - ID'_i) \oplus (N''_i - S_i), V^1_i \leftarrow (r^1_{2i} - N''_i) \oplus ID''_i, M^1_{3i} \leftarrow (S_i - (M^1_{1i} \oplus r^1_{2i})) \oplus ((M^1_{2i} \oplus r^1_{2i}) - S_i)$
6: Calculates $ID'^1_i \leftarrow (M^1_{1i} \oplus (S_i - N'_i)) + S_i, N'^1_i \leftarrow (M^1_{2i} \oplus (S_i - ID'_i)) + S_i,$
7: $r^1_{2i} \leftarrow (V^1_i \oplus ID'^1_i) + N'^1_i, S''_i \leftarrow (M^1_{3i} \oplus ((M^1_{2i} \oplus r^1_{2i}) - S_i)) + (M^1_{1i} \oplus r^1_{2i})$
8: **if** $S_i = S''_i$ **then**
9: Location traceability is successful and Forward secrecy is disrupted

ID_{inew}, N_{inew}, he can successfully mount the attacks such as eavesdropping, replay attack, man-in-the-middle attack, Location privacy between two successful sessions. $\mathcal{A}$ does the experiment depicted in Algorithm 17 to mount the attacks such as Information leakage, Location privacy. He will also get confirmation of attacks against Forward secrecy and Backward secrecy using the same experiment. Clearly, he can successfully mount these attacks. However, the success probability of the experiment in Algorithm 17 depends on the probability of accessing the various secrets. Therefore, the success probability of the experiment depicted in Algorithm 16 entails the success probability of the adversary $\mathcal{A}$ to mount various attacks described in the threat model. Hence the proposed scheme is secure against the attacks described in the threat model. □

Theorem 5.3.3. *The proposed object searching scheme (OSS) is secure under the experiment in Algorithm 18.*

Proof. The adversary $\mathcal{A}$ accesses the information transmitted in multiple sessions listed in $\mathcal{L}$ and tries to derive the secrets using XOR operation. Each equation in $\mathcal{L}$ consists of two components which are binded by XOR operation. The number of such unique components in $\mathcal{L}$ is 34, whereas the number of equations is 18. Therefore, there are two pair of equations which contains common components and are dependent. $\mathcal{A}$ performs the experiment depicted in Algorithm 18 to find those equations and then performs the XOR operation on the pairs. The list $\mathcal{L}$ consists of the equations for the parameters transmitted through insecure medium during a particular successful session Ses_i and in the next successful session Ses_{i+1}.

$$\mathcal{L} = \left\{ \begin{array}{l} \text{Equations in successful session Ses}_i \\ \hline K_i \leftarrow (N_i - r_{1i}) \oplus (ID_i - r_{1i}) \\ M_{1i} \leftarrow (S_i - N_i) \oplus (ID'_i - S_i) \\ M_{2i} \leftarrow (S_i - ID_i) \oplus (N'_i - S_i) \\ V_i \leftarrow (r_{2i} - N'_i) \oplus ID'_i \\ M_{3i} \leftarrow (S_i - (M_{1i} \oplus r_{2i})) \oplus ((M_{2i} \oplus r_{2i}) - S_i) \\ q_i = (ID'_i - r_{2i}) \oplus (N'_i + r_{1i}) \\ \text{Equations in sucessful session Ses}_{i+1} \\ \hline \\ K_i^1 \leftarrow (N'_i - r_{1i}^1) \oplus (ID'_i - r_{1i}^1) \\ M_{1i}^1 \leftarrow (S_i - N'_i) \oplus (ID''_i - S_i) \\ M_{2i}^1 \leftarrow (S_i - ID'_i) \oplus (N''_i - S_i) \\ V_i^1 \leftarrow (r_{2i}^1 - N''_i) \oplus ID''_i \\ M_{3i}^1 \leftarrow (S_i - (M_{1i}^1 \oplus r_{2i}^1)) \oplus ((M_{2i}^1 \oplus r_{2i}^1) - S_i) \\ q_i^1 = (ID''_i - r_{2i}^1) \oplus (N''_i + r_{1i}^1) \\ \text{Equations in an unsuccessful session between Ses}_i\text{and Ses}_{i+1} \\ \hline \\ K_i^2 \leftarrow (N_i - r_{1i}^2) \oplus (ID_i - r_{1i}^2) \\ M_{1i}^2 \leftarrow (S_i - N_i) \oplus (ID'''_i - S_i) \\ M_{2i}^2 \leftarrow (S_i - ID_i) \oplus (N'''_i - S_i) \\ V_i^2 \leftarrow (r_{2i}^2 - N'''_i) \oplus ID'''_i \\ M_{3i}^2 \leftarrow (S_i - (M_{1i}^2 \oplus r_{2i}^2)) \oplus ((M_{2i}^2 \oplus r_{2i}^2) - S_i) \\ q_i^2 = (ID'''_i - r_{2i}^2) \oplus (N'''_i + r_{1i}^2) \end{array} \right\}$$

It also contains the equations for the parameters transmitted through the insecure medium during

an unsuccessful session between Ses_i and Ses_{i+1}. Algorithm 18 takes the list of equations $\mathcal{L}$ as

Algorithm 18 *EXP 3*

Input: The list $\mathcal{L}$

Output: Modified $\mathcal{L}$

for each pair of equations in $\mathcal{L}$ **do**

if the pair has common component **then**

Apply XOR operation on the pair and obtain a new equation $\mathcal{E}'$

if $\mathcal{E}' \notin \mathcal{L}$ **then**

Include $\mathcal{E}'$ into $\mathcal{L}$

end for

input and finds a pair of equation which have one or more common components binded by XOR operation. It performs a XOR operation on this pair and derives a new equation. It includes the derived equation into $\mathcal{L}$ if there is no such equation in $\mathcal{L}$ which is similar to the derived equation. It selects this pair because existence of common components can help to have less number of components in the resultant equation and that may help to reveal one or more secret information. However, if the pair does not have any common component, the XOR operation will increase the difficulty for $\mathcal{A}$ to extract the secret information. Input of the experiment *EXP 3* is the list $\mathcal{L}$ and it modifies $\mathcal{L}$ by adding two new equations as follows:

$$\mathcal{L} = \mathcal{L} \bigcup \left\{ \begin{array}{l} E_{M_{1i} \oplus M_{1i}^2} = (ID_i' - S_i) \oplus (ID_i'' - S_i) \\ E_{M_{2i} \oplus M_{2i}^2} = (N_i' - S_i) \oplus (N_i'' - S_i) \end{array} \right\}$$

$\mathcal{A}$ can determine the secrets by solving any one equations in the modified $\mathcal{L}$. However, the probability of success depends on the hardness of problem 1. Therefore the proposed scheme is secure. □

XOR-approximation: An equation $H = C + D - F$ (C, D, F are three random numbers of size d bits) can be approximated to an equivalent equation $H' = C \oplus D \oplus F$ with probability $(\frac{3}{4})^{d-1}$ [**?**] by replacing bitwise +/- with bitwise XOR operation[3].

[3]For a given equation $H = C + D - F$ (C, D, F are three random numbers of size d bits), we replace +/- operation with XOR operation to obtain a new equation $H' = C \oplus D \oplus F$. The LSB of H' is same as LSB of H since there is no carry or borrow input bit in LSB. However there can be carry/borrow input bit in other bits and maximum probability that i^{th} bit of H is equals to the i^{th} bit of H' is $\frac{3}{4}$ [**?**]. Therefore, the probability of $H = H'$ is $(\frac{3}{4})^{d-1}$.

