Bit Shuffling Query Tree Algorithm for Secure Communication in RFID Networks

Nivedita Das
Dept. of Information Technology
Calcutta UniversityKolkata, India

Indrajit Bhattacharya
Dept. of Computer Application
Kalyani Govt. Engineering College
Kalyani, India

ABSTRACT

RFID uses radio frequency waves to track and identify objects. RFID system consists of tags and readers. Passive RFID tags are more popular now days due to its simpler circuitry, lower weight and lower cost. Generally passive tags are attached with an object for unique identification. A tag contains information about the particular product, to which it is attached. RFID reader is used to interrogate the tag to get the identity of the product attached with. RFID has numerous applications such as inventory management, asset tracking, library management etc. One of the major problems with RFID system is that whenever a reader interrogates a tag, all the tags residing in the read zone of the reader responds to the reader, which results in a collision at the reader's side. Thus makes it impossible to read the tags in time. To minimize the problem of collision at the reader side several anti collision algorithms has been proposed, but none of the algorithm concerns about the security aspect. Security is another major problem of RFID system. RFID tag discloses its identity to any reader which interrogates the tag, so any illegitimate reader can obtain the information contained in the tag. Now a day's situation is much worse because not only unique identification number but also user information such as name, address, phone number and other relevant information's of consumer are stored in the tag. There are different manufacturers of tags who provide tags with 1kb memory, which is large enough to store such kind of consumer information without pointing to backend database. In such cases security concern is much more as eavesdropping may lead to disclose some personal data, which is a big threat for consumer. Implementing cryptographic algorithm is not a feasible solution for the RFID system, as this may result in complex circuitry, and which in turn may raise size and cost of the passive tags. Here in this paper we have combined both the problem of collision and security together to find a suitable solution which solves both the problems. Taking into consideration of simpler circuitry and lower price of passive tags, here we propose a solution which modifies existing anti collision protocol to make it secure so that any illegitimate reader cannot read the information contained in the tag thus protecting tags from malicious reading.

General Terms

RFID secure singulation protocol.

Keywords

Secure anti collision protocol, Secure Tree based algorithm.

1. INTRODUCTION

The abbreviation RFID stands for "Radio Frequency Identification". It is a contactless automatic, without line-of-sight (LOS), low-power and low-cost wireless communication technology that provides automatic identification and data collection. This technology uses radio waves to transfer data from an electronic tag, called RFID tag, attached to an object, through a reader for the purpose of identifying and tracking the object. RFID and barcodes has several common properties with respect to automatic identification perspective. It has several advantages compared to any other identification technique, like it provides greater speeds, data encryption, covers greater distances; it is more immune to surrounding environment, it also reduces human involvement in the identifying process. RFID is regarded as a substitute technology for the barcode which is currently used in distribution and circulation fields and financial services. RFID technology depends on communication between reader and tag. The read range of the reader depends on the operational frequency of the reader [1]. RFID becoming the most popular identification technology day by day because of its low price and simple circuit design. RFID has a numerous number of applications, such as tracking and managing inventory, people, assets, domestic animals, library books etc. Health industry also used RFID technology to identify medicines, which are running out of date and should be replaced with a new one. Also in the recent past (2010) social media, announced to tie physical world to the virtual world by using this technology.

RFID technology comprises of three components namely tags, readers and data processing sub system.

[1] Tags are generally of two types, active and passive. Tags are also known as transponders. Usually tags are attached with an object for unique identification. Passive tags do not contain any battery source. Instead they derive power from reader when interrogated. [2]

[2] Readers are also known as transceivers. They are used to interrogate tags using radio frequency waves. The main function of reader is to wake up the passive tags and retrieve information stored in the tags memory.

[3] A Data processing subsystem can be application or database, depending on the application. This can be attached with the reader through wireless or wired media. [2]

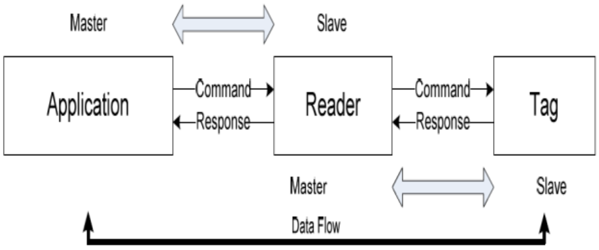


Figure 1: RFID system components

2. RELATED WORKS

RFID tag anti collision protocols can be grouped into two categories, deterministic and probabilistic.