Theorem 5.3.4. *Probability that an adversary $\mathcal{A}$ can mount various attacks mentioned in the threat model using XOR-approximation is bound above by* $(\frac{3}{4})^{d-1}$

Table 5.2: Success probability on various d vales

	$d(1)$	$d(2)$	$d(32)$	$d(64)$	$d(96)$	$d(128)$
ζ	1	7.5×2^{-3}	1.339366×2^{-13}	1.345425×2^{-27}	1.351512×2^{-40}	1.357627×2^{-53}

Proof. The equations used in the proposed scheme has +/- operation and $\mathcal{A}$ can mount various attacks with respect to the experiments in Algorithm 2, 3, 4 after performing the XOR-approximation on the equations. The probability that he can successfully approximate an equation is $\zeta = (\frac{3}{4})^{d-1}$, where d is the bit size of each parameter. Therefore, the proposed object searching scheme is secure for sufficiently small ζ and large d. Table 5.2 shows various ζ values. Therefore, depending on the security requirement, an appropriate d (bit size) needs to be chosen to implement the proposed scheme. □

5.3.2 Comparison

We have chosen four parameters, namely, security, computation, communication and storage overhead for comparing the proposed scheme with the existing schemes.

5.3.2.1 Security comparison

In this section, we compare the robustness of the proposed scheme and the existing schemes. According to Table 5.3, the proposed scheme satisfies all the security requirements except the impersonation attack. However, use of multiple tags in every object increases the difficulty for $\mathcal{A}$ to mount this attack. The other schemes [**?**] [**?**] [**?**] [**?**] [**?**] do not satisfy all the security requirements and hence in security aspect, the proposed scheme is more secure.

5.3.2.2 Computation overhead

Table 5.4 shows the comparative study of various operations performed in [**?**] [**?**] [**?**] [**?**] [**?**] and in the proposed scheme. The desired tag in the proposed scheme does not require to execute any computationally expensive operations such as random number generation and hash operation. However, it requires to compute maximum number of lightweight operations i.e, XOR and addition/subtraction. Though the hardware advances enables the tag to use complex operations such

Table 5.3: Security assurance

	a	b	c	d	e	f	g	h	i	j
Tan et al. Protocol 1 [?]	N	N	N	N	N	Y	Y	N	N	Y
Tan et al. Protocol 2 [?]	N	N	N	N	N	Y	Y	N	N	Y
Tan et al. Protocol 3 [?]	Y	N	N	N	N	Y	Y	Y	P	Y
Tan et al. Protocol 4 [?]	N	N	N	N	N	Y	Y	N	N	Y
Kulseng et al. Protocol 1 [?]	Y	Y	Y	N	Y	N	N	Y	N	N
Kulseng et al. Protocol 2 [?]	Y	Y	Y	N	Y	N	N	Y	N	Y
Kulseng et al. Protocol 3 [?]	Y	Y	Y	N	Y	N	N	Y	Y	Y
Hoque et al. [?]	Y	N	Y	N	Y	N	N	Y	Y	Y
Yoon et al. [?]	Y	N	Y	N	Y	P	P	Y	N	Y
Zheng et al. [?]	Y	N	Y	Y	N	Y	Y	Y	Y	Y
Proposed scheme	Y	P	Y	Y	Y	Y	Y	Y	Y	Y

a: Eavesdropping, b: Impersonation attack, c: Location privacy, d: Location privacy between two successful sessions, e: Man-in-the-middle attack, f: Forward security, g: Backward security, h: Replay attack, i: Information leakage, j: De-synchronization attack, Y: Satisfy, N: Not satisfy, P: Partially satisfy

Table 5.4: Number of operations performed in various scheme

	Desired Tag						Undesired tag						Reader						Backend Server					
	a	b	c	d	e	f	a	b	c	d	e	f	a	b	c	d	e	f	a	b	c	d	e	f
Tan et al. Protocol 1 [?]	2	3	1	3	0	0	1	2	0	2	0	0	2	2	1	3	0	0	0	0	0	0	0	0
Tan et al. Protocol 2 [?]	2	3	1	3	0	0	1	2	0	2	0	0	2	2	1	3	0	0	0	0	0	0	0	0
Tan et al. Protocol 3 [?]	1	1	1	2	0	0	1	1	1	2	0	0	f_1	f_1	1	$2f_1$	0	0	0	0	0	0	0	0
Tan et al. Protocol 4 [?]	2	3	3	1	0	0	1	2	2	2	0	0	$1+f_1$	$1+f_1$	1	$1+2f_1$	0	0	0	0	0	0	0	0
Kulseng et al. Protocol 1 [?]	5	6	0	0	0	0	2	1	0	0	0	0	5	4	1	0	0	0	0	0	0	0	0	0
Kulseng et al. Protocol 2 [?]	4	5	0	0	0	0	2	1	0	0	0	0	4	3	1	0	0	0	0	0	0	0	0	0
Kulseng et al. Protocol 3 [?]	4	5	0	0	0	0	2	1	2	0	0	0	$4f_1$	$3f_1$	1	0	0	0	0	0	0	0	0	0
Hoque et al. [?]	3	3	1	0	1	0	2	2	0	0	1	0	2	3	0	0	0	0	1	1	1	0	1	0
Yoon et al. [?]	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
Zheng et al. [?]	0	k_1+k_2	0	0	0	0	k_1	0	0	0	0	0	0	1	0	0	0	0	0	k_1+k_2	2	0	0	0
Proposed scheme	9	0	0	1	0	13	2	0	1	0	0	4	0	0	0	0	0	0	$7n+\delta+\alpha$	0	$4n$	$\delta+\alpha$	0	$9n+2(\delta+\alpha)$

a: XOR, b: Hash, c: Random number generation, d: Attachment/Detachment, e: Modulas, f: Addition/Subtraction, f_1: Number of tags replied, INC: Increment, k_1, k_2: Number of hash functions used in bloom filter, α: Number of false positive response, δ: Number of valid response to backend server, n: Number of tags attached to an object

as cryptographic hash functions, we have tried to use elementary operations (XOR, addition, subtraction) to keep the overall computation as less as possible. This will consume less power and can help the tag to perform other non-security related operations. The other schemes require to compute one or more hash operations and many random number generation operations which

are expensive in nature. Therefore, the desired tags in the proposed scheme are efficient in compared to other schemes. The undesired tag in the proposed scheme needs to generate only one random number. However, it does not require to compute any hash operations. Therefore, the undesired tags in the proposed scheme are also efficient compared to other schemes. The reader in the proposed scheme does not require to compute any operation while the reader in the other schemes require to compute many operations. The backend server in the proposed scheme does not require to compute any hash operation. It requires to compute more number of basic operations compared to other schemes since it has to process information for more than one desired tag for a desired object.

5.3.2.3 Computation overhead due to fake responses

Due to fake response from the tags in undesired objects, the backend server has to distinguish the legitimate responses. Therefore, it requires to process all the responses it receives which introduce an useless computation overhead.

Probability of useless computation in the proposed scheme: The reader partially checks r_{2i} in each response for its validity. The tags in the desired object extracts a valid r_{2i} and sends it to the reader. Therefore, this response is validated and forwarded to the backend server. The reader validates simply by comparing the received r_{2i} with the random numbers in A. Therefore, there is no such computational overhead in it. However, a tag in the undesired object may send a fake response with a random r_{2i} such that this $r_{2i} \in A$. Therefore, any response with such kind of r_{2i} would be forwarded to the backend server and the backend server has to process those fake responses. This introduces an useless computational overhead. The probability of such useless computation in the proposed scheme is $\frac{\lambda n M_{iv}}{M_v 2^b + \lambda n M_{iv}}$, where n is the number of valid r_{2i}, b is the number of bits constitutes a r_{2i}, M_{iv} is the number of invalid tags, M_v is the number of valid tags, and λ is the probability that an invalid tag will respond with fake information.

Probability of useless computation in other schemes [**?**] [**?**] [**?**]: To prevent information leakage attack, the other schemes also suffer from useless computation overhead. The probability of such overhead in these schemes is $\frac{\lambda M_{iv}}{1+\lambda M_{iv}}$.