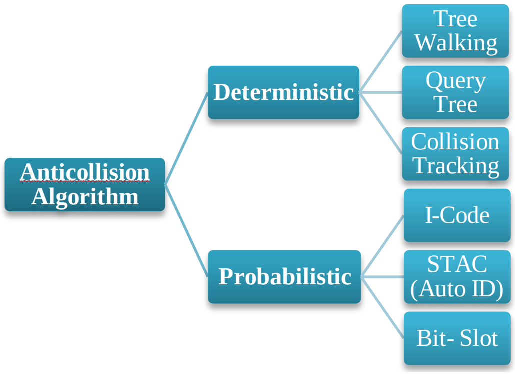


Figure 2: Taxonomy of the existing tag anti-collision protocols

Deterministic methods take more time to identify tags as compared to probabilistic methods but ensure that after certain iterations all the tags present in the read zone of a reader will be successfully read. Probabilistic methods do not ensure that all the tags will be read or not after certain amount of time. Here in this paper we have taken only deterministic approaches into consideration. [2]

3. DETERMINISTIC APPROACHES FOR PASSIVE MEMORY LESS RFID TAGS

Tree based deterministic algorithms generally traverse from the top to the bottom of the tree. The root is always same for all the tags, where leaf represents the tags.

3.1 Binary Tree Walking Singulation Protocol (BTWA)

Singulation protocol enables reader to talk to each tag singly. Here reader chooses a 0 or 1 for the initiative. If the reader makes a choice, the identification process should keep the way of choice order when the tree splits at a node. Then the binary tree walking algorithm (BTWA) is operated as follows:
Step 1: The reader transmits k-length prefix.
Step 2: Tags send (k+1)th bit if the first k bits of tag IDs are the same as the prefix.
Step 3: If the received bits collide, the extended prefix attached ‘0’ or ‘1’ to the prefix is retransmitted by the reader. If they do not collide, the received bit is attached to the prefix for the next prefix. If there is no response, the branch is ignored. Also, a collision occurs at the last bit of the tag IDs, the reader assumes there are two tags because of the uniqueness of the tag IDs.
Step 4: The reader repeats the procedure until all branches are searched [2].
The functionality of the protocol is shown in figure 3.

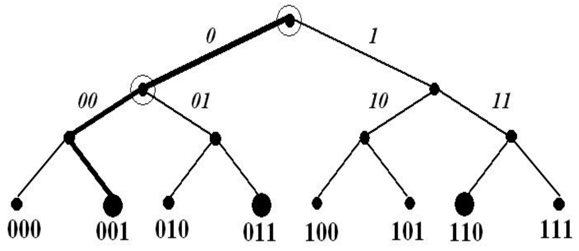


Figure 3: Binary tree walking protocol

Table 1. Sequences of Binary Tree Walking algorithm

Reader Query	Tag 1 001	Tag 2 011	Tag 3 110	Result
NULL	R	R	R	collision
0	0	1	-	collision
00	1	-	-	Tag 1 is read
01	-	1	-	Tag 2 is read
1	-	-	1	Tag 3 sends response
11	-	-	0	Tag3 is read

This tree, shown in figure 3 is of depth 3, has 2^3 = 8 tag serial numbers represented at its leaves. The prefixes associated with sub trees are denoted in italics. In this example, we consider three tags as being present, the ‘001’, ‘011’, and ‘110’ tag. These are indicated by large black circles at their respective leaves. The tree walking algorithm here first singulates ‘001’ tag. It does this by following the path denoted by the darkened edges. At two nodes, namely the root of the tree and the root for all tags with a ‘0’ prefix, there are collisions in the bits broadcast by tags, because there are tags present in both the left and right sub trees. We denote these collision points with hollow circles. Singulation of the ‘011’ and ‘110’ tags would follow by recursion on the collision points [3].