Comparison: Table 5.5 illustrates the probability of useless computation for $n = 8, M_v = 8, \lambda = \frac{1}{2}$ and various number of invalid tags (M_{iv}). (We choose $\lambda = \frac{1}{2}$ since this makes a balance as we have mentioned in Section 5.2.2.2.). The probability that the backend server has to process useless response in the proposed scheme is only 0.0007 for 100 invalid tags. However,

Table 5.5: Probability of useless computation in backend server

Number of invalid tags	10	20	30	40	50	60	70	80	90	100
Proposed scheme	0.000076	0.000153	0.000229	0.000305	0.000381	0.000458	0.000534	0.000610	0.000686	0.000762
Other schemes	0.833333	0.909091	0.937500	0.952381	0.961538	0.967742	0.972222	0.975610	0.978261	0.980392

the probability of this overhead in other schemes is 0.9803 for the same number of invalid tags. Therefore, the proposed scheme prevents information leakage attack with negligible probability of useless computation in backend server.

Table 5.6: Communication overhead of various scheme

	Desired tag	Undesired tag	Reader	Backend Server
Tan et al. Protocol 1 [**?**]	5	3	5	0
Tan et al. Protocol 2 [**?**]	5	3	5	0
Tan et al. Protocol 3 [**?**]	5	3	$3 + 2f_1$	0
Tan et al. Protocol 4 [**?**]	5	3	$3 + 2f_1$	0
Kulseng et al. Protocol 1 [**?**]	5	3	5	0
Kulseng et al. Protocol 2 [**?**]	4	2	4	0
Kulseng et al. Protocol 3 [**?**]	4	2	$2(f_1 + 1)$	0
Hoque et al. [**?**]	$f + 5$	$f + 3$	$f + k + 6$	3
Yoon et al. [**?**]	6	3	12	6
Zheng et al. [**?**]	$k_2 + 8$	4	$15 + k_2 + f_1$	8
Proposed scheme	7	7	$13n + \delta + \alpha + f_1$	$7n + \delta + \alpha$

f: Length of Bit Record(BR) in Hoque et al., k: Number of single or collied reply, f_1: Number of tags replied, k_2: Number of slots in Zheng et al., δ: Number of valid responses to backend server, α: Number of false positive responses, n: Number of tags attached to an object

5.3.3 Communication overhead

In this section, we analyze and compare the communication overhead of the proposed scheme and the existing schemes. According to Table 5.6, the communication overhead for desired or undesired tags in the proposed scheme is almost equals to the schemes in [**?**] [**?**] [**?**]. However, it is less than the schemes in [**?**] [**?**]. The communication overhead in reader and backend server is more than the other schemes since an object in the proposed scheme is attached with multiple number of tags.

5.3.4 Storage overhead

The RFID tags has limited memory and hence we analyze and compare the storage requirement of the proposed scheme and the existing schemes.

Table 5.7: Storage requirement

	Tag	Reader	Backend Server
Tan et al. Protocol 1 [?]	2	$2\beta + 1$	$3\beta + 1$
Tan et al. Protocol 2 [?]	$2 + l$	$2\beta + 1$	$3\beta + 1$
Tan et al. Protocol 3 [?]	2	$2\beta + 1$	$3\beta + 1$
Tan et al. Protocol 4 [?]	2	$2\beta + 1$	$3\beta + 1$
Kulseng et al. Protocol 1 [?]	2	4β	0
Kulseng et al. Protocol 2 [?]	3	3β	0
Kulseng et al. Protocol 3 [?]	3	3β	0
Hoque et al. [?]	2	2β	2β
Yoon et al. [?]	3	0	$2\beta + 1$
Zheng et al. [?]	1	0	β
Proposed scheme	5	0	$3\beta n$

l: Number of completed sessions, β: Number of objects n: Number of tags in each object

Table 5.7 shows that the requirement of storage in the tag for the proposed scheme is more since it requires to keep old information to prevent de-synchronization attack. The storage requirement in backend server is more due to the fact that an object is attached with multiple number of tags and hence it has to keep information about all the tags attached to an object. There is no storage requirement in the reader for the proposed scheme whereas almost all the other schemes have this requirement.

5.4 Conclusion

RFID technology can help to search an object efficiently. In order to increase the detection probability, multi-tag environment can be adopted. In this chapter, we demonstrate that multiple number of tags in an object also can be utilized to increase the difficulty for the adversary to mount certain kind of attacks during a searching process. The proposed scheme is lightweight since the RFID tag needs to compute a few basic operations and only one random number generation operation. The proposed scheme is evaluated through a proper analysis which confirms that it can prevent the possible attacks. The useless computation due to fake responses is negligible in the proposed scheme and much less in comparison to the existing schemes. The searching process can be extended to another application which requires to find a set of relevant objects which are coexisting and can be accessed simultaneously. A proof of such coexistence can also

help to perform a certain tasks in future. In the next chapter (Chapter 6), we address this problem and propose a solution to this problem.

Chapter 6

Generation of a Proof of Coexistence of Multiple Objects in Multi-tag Arrangement

Sometime two or more objects need to coexist which may serve the due purpose of an event. A proof may assure the coexistence of two or more relevant objects. The assurance can help to execute the event without any error. For example, an automated assembling unit may prepare an instrument in a sequence of various stages and the assembling process in one stage may depend on the successful assembling process in previous stage. Therefore, this stage needs to verify the correctness of the parts which has assembled in previous stages. A coexistence proof of the parts assembled in the previous stages can assure and help to perform the task assigned in the current stage. Radio Frequency Identification (RFID) technology can help to generate the coexistence proof. In this technology, assembled parts are attached with RFID tags with unique ids. The RFID reader generates a request and this request is sign by a tag. The signed version response is resigned by another tag and this process is repeated until the all the desired tags put signatures. The RFID reader intermediates and generates the coexistence proof. This proof is then verified and an appropriate action is taken at the current stage. Thus there are many applications which needs to generate a coexistence proof using RFID technology. However, many security implications during the proof generation process are involved in RFID technology such as eavesdropping, man-in-the-middle attack, replay attack, location privacy, location privacy between two successful sessions, denial of proof, forward secrecy and backward secrecy, denial of service, physical attack, and relay attack. Existing proof generation schemes have looked into some of these security implications and assume that every objects are attached with single RFID tag. However, the detection probability of an object can be increased using multiple number of

tags in the object. The existing schemes are not applicable to this multi-tag arrangement. This is because the signature of one tag can be re-signed by another tag attached to the same object whereas this is expected to be signed by a tag attached to another relevant object. Thus proof generation process will be disturbed. Therefore, the challenge is to securely generate a concrete proof of coexistence for a set of relevant objects where each object is attached with multiple number of RFID tags. In this chapter, we describe our proposed coexistence proof generation protocol which is lightweight and fulfills the possible security requirements. This is also applicable to multi-tag environment. Necessary analysis has been carried out in this paper to evaluate the proposed scheme.

6.1 Communication model

Fig. 6.1 shows the communication model we follow in this paper which consists of two objects G_1, G_2, a RFID reader, an online server, and a verifier. Each object is attached with multiple number of RFID tags in proper alignment as described in [**?**]. A workstation having a database of all the RFID tags acts as online server. RFID reader acts as an intermediary between online server and the tags i.e., objects. The online server itself can act as a verifier or there can be an additional workstation which can act as verifier. We assume that the tags are passive and hence they cannot initiate any communication. The reader can communicate to tags scattering electromagnetic signal and the tags attached in the objects can respond by backscattering to this signal. We assume that this communication is insecure. The reader also can communicate to online server using wired or wireless communication. We assume that this communication is secure. If online server itself does not act as verifier there is a communication link between online server and verifier. We assume that this communication is also secure.