3.2 Query Tree Algorithm (QTA)

The query tree algorithm (QTA) is based on BTWA and it works as follows:
Step 1: The reader transmits k-length prefix.
Step 2: Tags send from (k+1)th bit to the end bit of tag IDs if the first k bits of tag Ids are the same as the prefix.
Step 3: If there is a collision, the extended prefix attached ‘0’ or ‘1’ to the prefix is retransmitted. Furthermore, if there is no collision, the reader identifies a tag corresponding to the detected ID, which is the connection of the prefix and the response [2].

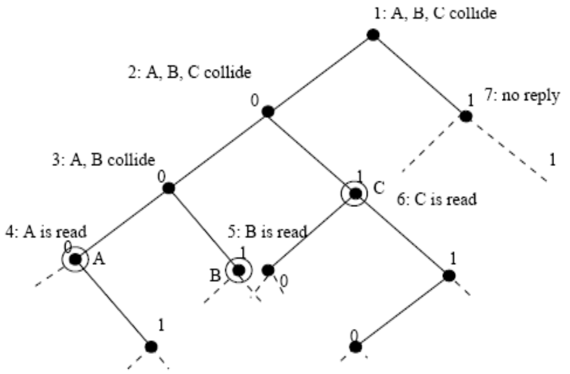
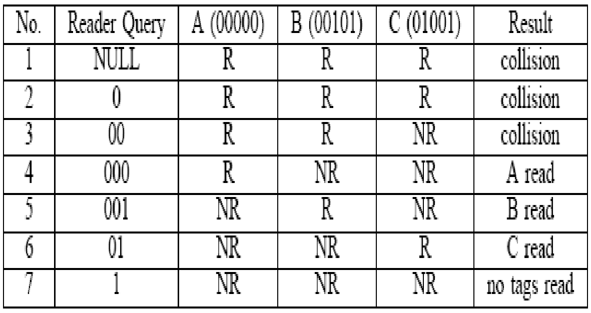


Figure 4: Example of Query tree algorithm

Table 2. Sequences of Query Tree algorithm



No.	Reader Query	A (00000)	B (00101)	C (01001)	Result
1	NULL	R	R	R	collision
2	0	R	R	R	collision
3	00	R	R	NR	collision
4	000	R	NR	NR	A read
5	001	NR	R	NR	B read
6	01	NR	NR	R	C read
7	1	NR	NR	NR	no tags read

Here in table 2, the sequences of Query Tree algorithm are drawn. Here a reader surrounded by 3 tags is considered. Initially reader sends a broadcast frame to initiate communication with the tags. Then reader sends a “0” and all the three tags send their ID [from (k+1) th position to the end of their ID] to the reader, thus a collision occurs. Then reader sends a “00”. As there are two tags with “00” prefix, the tags send their responses, which results in a collision. Then reader sends a”000”. There is only one tag with this prefix so only that tag replies and gets identified by the reader. The reader identifies that tag corresponding to the detected ID, which is the connection of the prefix “000” and the response “00”. The same process continues until all the tags are identified. [4]

3.3 Collision Tracking Tree Algorithm (CTTA)

The collision tracking tree algorithm (CTTA) is based on QTA except that this scheme uses collision tracking. The algorithm works as follows:

Step 1: The reader transmits k-length prefix.

Step 2: Tag: the tags send their IDs from (k+1)th bit to the end bit if the prefix is the same as the first k bits of tag IDs. However, the tags stop sending their IDs when an ACK signal is received.

Reader: the reader checks whether a collision occurs or not in each bit on the received sequences, and transmits an ACK signal to stop being sent the tag IDs by the tags if there is a collision.

Step 3: If there is a collision at nth bit in the received sequences, the two new prefixes, ‘the former prefix k bits + the received n-1 bits + 0 or 1’, are retransmitted sequentially to the tags in the field of the reader. Furthermore, if there is no collision, the reader identifies a tag corresponding to the detected ID, which is the connection of the prefix and the response [2].

4. DIFFERENT APPROACHES TO SECURE CONSUMER PRIVACY

In this section we will discuss some of the previously suggested approaches that are suggested for protecting consumer privacy.