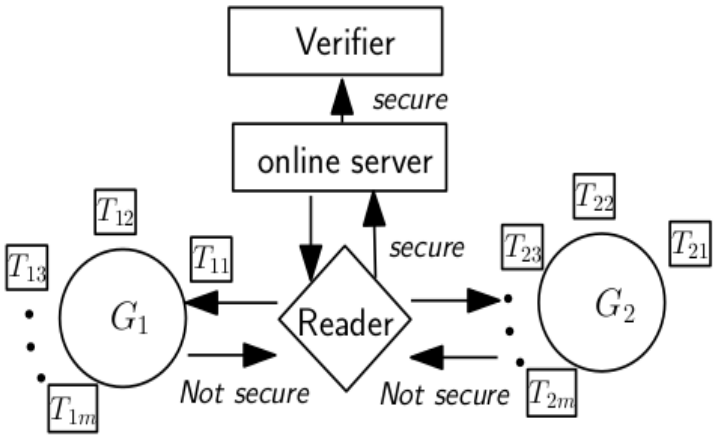

Figure 6.1: Communication model for the proposed coexistence proof generation protocol

6.2 Proof generation protocol

Based on the communication model mentioned in Section 6.1 and the threats associated to this application, we propose a proof generation protocol. We first introduce the data structure required to keep various information in the components we have mentioned in section 6.1.

6.2.1 Database

A RFID tag T_{ji} attached in object G_j contains a nonce value $Nonce_{ji}$ in its first field. The second field contains the secret key S_{ji}, the third and fourth fields contain the new and old session keys N_{jinew} and N_{jiold} respectively. Fifth and sixth fields contain the new and old identifiers ID_{jinew}, ID_{jiold} respectively.

$Nonce_{ji}$	S_{ji}	N_{jinew}	N_{jiold}	ID_{jinew}	ID_{jiold}

Tag database

We keep information about each object in a database in online server. There is a table which contains the records for each object. We assume an object is attached with m number of tags.

T_{j1}	T_{j2}	...	T_{jm}
S_{j1}, N_{j1}, ID_{j1}	S_{j2}, N_{j2}, ID_{j2}	...	S_{jm}, N_{jm}, ID_{jm}

Record for object G_j in online server

The record for object G_j contains information for each tag attached in G_j. For each tag $T_{ji}, i = 1, 2, ..., m$, the first field contains secret key S_{ji}, the second and third fields contain the session key N_{ji} and identifier ID_{ji} respectively.

6.2.2 Proposed protocol

We describe our scheme in context of a set of two relevant objects $\langle G_1, G_2 \rangle$. However, our scheme can be extended to more than two objects. The protocol has three phases, namely, Setup, Proof generation and Proof verification.

6.2.2.1 Setup phase

In Setup phase the tags are assigned to objects where each object is assigned multiple number of tags and various keys and ids are preloaded into the tags as follows:

- For each tag T_{ji} assigned in object G_j, a session key N_{ji} and an id ID_{ji} are generated randomly. N_{ji} is loaded into both the fields N_{jinew}, N_{jiold} in tag memory. ID_{ji} is also loaded into both the fields ID_{jinew}, ID_{jiold} in tag memory. There is a record for object G_j in the database in online server. Both N_{ji} and ID_{ji} are loaded into the corresponding fields.

- For each tag T_{ji} assigned in object G_j, a pair-wise secret key S_{ji} is generated and loaded into the corresponding field in tag memory. This is also loaded into S_{ji} field for tag T_{ji} in the record for object G_j in the database of online server.

6.2.2.2 Proof generation phase

In this phase, a session is executed and a coexistence proof for a schedule $\langle G_1, G_2 \rangle$ is generated. Here a schedule means a set of two relevant objects due for a certain purpose. There can be many schedules and a pair of schedules may have common objects. We assume that one session handles the proof generation process for only one schedule. We also assume that two sessions for two different schedules having a common object cannot be executed simultaneously. Figure 6.2 illustrates a session of proof generation process in the proposed scheme. The proof generation phase has three sub phases as shown in Figure 6.2. For the sake of clarity, we divide the whole protocol into a number of sub-protocols. Algorithm 20 is the main protocol which uses Algorithm 19, 21, 22 to perform the proof generation task. Algorithm 19 is executed in online server to generate search information for the objects in the given schedule. Tag executes Algorithm **??**

to verify the search request from RFID reader and responds with authentication information. Algorithm **??** is executed in the online server to verify the responses from the tags and updates the database accordingly. We use a lightweight *MAC* operation to generate the signatures and the authentication information. We also use a Pseudo Random Number Generator (*PRNG*) function to update the session key and id.

Algorithm 19 executed by online server to generate search information

for each tag in G_j **do**
  Generates new id and new session key
  $ID'_{ji} \| N'_{ji} \leftarrow PRNG((S_{ji} - ID_{ji}) \| (N_{ji} - S_{ji}))$
  Computes $K_{ji} \leftarrow MAC(ID_{ji}, r_j)$
  Sends r_j, K_{ji} to reader
end for

6.2.2.3 Brief description of the protocol

The proof generation process is divided into three sub phases. The steps 1 to 3 are executed in sub phase 1. The steps 4 to 6 executed in sub phase 2 and the rest of the steps are executed in sub phase 3. In Step 1, the online server selects one object G_1 randomly from two relevant objects of a schedule $\langle G_1, G_2 \rangle$. It then generates a random number r_1, new session key N'_{1i}, new id ID'_{1i} for each tag $T_{1i}, (i = 1, 2, ..., m)$ attached to G_1 and then adds r_1 to proof *PR*. It then sends the search information r_1, K_{1i} for G_1 to reader. It uses a lightweight *MAC* operation to generate K_{1i}. The reader broadcasts r_1, K_{1i}.

In Step 2, the tag attached to the desired object successfully verifies the authentication of search information and then responds with P_{1i}. It also updates the id and session key stored in its memory. The online server verifies (in Step 3) the authentication of each response if it receives the responses within a time period Δ after sending the search information for G_1, and updates the id and session key using the newly generated id and session key for the tags whose responses are valid. If there is no valid response for object G_1, the online server declares that the proof generation has failed and stops. If it obtains $l(1 \leq l)$ number of valid responses then it adds G_1, N'_{1i}, TS (*TS* is an encrypted value of current time) to the proof *PR*.

Algorithm 20 to generate coexistence proof

Step 1: Broadcasts the search information for G_1

Online server generates a random number r_1 and a time stamp $TS \leftarrow F(K_{OS}, Current\_time)$, K_{OS} is the secret of online server

Adds r_1, TS into the proof *PR*

Sets $j \leftarrow 1$ and executes Algorithm 19

Reader broadcasts $r_1, K_{1i}, i = 1, 2, ..., m$

Step 2: Verifies and responds the authentication information

Sets $j \leftarrow 1$ and executes Algorithm 21

Reader forwards P_{1i} to online server

Step 3: Verification in online server

Receives P_{1i}

If time between Step 1 and Step 3 is less than Δ **then**

Sets $j \leftarrow 1$ and executes Algorithm 22

if does not find any valid entry for G_1 **then**

Reports proof generation is unsuccessful and stops

Step 4: Broadcasts the search information for G_2

The online server generates $r_2 \leftarrow MAC(P_{11} \oplus P_{12} \oplus ... \oplus P_{1l} \oplus TS, \beta_1)$, ($l$ indicates the number of valid responses and $\beta_1 = S_{11} \oplus S_{12} \oplus ... \oplus S_{1m}$)

Sets $j \leftarrow 2$ and execute Algorithm 19

Reader broadcasts $r_2, K_{2i}, i = 1, 2, ..., m$

Step 5: Verifies and responds the authentication information

Sets $j \leftarrow 2$ and executes Algorithm 21

Reader forwards P_{2i} to online server

Step 6: Verification in online server

Receives P_{2i}

If time between Step 4 and Step 6 is less than Δ **then**

Sets $j \leftarrow 2$ and execute Algorithm 22

if does not find any valid entry for G_2 **then**

Reports proof is unsuccessful and stops

Step 7: Signature request for G_1

The online server generates $r_3 \leftarrow MAC(P_{21} \oplus P_{22} \oplus ... \oplus P_{2l'}, \beta_2)$, ($l'$ indicates the number of valid responses and $\beta_2 = S_{21} \oplus S_{22} \oplus ... \oplus S_{2m}$)