4.1 Kill Tag Approach

In a RFID enabled super market, every item is attached with a tag for theft control. When a consumer buy a product, if the tag remains at the consumer’s product then anyone anywhere can intercept tag’s identity to track consumer’s location, which is not desirable. The kill tag approach is used to overcome this problem. In this approach after buying products when consumer goes to the counter for payment, the clerk at the counter kills the tag. A killed tag is totally dead and can never be reactivated. The standard mode of operation proposed by the Auto ID Centre is indeed for tags to be killed upon purchase of the tagged product. With their proposed tag design, a tag can be killed by sending it a special “kill” command (including a short 8-bit “password”). [5, 6]

There are some situations when tag killing approach does not work. Some consumer wants RFID tags to remain operative in their possession also. Here are the examples of some applications where a tag needs to be active after purchase of the product.

- Stores may wish products to have tags scan able if the products are returned as defective.
- Products may need to be scanned so they may be categorized for recycling purposes.
- Stores may issue receipts with embedded RFID tags, so they can confirm purchase details when a product is returned.

Thus, while the “kill-tag on purchase” approach may handle many or even most instances of potential concern for privacy, it is unlikely to be a fully satisfactory solution. [5, 7]

4.2 The Faraday Cage Approach

An RFID tag may be shielded from being read using what is known as Faraday’s cage. Faraday’s cage is made of metal mesh or foil impenetrable by radio signals (Of certain frequencies). Petty thieves are already known to use foil-lined bags in retail shops to circumvent shoplifting detection mechanisms. RFID has numerous numbers of applications. In many cases it is impossible to place the product in any containers, such as clothing, wrist watches etc. [8, 11] Faraday cages thus represent at best a very partial solution to consumer privacy. [3, 9, 10, 15]

4.3 The Active Jamming Approach

Active jamming is another form of shielding tags. The consumer could carry a device that constantly broadcasts radio frequency signals to jam the network, so that illegitimate reader can not read the consumer’s tag. This type of activity is illegal, especially if the broadcast signals are too high in power. This approach disrupts another RFID systems surrounded by the device. So we can conclude that this approach is effective in only some limited cases. [3]

4.4 The Smart RFID Tag Approach

Three instances of the “smart RFID-tag” approach that have been proposed are the hash-lock method, the re-encryption method (in several forms), and silent tree-walking.

4.4.1 The “Hash Lock” Approach

In this approach a tag may be in two states locked or unlocked. When a tag is locked it refuses to reveal its ID. In its simplest scenario when a tag is locked it is given a value y (Meta ID), and then it will be only unlocked by presentation of a key /pin value x such that y=h(x) for a standard one way function h. In this approach if a reader wants to query a tag then it must know the Meta ID of the tag, to find out the key/ pin. This is an effective way of managing tags but some times it become difficult for the consumers to manage lock unlock patterns and associated pins. [3, 12, 13]

4.4.2 The re encryption approach

In this approach the RFID-tags embedded in banknotes, with a scheme where banknote tag serial numbers are encrypted with a law-enforcement public key. The resulting cipher texts undergo periodic re encryption to reduce the link ability of different appearances of a given tag. Because of the severely restricted computing resources of RFID tags, they propose that re-encryption be performed by external computing agents, e.g., publicly provided privacy enhancing stations in stores. The correct behaviour of such re-encryption agents may be verified when banknotes are handled in stores and banks. The main drawback to this approach is its resource-intensive nature. While RFID tags in their scheme do not perform cryptographic operation and would not be unrealistically costly, the required infrastructure of re-encryption agents and optical verifiers would probably be burdensome. [3, 14]

4.4.3 Silent Tree Walking

Signals generated by readers are more powerful as compared to tags. Generally reply generated by tags is powered up by reader's signals. So tag's generated signal is less powerful than reader, so travels much lesser distance than readers. So passive eavesdroppers can easily eavesdrop to readers signal from hundreds of meters away. In case of tree walking algorithm reader sends each and every bit to the tag. So in this can if an intruder eavesdrop to readers channel only then he will be able to get the tag Ids from hundred of meters away. There is no need to eavesdrop to tags response. To overcome this reader's signal is being encrypted so that any passive eavesdropper can not infer the Ids being read. [3, 12]