For the tags in G_1, which responded with valid information, the online server generates $T_{1i} \leftarrow MAC(ID'_{1i}, r_3)$

Reader broadcasts T_{1i}, r_3

Step 8: Signature generation in i^{th} ***tag in*** G_1

if $MAC(ID_{inew}, r_3) = T_{1i}$ **then**

$F_{1i} \leftarrow MAC(N_{inew}, r_3)$

Sends F_{1i} to reader

Reader submits $signature = \{F_{1i}\}$

Step 9: Signature collection in online server

Adds *signature* to *PR*

Algorithm 21 executed by tag T_{ji} to verify request and then respond

if $MAC(ID_{jinew}, r_j) = K_{ji}$ **then**
  Computes new id and new session key $ID'_{ji} \| N'_{ji} \leftarrow PRNG((S_{ji} - ID_{jinew}) \| (N_{jinew} - S_{ji}))$
  Computes $P_{ji} \leftarrow MAC(N'_{ji}, r_j)$
  Sends P_{ji} to reader
  Updates $N_{jiold} \leftarrow N_{jinew}, ID_{jiold} \leftarrow ID_{jinew}$ and then $N_{jinew} \leftarrow N'_{ji}, ID_{jinew} \leftarrow ID'_{ji}, Nonce_{ji} \leftarrow r_j$
else
  if $r_j ! = Nonce_{ji}$ **then**
    if $K_{ji} = MAC(ID_{jiold}, r_j)$ **then**
      Sends $P_{ji} \leftarrow MAC(N_{jinew}, r_j)$ to reader
      Updates $Nonce_{ji} \leftarrow r_j$

Algorithm 22 executed by online server to check validity of a response

if $P_{ji} = MAC(N'_{ji}, r_j)$ **then**
  The entry for tag T_{ji} in record for G_j is valid
  Adds $\{G_j, N'_{ji}\}$ to proof *PR*
  Updates $ID_{ji} \leftarrow ID'_{ji}, N_{ji} \leftarrow N'_{ji}$

Tag T_{ji} in object G_j		**Reader**		**Online server**
Setup phase				
				$S_{ji}, ID_{ji}, ..., N_{ji}$
$Nonce_{ji}, S_{ji}, N_{j\,i\,new}$, $ID_{j\,i\,new}, N_{j\,i\,old}, ID_{j\,i\,old}$				
Proof generation phase				
Sub phase 1				
				1. Generates $r_1, N'_{1i}, ID'_{1i}, K_{1i}$ $PR = [r_1, TS]$
2. Verifies K_{1i} Generates P_{1i} Updates memory	$\xleftarrow{r_1, K_{1i}}$		$\xleftarrow{r_1, K_{1i}}$	
	$\xrightarrow{P_{1i}}$		$\xrightarrow{P_{1i}*}$	3. $PR+ = [G_1, N'_{1i}]$
Sub phase 2				
				4. Generates $r_2 = MAC(P_{11} \oplus P_{12} \oplus ... \oplus P_{1l} \oplus TS, \beta_1)$** Generates $N'_{2i}, ID'_{2i}, K_{2i}$
5. Verifies K_{2i} Generates P_{2i} Updates memory	$\xleftarrow{r_2, K_{2i}}$		$\xleftarrow{r_2, K_{2i}}$	
	$\xrightarrow{P_{2i}}$		$\xrightarrow{P_{2i}*}$	6. $PR+ = [G_2, N'_{2i}]$
Sub phase 3				
				7. Generates $r_3 = MAC(P_{21} \oplus P_{22} \oplus ... \oplus P_{2l'}, \beta_2)$** Generates T_{1i}
8. Verifies T_{1i} Generates F_{1i}	$\xleftarrow{r_3, T_{1i}}$		$\xleftarrow{r_3, T_{1i}}$	
	$\xrightarrow{F_{1i}}$	*Signature* $= \{F_{1i}\}$		
			Signature $\xrightarrow{\quad}$	
				9. $PR+ = [Signature]$

*If online server receives P_{1i} and P_{2i} within Δ time, then only they will be considered for verification.
**$\beta_k = S_{k1} \oplus S_{k2} \oplus ... S_{km}, (k = 1, 2)$.

Figure 6.2: Proof generation protocol

The online server then (in Step 4) generates r_2 using l number of valid responses $P_{11}, P_{12}, ..., P_{1l}$, encrypted time stamp *TS* and β_1, i.e. the XORed value of pairwise secrets of all the tags attached to the desired object. It generates new tag id ID'_{2i} and new session key $N'_{2i}, (i = 1, 2, ..., m)$ for each tag attached to the remaining object G_2. It then sends the search information r_2, K_{2i} for G_2 to reader. The reader broadcasts r_2, K_{2i}, and the desired tags respond with authentication information in Step 5. In Step 6, the online server checks the validity of the responses if it receives the responses within a time period Δ after sending the search information for G_2, and adds G_2, N'_{2i} into the proof *PR*. The online server then (in Step 7) generates r_3 using the valid responses and β_2 which is the XORed value of pairwise secrets of all the tags attached to G_2. This r_3 is generated only if the online server obtains at least one valid response. It then sends this r_3 along with $T_{1i}(= MAC(ID'_{1i}, r_3))$ to reader. The reader broadcasts these in the same step. In Step 8, the tag attached to object G_1 verifies T_{1i} and replies with valid signature F_{1i}. The reader obtains F_{1i} from tags attached to the desired object and submits *signature*=$\{F_{1i}\}$ to online server. In Step 9, the online server adds this *signature* to proof *PR*.

6.2.2.4 Proof verification

In verification phase, the online server itself can act as the verifier or a different workstation can act as the verifier. To verify the proof, the verifier needs the proof *PR* and secret information β_1, β_2. In case of different workstation as verifier, we assume that there is a secure communication through which the online server sends *PR* and β_1, β_2 to verifier. The verifier verifies the proof and either declares that the proof is valid or raises an alarm. Algorithm 23 depicts the proof verification process.

During the proof verification phase, the verifier has the proof *PR*=$[r_1, TS, \{G_1, N'_{1i}\}, \{G_2, N'_{2i}\}, signature]$ and the secret information β_1, β_2. (Here N'_{1i}, N'_{2i} are the new session keys of only the desired tags which were responded. These session keys were stored into the database in online server after verification of the authentication of the corresponding responses.) The verifier have to access β_1, β_2 to verify the proof. This is because the proof generator can be malicious and supply invalid proof. The verifier computes $q_{1k}(= MAC(N'_{1i}, r_1)), k = 1, 2, ..., l$ and then using these q_{1k}, β_1 and *TS*, it computes r_2. It then computes $q_{2k}(= MAC(N'_{2i}, r_2)), k = 1, 2, ..., l'$ and then using these q_{2k} and β_2, it computes r_3. After computing r_3, it verifies whether each $MAC(N'_{1i}, r_3)? \in signature$ or not. If all checks are successful, it declares the presence of relevant objects G_1 and G_2. Otherwise, it raises an alarm saying one or more objects are missing. The proof *PR* also can be saved and checked later for the verification of coexistence of two object G_1, G_2. The proof has a time stamp *TS* which can be decrypted to obtain the time when this proof were generated and if there is a need to generate proofs multiple times for a particular

Algorithm 23 for proof verification

Verifier has the proof *PR*=$[r_1, TS, \{G_1, N'_{1i}\}, \{G_2, N'_{2i}\}, signature]$ and β_1, β_2, Here N'_{1i}, N'_{2i} are the session keys of desired tags which were responded

$q_{1k} \leftarrow MAC(N'_{1i}, r_1), k = 1, 2, ..., l$

$r_2 \leftarrow MAC(q_{11} \oplus q_{12} \oplus .... \oplus q_{1l} \oplus TS, \beta_1)$

$q_{2k} \leftarrow MAC(N'_{2i}, r_2), k = 1, 2, ..., l'$

$r_3 \leftarrow MAC(q_{21} \oplus q_{22} \oplus .... \oplus q_{2l'}, \beta_2)$

if $MAC(N'_{1i}, r_3) \in$ *Signature* for all N'_{1i} in *PR* **then**

    Declares the coexistence of G_1, G_2

else

    Raise an alarm saying one or more objects are missing

operation, the proofs can be organized to check the sequence of a particular operation.

6.3 Analysis of the proposed scheme

We analyze the proposed proof generation protocol in order to verify the applicability in various applications. This analysis is based on the parameters such as security, computation and storage requirements.

6.3.1 Security analysis

In this section we analyze how our scheme prevents the possible attacks during proof generation process. In this section, we formally and informally analyze the security of the proposed protocol.

6.3.1.1 Informal analysis

Apart from analyzing the success probability to generate a valid proof in absence of either G_1 or G_2 or both, we informally analyze the prevention mechanisms corresponding to other kind of attacks the adversaries can apply.

Eavesdropping: The parameters which are accessible to adversary $\mathcal{A}$ are $K_{ji}, V_{ji}, r_j, P_{ji}$, $T_{1i}, r_3, F_{1i}, (i = 1, 2, ..., m$ and $j = 1, 2)$. However the secure information such as session key and id etc. are not accessible to $\mathcal{A}$. For example, the identifier ID_{ji} is unknown to $\mathcal{A}$ and she cannot be able to access it from $K_{ji}(= MAC(ID_{ji}, r_j))$ with random oracle assumption on *MAC* operation.

Replay attack: In a legitimate session of a proof generation process for the twin $\langle G_1, G_2 \rangle$, the adversary $\mathcal{A}$ intercepts $\langle P_{1i}, P_{2i} \rangle$. In another session when reader requests for the proof for

the same twin there can be three situations as follows:

Case 1: G_1 absent and G_2 present. $\mathcal{A}$ may try to provide stored P_{1i}. However, this $P_{1i}(= MAC(N'_{1i}, r_1))$ will not be verified since the r_1 included in it is not equals to the r_1 sent with the request. Also N'_{1i} which is newly generated session key in online server for this session is not equal to the session key involved in stored P_{1i}.

Case 2: G_1 present and G_2 absent. Similar to Case 1, $\mathcal{A}$ cannot be able to succeed replaying stored P_{2i} to prove that G_2 were present.

Case 3: G_1 absent and G_2 absent. According to Case 1 and 2, clearly $\mathcal{A}$ cannot prove the presence of G_1, G_2 replaying the stored P_{1i}, P_{2i}.

From the above cases we can conclude that $\mathcal{A}$ is unable to generate a valid proof in absence of one or more objects of a schedule by replaying stored information. However she may try to replay the stored information $\mu_{ji}(= r_j, K_{ji}, V_{ji})$ and wait for a reply which is same as the reply received in previous legitimate session. Thus she can trace an object. In this case, the tag will successfully verify K_{ji} using ID_{jiold} stored in its memory and then check to see whether the received r'_{ji} is equals to $Nonce_{ji}$ and will discover that the message had been replayed. If $\mathcal{A}$ try to replay the search information changing r'_{ji}, however, she cannot inject it into K_{ji} and hence K_{ji} will be invalid. Therefore the tag will reply with a random P_{ji} using which $\mathcal{A}$ is unable to trace the object. $\mathcal{A}$ may store and replay P_{ji} in response to a legitimate search request from reader. The reader/online server will reject this since r_j, r'_{ji} are not same in the new search request and $\mathcal{A}$ cannot inject r_j used in new search request into P_{ji}.

Denial of proof: Adversary $\mathcal{A}$ may put a non-legitimate tag and if the response from this tag is added into the proof *PR* the proof will be invalid. Thus in the presence of all relevant objects the generated *PR* is invalid. In our scheme the online server checks the authentication of responses from all the tags and uses only the valid response q_{ji} to generate r_{j+1} and therefore *PR* does not contain any invalid response. Hence existence of a non legitimate tag cannot disrupt the proof generation process.

Traceability: Adversary $\mathcal{A}$ may try to trace an object by observing the responses from tags. In our scheme tag uses r'_{ji}, r_j to generate $P_{ji}(= r'_{ji} \| MAC(N'_{ji}, r_j))$ in response to a search request. However the r_j and r'_{ji} are random numbers and these are not same or related to the r_j and r'_{ji} used in later or previous legitimate sessions. Therefore $\mathcal{A}$ is unable to relate the response of one legitimate session and the responses in other legitimate sessions and hence cannot trace the object.

Traceability in between two successful sessions: Adversary $\mathcal{A}$ may try to trace an object using the search information used in last successful session expecting that this information will

be validated in tag using the old information stored in the corresponding tag memory and will respond with same P_{ji}. She can use this information until the next successful session. Therefore between two successful sessions she can trace an object by replaying the search information. However as we have mentioned in the replay attack she is unable to trace the object by replaying the search information.

Man-in-the-middle attack: Adversary $\mathcal{A}$ may modify information transmitted in an insecure medium. Modifying r_j and/or $K_{ji}(= MAC(ID_{ji}, r_j))$, she cannot achieve her goal. Because she does not have ID_{ji} and hence she is unable to generate a valid K_{ji} using modified r_j. Her inability is based on random-oracle assumption of *MAC* operation. However if she modifies V_{ji}, the tag will retrieve wrong r'_{ji} and hence will reply with a wrong r'_{ji}. Therefore reader will block this response. It seems that there will be a synchronization problem between online server and tag since the online server will not update the id and session key. However in our scheme the verified id and session key are kept as old items and further search request from reader will be verified using these old information. Same reasoning holds if the adversary modifies the response P_{ji}.

Denial of service: Blocking one or more information the adversary $\mathcal{A}$ may try to de-synchronize between online server and tag. Suppose $\mathcal{A}$ blocks search information. Therefore the desired tag will not respond and also not update its session key or id and since the online server does not obtain a valid response from the desired tag it will not update the id and session key. Therefore there is no de-synchronization problem. However if she blocks the response P_{ji} then two cases may arise as follows.

a) Search request from backend server had been verified using N_{jinew}, ID_{jinew}. In this case the tag kept these information as N_{jiold}, ID_{jiold} and update N_{jinew}, ID_{jinew}. The online server did not update and keep the previous session key and id.

b) Search information had been verified using N_{jiold}, ID_{jiold}. In this case the tag did not replace N_{jiold}, ID_{jiold}. The online server also did not change the previous id and session key.

If the online server send a search information the tag will verify this using N_{jiold}, ID_{jiold} kept in its memory. Therefore blocking of any information cannot not de-synchronize.

Relay attack: Suppose one or more objects in the given schedule are not within the coexisting range. The adversary $\mathcal{A}$ will put one or more intermediate transceivers and relay the search request to desired object. Thus the proof generation process will generate a valid proof although the desired objects are not present within the coexisting range. In the proposed scheme, the online server will accept the responses P_{ji} from the tags attached to the desired object if it receives

them within a predefined time limit Δ. We assume that the relay will take more time to reach the valid response to online server and therefore relaying the search information and response from valid tag does not help $\mathcal{A}$ to generate a valid proof where one or more desired objects are not within the coexisting range.

Forward and Backward security: Adversary $\mathcal{A}$ will intercept variable secrets ID_{ji}, N_{ji} in one session and using these secrets try to intercept ID'_{ji}, N'_{ji} in the next session or vice versa. We assume that she is unable to intercept S_{ji} and hence she can not compute ID'_{ji}, N'_{ji} using ID_{ji}, N_{ji}. Therefore our scheme provides forward secrecy criteria. She also cannot compute ID_{ji}, N_{ji} using ID'_{ji}, N'_{ji} and hence our scheme provides backward secrecy.

Physical attack: Adversary $\mathcal{A}$ may capture a tag and clone this to impersonate as a legitimate tag. Our scheme does not provide prevention mechanism against this attack.

6.3.1.2 Formal security analysis

In this section, we formally analyze the proposed scheme and provide formal proofs. We first show that $\mathcal{A}$ cannot be able to generate a valid proof in absence of one or more objects in a given group, and then we show that $\mathcal{A}$ is unable to mount any attack mentioned in the threat model during or after the proof generation process.