5. PROPOSED SOLUTION TO THE RFID SYSTEM

5.1 Background

Several anti collision protocol have been proposed to overcome collision by tag's responses, but none of the anti collision algorithm implements built in security mechanisms. As mentioned earlier, security threat is one of the major threats of RFID system. RFID tag generally stores a unique identification number. The unique identification number points to some other related information in the backend database. These days tag can also store other information without pointing to a backend database as now different vendors provides tags with larger memory to store all the data in the tag's memory. In case of library management the information including the book's title, author and the name and library card number of the person borrowing the book can be stored into the tag. To support offline processing these information are added to the tags memory. Earlier when tags used to contain only identification number, if backend database goes down, the total system has to pause. To overcome this difficulty now tags can provide other user information and history of usage of the tag without pointing to database. Uses of this type of tag include library cards or loyalty cards. Loyalty cards stores information like consumer name, address, phone number, fascination of the consumer about some particular products, purchasing habit etc. So it can be concluded that the tags with larger memory increases the security risk of the consumer, as eavesdropping to the channel while transmitting tag information to reader may disclose private data, such as name, address, phone number etc. Nobody wants to disclose their identity to any unknown person, as the gathered information of a particular person may be used in some crime. We have to find some way to protect passive tags from malicious reading without increasing the circuit complexity and price of tags.

Keeping those constraints in mind here in this paper we propose Bit Shuffling Query Tree algorithm which improves existing Query Tree anti collision algorithm by implementing some basic security mechanisms to it so that any intruder cannot decrypt the information. Here we have selected Query Tree algorithm because a number of anti collision algorithms have been proposed, but the anti collision algorithms do not offer any security mechanism built into it. So we would like to implement some built in security parameter into existing Query Tree algorithm. Here we apply bit shuffling algorithm to shuffle the Tag information, before sending the information to the reader/ interrogator. So in this case the actual information of the tag is not sent rather the shuffled information is sent to the reader, which can be decoded by a legitimate reader at the reader's end. Any one in the read zone of the tag can intercept the tag's information but will not be able to get the actual data, thus ensuring security to the consumer. Here we are applying very simple bit shuffling algorithm with very few computations so it will not increase the tag's size and cost.

5.2 Proposed Bit Shuffling Query Tree (BSQT) Algorithm

Our proposed algorithm is an improvement over Query Tree Algorithm and it works as follows:

Step 1: Each tag shuffles its TagID using our bit shuffling algorithm.

Step 2: Reader transmits a k bit prefix.

Step 3: Tags send from (k+1)th bit to the end bit of their shuffled TagIDs if the first k bits of shuffled tag IDs are the same as the prefix. Tag does not send their actual ID in fact they send the ID shuffled by applying bit shuffling algorithm.

Step 4: If there is a collision, the extended prefix attached '0' or '1' to the prefix is retransmitted. Furthermore, if there is no collision, the reader identifies a tag corresponding to the detected ID (Shuffled ID), which is the connection of the prefix and the response.

Step 5: The targeted reader applies decryption algorithm to get the actual identity of the tag.

The functionality of the algorithm is figure 5.

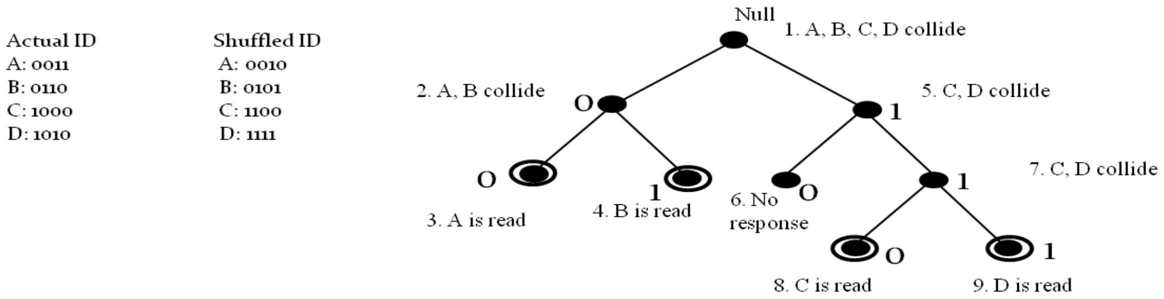


Figure 5: Example of Bit Shuffling Query tree algorithm

Table 3. Sequences of Bit Shuffling Query Tree searching scheme.