Security of Proof Generation Scheme: A Proof Generation Scheme (PGS) is secure if, any efficient adversary, given any one interaction (not necessarily complete) and a history of earlier interactions, cannot derive (with probability greater than $0.5 + \theta$, for a non-negligible θ) any secret and consequently cannot generate any valid proof in absence of one or more relevant objects.

Theorem 6.3.1. *The success probability that $\mathcal{A}$ can generate a valid proof in absence of either G_1 or G_2 or both is bounded above by $m.2^{-d}$ (m is the number of tags attached to an object and d is the bit size of P_{1i}, P_{2i}, F_{1i}).*

Proof. $\mathcal{A}$ is asked to play a game. The tags are initialized with the necessary security parameters and $\mathcal{A}$ is permitted to interact with all the tags for arbitrarily long period of time with arbitrarily any number of interactions and sometime $\mathcal{A}$ submits the *signature* to verifier $\mathcal{V}$ through online server without interacting any tag attached to either G_1 or G_2 or both within a time period t. If $\mathcal{V}$ accepts the corresponding proof *PR* then $\mathcal{A}$ is declared as winner where $\mathcal{A}$ has not read both G_1 and G_2 simultaneously. Let the proof be *PR*=$[r_1, TS, \{G_1, N'_{1i}\}, \{G_2, N'_{2i}\}$, *signature*$]$. There can be three cases

Case 1: $\mathcal{A}$ does not obtain any P_{1i} from the tags attached to G_1 any time. Also he does not know the id of any tag attached to G_1. Therefore, $\mathcal{A}$ computes one correct P_{1i} after

guessing any one id of the tags attached to G_1 with probability at most $m.2^{-d}$(more than one correct guesses does not change the success possibility). Subsequently, $\mathcal{A}$ obtains correct P_{2i} from atleast one tag attached to G_2 and then atleast one correct F_{1i} from the tags attached to G_1. It submits these correct F_{1i} as *signature* to $\mathcal{V}$ through online server.

Case 2: $\mathcal{A}$ obtains atleast one correct response P_{1i} from the tags attached to G_1, however it does not obtain any P_{2i} anytime from the tags attached to G_2. Also he does not know the id of any tag attached to G_2. Therefore, $\mathcal{A}$ computes one correct P_{2i} after guessing any one id of the tags attached to G_2 with probability at most $m.2^{-d}$ and subsequently obtains atleast one correct F_{1i} from the tags attached to G_1 and submits this as *signature* to $\mathcal{V}$ via online server.

Case 3: $\mathcal{A}$ obtains atleast one correct P_{1i} from the tags attached to G_1 and atleast one correct P_{2i} from the tags attached to G_2. However it does not obtain any correct F_{1i} anytime from the tags attached to G_1 then $\mathcal{A}$ guesses a correct F_{1i} with probability at most $m.2^{-d}$ and submits it as *signature* to $\mathcal{V}$ through online server.

If none of the three cases happens, $\mathcal{A}$ must have simultaneously read at least one tag each attached in G_1 and G_2. □

MAC operation: This operation is used in the proposed scheme to generate authentication information. Due to resource constraints in RFID tags, an appropriate lightweight MAC operation is suggested. The adversary $\mathcal{A}$ can misuse this operation. We formally define the advantage of $\mathcal{A}$.

Definition 1: Finding q from the given numbers $p, n(= MAC(p, q))$, where q is an unknown random number, formally if $Adv_{\mathcal{A}}^{MAC}$ denotes $\mathcal{A}$'s advantage in finding q from the given p, n, we have $Adv_{\mathcal{A}}^{MAC} = Pr[q \Leftarrow_R \mathcal{A} : q$ being a random number]. [Note: $y \Leftarrow_R \mathcal{A}$ denotes y is selected randomly by $\mathcal{A}$]. $\mathcal{A}$ is allowed to be probabilistic and the probability in the advantage is computed over the random choices made by $\mathcal{A}$; we call MAC is computationally infeasible, if $Adv_{\mathcal{A}}^{MAC} \leq \epsilon$, for any sufficiently small $\epsilon > 0$.

Theorem 6.3.2. *The proposed Proof Generation Scheme (PGS) is secure under the experiment depicted in Algorithm 24 from an efficient adversary $\mathcal{A}$ to intercept any secret information from the information communicated through an insecure medium.*

Proof. We define a random oracle namely, Disclose in order to prove the theorem.

- **Disclose:** This random oracle unconditionally outputs p, q from a given input n, where $n = MAC(p, q)$.

Algorithm 24 *EXP* $1^{MAC}_{\mathcal{A},OBS}$

1: Intercepts $< r_j, K_{ji} >, j = 1, 2.i = 1, 2, ..., m$ where, $K_{ji} \leftarrow MAC(ID_{ji}, r_j)$
2: Call Disclose on input K_{ji} and obtains $r'_j, ID_{ji} \leftarrow Disclose(K_{ji})$
3: Intercepts $< P_{ji} >, j = 1, 2$ where, $P_{ji} \leftarrow MAC(N_{ji}, r_j)$
4: Call Disclose on input P_{ji} and obtains $r''_j, N_{ji} \leftarrow Disclose(P_{ji})$
5: **If** $MAC(ID_{ji}, r_j) \neq K_{ji}$ and $MAC(N_{ji}, r_j) \neq P_{ji}$ **then**
6: Return 0 (Failure)
7: **Else**
8: Successfully intercepts $ID_{ji}, N_{ji}, j = 1, 2.i = 1, 2, ..., m$

$\mathcal{A}$ does an experiment depicted in Algorithm 24 and he can successfully intercept the secret information such as session keys and ids. However, he can be successful if he is able to solve the MAC operation. The probability that he can solve this operation is ϵ, where ϵ is sufficiently small. Therefore, the proposed Proof Generation Scheme is secure from intercepting the secure information. □

Corollary 6.3.3 (Theorem 6.3.2)**.** *The Proof Generation Scheme (PGS) is secure from the attacks defined in the threat model under the experiments depicted in Algorithm 24, 25.*

Proof. $\mathcal{A}$ obtains the secret information using the experiment depicted in Algorithm 24 and he is given the information transmitted in the next session. He does the experiment depicted in Algorithm 25 using these information to mount the attacks defined in the threat model.

Algorithm 25 *EXP* $2^{attacks}_{\mathcal{A},PGS}$

1: Adversary $\mathcal{A}$ has the information $ID_{ji}, N_{ji}, r_j, j = 1, 2, i = 1, 2, ..., m$
2: Intercepts $< K^1_{ji}, r^1_j >$ in the next session, where, $K^1_{ji} \leftarrow MAC(ID^1_{ji}, r^1_j)$
3: Call Disclose on input K^1_{ji} and obtains $r^{1\prime}_j, ID^1_{ji} \leftarrow Disclose(K^1_{ji})$
4: **If** $ID^1_{ji} \neq ID_{ji}$ **then**
5: Return 0 (Failure).
6: **Else**
7: Location traceability is successful

Since A has the secret information, he has successfully mounted the Eavesdropping attack. He also can be able to mount the attacks such as Replay attack, Man-in-the-middle attack, Denial-of-service attack, Location traceability between two successful sessions. Using experiment depicted in Algorithm 25, he can be able to trace the object. Therefore, $\mathcal{A}$ is successful to mount most of the attacks defined in the threat model. However the probability of his success