Serial Number	Reader Query	A:0010	B:0101	C:1100	D:1111	Result
1.	Null	R	R	R	R	Collision
2.	0	R	R	-	-	A, B collide
3.	00	R	-	-	-	A identified
4.	01	-	R	-	-	B identified
5.	1	-	-	R	R	C, D Collide
6.	10	-	-	-	-	No response
7.	11	-	-	R	R	C, D collide
8.	110	-	-	R	-	C identified
9.	111	-	-	-	R	D identified

In the above figure, R represents response from the tag. The actual Tag ID is shuffled by applying bit shuffling algorithm, which is described in the next section.

5.3 Bit Shuffling Algorithm Implemented For Encryption of Tag ID

In our scheme we have implemented a simplified bit shuffling algorithm to incorporate the security of the RFID system and it works as follows

Step 1: Let E1 is the original Tag ID. A '0' is appended to the MSB of the tag's ID (i. e E1).

Step 2: length of the tag ID after appending '0' is calculated and stored in a variable n.

Step 3: Apply EX-OR operation to the nth (MSB) and (n-1)th bit of E1 and store the result at the (n-1)th position of another expression E2.

Step 4: Decrement n by 1.

Step 5: Repeat steps 3 and 4 until n becomes 2 or in other words (n-1) bit reaches the LSB of the tag's ID.

The process has been described in figure 6

Truth Table for EX-OR Gate

Expression Z=X'Y+XY'

E1:0110

00110 ⇒ $0\oplus0\oplus1\oplus1\oplus0$

⇓

E2: 0101

Figure 6: The Bit Shuffling Process

Example

E1: 0110
Step 1: 00110, zero is appended
Step 2: n=5
Step 3: E2=0
Step 4: n=4
Step 5: E2=01
Step 6: n=3
Step 7: E2=010
Step 8: n=2
Step 9: E2=0101
Step 10: n=1 n reaches 1 so here the loop stops and the string contained in E2 is the encrypted tag ID.

5.4 Decryption of Tag ID by the Targeted Reader

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

The encrypted tag id E2 is received at the receiver side. After receiving the encrypted string the reader does the following steps to determine the actual tag ID.

Step 1: The length of the received Tag ID is calculated and stored in a variable n.

Step 2: The MSB of the E2 is copied into the nth position of another expression E3.

Step 3: Now the nth bit of E3 is taken and performed Ex-OR operation with (n-1) bit of E2 and stored in the (n-1)th location of E3.

Step 4: Decrement n by 1.

Step 5: Repeat steps 3 and 4 until n becomes 2 or in other words (n-1) become LSB of E2.

E3 is the decrypted Tag ID. Hence the string contained in E3 is an exact replica of E1.The method is described in Figure 7.

.

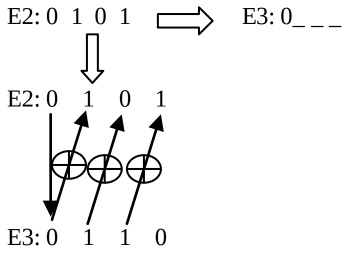


Figure 7: The Method for decryption

5.5 Functionality of the System

This approach minimizes the threat to RFID system by implementing bit shuffling algorithm to existing Query Tree algorithm. At step 1 each tag shuffles its containing information. When a reader sends a prefix for singulation of tags, the tag compares weather the sent prefix is same as their shuffled tag ID. If the prefix is same as the shuffled ID, the tags send their remaining portion of tag ID to the reader. If a collision occurs, reader proceeds in depth first search order. If collision does not occur then reader identifies a tag with its shuffled Tag ID, which is a connection of prefix and response. After identifying shuffled tag ID, reader applies decryption algorithm to get the original tag ID. The tag's containing information can only be decrypted by legitimate readers. Hence it protects consumer information from being read by illegitimate readers.

6. IMPLEMENTATION OF THE PROPOSED ALGORITHM

We have carried out the implementation of the proposed Query Tree algorithm in core java platform. We have considered a scenario where a fixed reader is located at about central point in a 400x400m^2 simulation area and 50 tags are surrounded the reader, each containing a unique tag ID of 8-bit length. A tag can be successfully read by a reader if it is situated in the read zone of a reader. In our experiment we have set up the communication range of a reader as 50m. Hence our proposed system first verifies which tags are situated in the read range of the reader and then applies our proposed algorithm for singulation of tags. Figure 8 shows the snapshot of initial deployment of tags and reader in the 400x400m^2 simulation area. Figure 9 shows the snapshot of how the tags get identified by the reader after the proposed algorithm has been implemented.