Table 6.1: Security assurance

	a	b	c	d	e	f	g	h	i	j	k
Yoking proof [**?**]	Y	N	N	N	N	Y	Y	Y	N	N	N
Bolotonny and Robin [**?**]	Y	Y	N	Y	Y	Y	Y	Y	N	N	N
Yao et al. [**?**]	Y	Y	N	Y	N	Y	N	Y	N	Y	N
Duc and Kim [**?**]	Y	Y	Y	N	N	Y	Y	Y	N	N	N
Our scheme	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	P

a: Eavesdropping, b: Replay attack, c: Denial of Proof d: Traceability, e: Traceability between two successful session f: Man-in-the-middle attack, g: Denial of service, h: Information leakage, i: Relay attack j: Forward and Backward security k: Physical attack, Y: Satisfy, N: Not satisfy, P: Partially satisfy

depends on the success probability in Theorem 1. □

6.3.1.3 Comparison

Table 6.1 shows the comparison result based on the prevention mechanism of our scheme and the related schemes [**?**] [**?**] [**?**] [**?**]. The columns of the table shows various attacks indicated using the symbol $a, b, c, ...etc$. The meaning of the symbols are written under the table. Rows indicates the schemes under consideration. Therefore an entry in the table indicates that the scheme in corresponding row assures the prevention criteria against the attack in corresponding column. According to Table 6.1 we can conclude that our scheme is more secure than the existing schemes in [**?**] [**?**] [**?**] [**?**]

6.4 Resource requirement and efficiency

We analyze various resource requirements for RFID tag since it suffers from this constraint. Tag memory contains $6b$ bits information where the size of each parameter it contains is b bits. According to the security analysis in Section 6.3.1, if size of id or session key is 128 bits then the probability that the adversary can generate a valid proof in absence of either G_1 or G_2 or both is $2.938736 \times 2^{-129.555195701}$, which is negligible. Therefore, assuming key size $b = 128$ bits, the amount of tag memory requirement is only 768 bits. In terms of efficiency, a tag needs to compute maximum three lightweight *MAC* operations, one *PRNG* operation and some basic operations. In terms of communication cost, tags in G_1 requires maximum $8b$ bits whereas tags in G_2 requires only $5b$ bits.

6.5 Proof generation protocol for a group of more than two objects

The proposed protocol can be modified to adopt the proof generation process for a group of $n(n > 2)$ relevant objects $< G_1, G_2, ..., G_n >$. The modification consists of the phases in the original protocol and $n - 2$ additional phases. The Setup phase is same as in the original protocol.

Tag T_{ji} in object G_j	**Reader**	**Online server**
Setup phase		
		$S_{ji}, ID_{ji}, ..., N_{ji}$
$Nonce_{ji}, S_{ji}, N_{j\,i\,new}$, $ID_{j\,i\,new}, N_{j\,i\,old}, ID_{j\,i\,old}$		
Proof generation phase		
Sub phase 1		
		1. Generates $r_1, N'_{1i}, ID'_{1i}, K_{1i}$ $PR = [r_1]$
2. Verifies K_{1i} Generates P_{1i} Updates memory	$\xleftarrow{r_1, K_{1i}}$	$\xleftarrow{r_1, K_{1i}}$
	$\xrightarrow{P_{1i}}$	$\xrightarrow{P_{1i}}$ 3. $PR+ = [N'_{1i}, TS]$
Sub phase 2		
		4. Generates $r_2 = MAC(P_{11} \oplus P_{12} \oplus ... \oplus P_{1l_1} \oplus TS, \beta_1)^*$ Generates $N'_{2i}, ID'_{2i}, K_{2i}$
5. Verifies K_{2i} Generates P_{2i} Updates memory	$\xleftarrow{r_2, K_{2i}}$	$\xleftarrow{r_2, K_{2i}}$
	$\xrightarrow{P_{2i}}$	$\xrightarrow{P_{2i}}$ 6. $PR+ = [N'_{2i}]$
	$\vdots$	
Sub phase n		
		3n-2. Generates $r_n = MAC(P_{n-1,1} \oplus P_{n-1,2} \oplus ... \oplus P_{n-1,l_n} \oplus TS, \beta_{n-1})^*$ Generates $N'_{ni}, ID'_{ni}, K_{ni}$
3n-1. Verifies K_{ni} Generates P_{ni} Updates memory	$\xleftarrow{r_n, K_{ni}}$	$\xleftarrow{r_n, K_{ni}}$
	$\xrightarrow{P_{ni}}$	$\xrightarrow{P_{ni}}$ 3n. $PR+ = [N'_{ni}]$
Sub phase 3		
		3n+1. Generates $r_{n+1} = MAC(P_{n1} \oplus P_{n2} \oplus ... \oplus P_{nl_n}, \beta_n)^*$ Generates T_{1i}
3n+2. Verifies T_{1i} Generates F_{1i}	$\xleftarrow{r_{n+1}, T_{1i}}$	$\xleftarrow{r_{n+1}, T_{1i}}$
	$\xrightarrow{F_{1i}}$ *Signature* $= \{F_{1i}\}$	
		$\xrightarrow{\text{Signature}}$
		3n+3. $PR+ = [Signature]$

*$\beta_k = S_{k1} \oplus S_{k2} \oplus ... S_{km}, (k = 1, 2, ..., n)$.
**If time between step 5 and 13 is less than $\triangle_1$, the *Signature* will be added to *PR*. Else proof will not be accepted.

Figure 6.3: Proof generation protocol for a group of more than two objects

In the proof generation phase, the first two sub phases in the modified version are similar to the corresponding sub phases in the original version. In the third sub phase, the online server generates r_3 using the valid responses P_{2i} (from the tags attached to G_2) and β_2 (Xored value of the secret keys of the tags attached to G_2). This r_3 has been used to generate the search information for the object G_3. It also generates the new session key and id N'_{3i}, ID'_{3i}. The online server follows the same process when it had generated the same parameters for G_2. The tags attached to G_3 verify the search information (similar to tags attached to G_2) and respond with authentication information P_{3i}. The online server verifies the responses if it receives them within the time period Δ from when it sends the search information for object G_3, and adds G_3, N'_{3i} to proof *PR* if it obtains at least one valid response. The next phase ie, sub phase 4 is executed in similar fashion for object G_4. Thus the objects $G_5, G_6, ..., G_n$ are searched in a chain. After finding the object G_n (execution of phase n), the modified protocol generates r_{n+1} using the responses from the tags attached to G_n and β_n (Xored value of the secret keys of the tags attached to G_n). This r_{n+1} is used to generate the signature request T_{1i} in the last phase of the modified scheme. The last phase of the modified scheme also follows the same process in the original scheme and it adds the *signature* into the proof *PR* if it finds the *signature* within certain period of time $\triangle_1$. Thus, the modified scheme generates the the proof *PR* which can be verified to check the coexistence of $< G_1, G_2, ..., G_n >$. The online server cannot generate the proof *PR* in absence of any object in $< G_1, G_2, ..., G_n >$.

The verification phase of the modified scheme also similar to the verification process in the original scheme. The proof in the modified scheme is *PR*=$[r_1, \{G_1, N'_{1i}\}, TS, \{G_2, N'_{2i}\}, ..., \{G_n, N'_{ni}\}, signature]$. The verifier generates $< r_1, q_{1k} >, < r_2, q_{2k} >, ..., < r_n, q_{nk} >, r_{n+1}$ in a chain and finally verifies whether each $MAC(N'_{1i}, r_{n+1})? \in$ *signature* or not. If all checks are successful, it declares the presence of relevant objects $G_1, G_2, ..., G_n$. Otherwise, it raises an alarm saying one or more objects are missing.

6.6 Conclusion

In this paper we propose a proof generation protocol for the objects attached with multiple number of tags and we analyze our scheme based on various security and resource requirements. We find that our scheme is secure against most of the possible threats and the security is based on random-oracle assumption on a lightweight *MAC* operation except Information leakage threat where this attack is based on the hardness assumption of *XOR* operation. However our scheme is not secure against Physical attack where the adversary may clone and impersonate a legitimate tag. A suitable *Physically Unclonable Function* can solve this problem. We have designed our

scheme in such a way that the overhead due to undesired response is negligible. The proposed scheme is applicable for a schedule of two objects. This scheme can be extended to deal with the schedules having more than two objects.

Chapter 7

Conclusion

htihktnhkr tgkhjotjyhjty tyopjpop

Bibliography