● Reader
○ Unidentified Tags
● Identified Tags

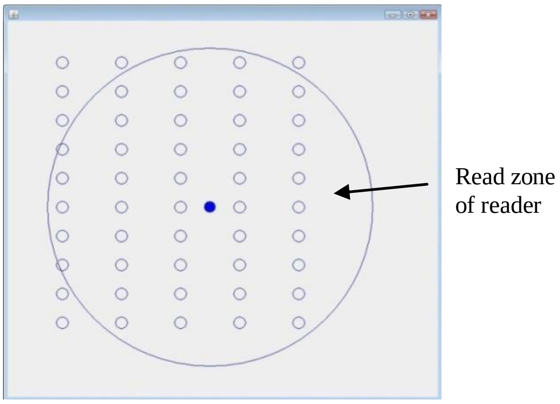


Figure 8: Snapshot of the initial deployment of tags and reader

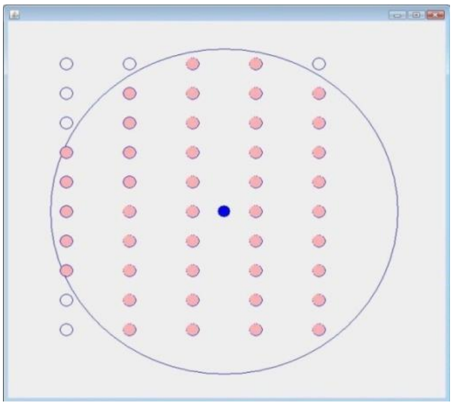


Figure 9: Snapshot of the tags identified by the reader

7. RESULTS

We have analysed the performance of the proposed system in terms of the time required to identify the tags by the reader. It is desirable that the time taken to identify the tags by the reader should be less even if the encryption and the decryption logics get implemented. Figure 10 represents a comparative study between Query Tree algorithm and our proposed Bit Shuffling Query Tree algorithm. First we analysed time taken by Query Tree approach to find the tags. Then we implemented our Bit Shuffling algorithm in Query Tree algorithm and analysed with respect to time. We can see from figure 10 that our proposed Bit Shuffling QT (BSQT) algorithm takes a little more time to identify tags compared to Query Tree Algorithm. So we can conclude that time required for encryption and decryption is very minimum and well-suited for passive RFID tags. It can be observed that if the number of tags with in the range of a reader is increased it will require larger amount of time to identify them.

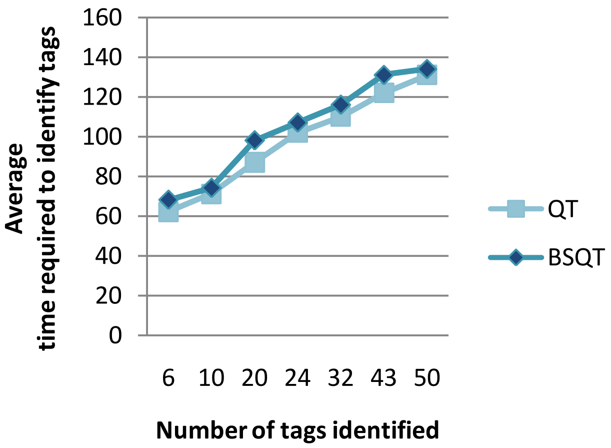


Figure 10: Graph showing comparison of average time required (in milliseconds) to identify the tags using QT and Bit Shuffling QT (BSQT)

We have also analysed string or number of messages required to identify tags. Figure 11 shows the obtained result. From the graph, it can be observed that when number of tags increases, reader needs greater number of strings to identify the tags.

8. CONCLUSION

In this work we proposed an algorithm to secure existing Query Tree protocol, to protect data contained in the tag. The RFID tags, which contains information about user, for example loyalty cards, library cards, are very vulnerable to threats. As eavesdropping to this will disclose personal identity of individuals, which is a big threat to RFID system. Here we have proposed an algorithm which enforces security parameters to existing anti collision algorithm named Query Tree algorithm (QTA). By modifying anti collision algorithm we can have some built in mechanishm to protect data contained in the tag. We are using our proposed bit shuffling algorithm to shuffle the data contained in the tag so any eavesdropper can get the encrypted data hence they are not able to get the actual identity of the consumer. Especially where the tags contains consumer's information, this approach will be very effective. As we have used very simple computations to shuffle the tag information, this approach is feasible to passive low cost, lighter weight tags. We have analysed the time needed to identify tags using Query Tree and Bit Shuffling Query Tree algoritm.The comparative study of shows that a very little more time will be needed to shuffle the tag information. The experimental results assured us that our proposed algorithm can identify the all the tags in a very minimum time satisfying the time constraints. That means time time for encryption and decrption of data is very minimum. It can also be observed from the obtained graph that if number of tags increases time taken by reader to identify all the tags increases and vice versa. We have also analysed number of strings needed to identify tags within the read zone of a reader. After having the result of both the experiments we can conclude that this approach will be very effective where tag contaions consumer related information by protecting tags from malicious reading. Only targeted reader will be able to get the tag's information.

9. REFERENCES

[1] B. Glover, & H. Bhatt, O'Reilly Media, Inc, Sebastopol, (2006), ISBN 0-596-00944-5.

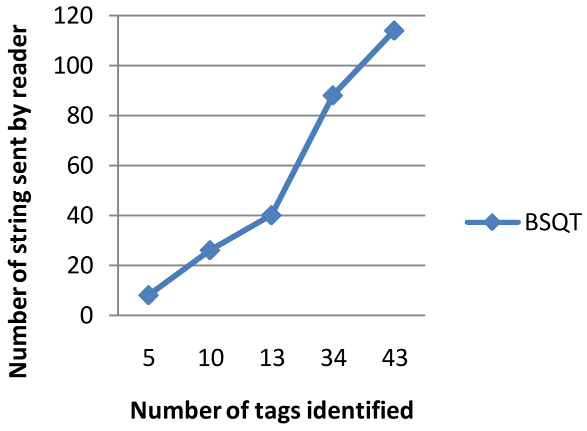


Figure 11: Graph showing number of strings required to identify the tags.

[2] Okkyeong Bang (ICU), Ji Hwan Choi (ICU), Dongwook Lee (ICU), Hyuckjae Lee (ICU), Efficient Novel Anti Collision Protocols for Passive RFID Tags, Auto-ID Labs White Paper WP-HARDWARE-050, March 2009.

[3] Ari Juels,Ronald L Rivest, Michael Szydlo, The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy.

[4] Vinod Namboodiri, Lixin Gao, Energy-Aware Tag Anti-Collision Protocols for RFID Systems.

[5] S. E. Sarma, S. A. Weis, D.W. Engels, Radio-frequency identification systems. In Burton S. Kaliski Jr., C¸ etin Kaya Ko¸c, and Christof Paar, editors, CHES '02, pages 454–469. Springer-Verlag, 2002. LNCS no. 2523.

[6] S. E. Sarma, S. A. Weis, and D.W. Engels. RFID systems, security and privacy implications. Technical Report MIT-AUTOID-WH-014, AutoID Center, MIT, 2002.

[7] S. E. Sarma, S. A. Weis, and D.W. Engels. Radio-frequency-identification security risks and challenges. CryptoBytes, 6(1), 2003.

[8] Benetton undecided on use of 'smart tags'. Associated Press, 8 April 2003.

[9] R. Shim. Benetton to track clothing with ID chips. CNET, 11 March 2003.

[10] S. Garfinkel. An RFID Bill of Rights. Technology Review, page 35, October 2002.

[11] D. McCullagh. RFID tags: Big Brother in small packages. CNet, 13 January 2003.

[12] S. A. Weis, S. Sarma, R. Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In First International Conference on Security in Pervasive Computing, 2003.

[13] S.A. Weis. Radio-frequency identification security and privacy. Master's thesis, M.I.T. June 2003.

[14] A. Juels and R. Pappu. Squealing Euros: Privacy protection in RFID-enabled banknotes. In R. Wright, editor, Financial Cryptography '03. Springer-Verlag, 2003.

[15] mCloak: Personal / corporate management of wireless devices and technology, 2003. Product description at www.mobilecloak.com.