JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

BILLING SYSTEM BASED ON RFID USING BLOWFISH ALGORITHM

Vikas Singhal[1] , Shivani Dubey[2] , Ashish Kumar Singh[3] , Abhishek Awasthi[4] , Rishu Singh[5]

Professor[1] , Associate Professor[2] , Student[3] , Student [4] , Student [5]

[1] Greater Noida Institute Of Technology, Greater Noida, 201310, India
[2] Greater Noida Institute Of Technology, Greater Noida, 201310, India
[3] Greater Noida Institute Of Technology, Greater Noida, 201310, India
[4] Greater Noida Institute Of Technology, Greater Noida, 201310, India
[5] Greater Noida Institute Of Technology, Greater Noida, 201310, India

ABSTRACT

In this study, the Blowfish approach is used to create a billing system for radio frequency identification (RFID). It will employ RFID in billing systems that are designed to quickly identify billing in order to save time and develop a new payment revolution processing. So it is anticipated that using such a system will lessen the congestion that frequently occurs in the billing queue. This is typically caused by how long barcode payment processes take. Data security from the user's device, such as electronic card RFID tags, is also necessary in addition to transaction time speed. The victim of billing card data theft will suffer. Since the blowfish approach is being used in this instance to protect or conceal user data, safeguarding user data can prevent tampering that would otherwise result in the creation of duplicate RFID tags. For compressing image and to choose best blowfish algorithm feature so that to reduce image without data lost blowfish method is used.

IndexTerms

Cryptography; Security; Zigbee; Blowfish Algorithm; RFID Module; billing.

1. INTRODUCTION

Radio Frequency Identification, or RFID as it is more often known, is a system that uses radio frequencies to automatically identify objects without needing to come into touch with any people. The auto-ID system, commonly known as RFID, typically employs radio waves. Such systems typically include a tag and reader for use in the RFID system processing. The RFID tag must be close to the RFID reader in order for it to function, and the reader will recognise data depending on the data or information stored on the RFID tag.

RFID tags typically take the form of cards, stickers, and a variety of other objects. Each tag has a different data ID because each tag has a unique data ID. RFID labels typically take the form of cards, stickers, and a variety of other objects. The data ID is distinct for each tag since each tag has an uniquely unique data ID.The Blowfish symmetric-key block cypher was developed by B Schneier in 1993. It is renowned for its quick encryption and decryption times as well as its key size versatility. Although there is only one version of the Blowfish algorithm, it can be utilized in a variety of ways to obtain various security features.

2. BLOWFISH ALGORITHM

A practical symmetric-key block cypher to encrypt and decrypt data is called blowfish. Although it wasn't made with data compression in mind, it can also be used for that.

Protecting user data can prevent manipulation to produce duplicate RFID tags in this situation since the blowfish strategy is used to protect or hide user data.[1]
The blowfish algorithm consist of two important parts:

I. Data encryption

Data is encrypted using a 16-round Feistel network with key- and data-dependent substitutions and permutations in each round. To encrypt data in Blowfish, the replacement strategy is combined with big, keys-based on S-boxes.
All encryption processes include adds on 32-bit words and XORs, a sort of logic gate. This qualifies it for a variety of security needs and applications.
The module operates by splitting the plaintext into 64-bit blocks and doing many rounds on each block.[2] Four operations—Substitution, Permutation, Key Mixing, and Key Addition—make up each round.

II. Data Decryption

When ciphertext has to be converted back to plaintext, the decryption module is employed. The encryption algorithm is used in reverse order by the module to operate. The four procedures that make up each round of the decryption process are the same as those that make up each round of encryption, but the subkeys are utilised in the opposite order. The encrypted image is divided into blocks of the block length of the Blowfish method from top to bottom. While the picture can be decoded using the same encryption key as the first block, the subkeys are applied in the opposite direction.[3] From top to bottom, additional image blocks are added during the decryption process.

III. Key expansion and subkeys

448-bit keys are stretched to a maximum capacity of several subkey arrays totaling 4,168 bytes. throughout the key expansion procedure.[5] The Blowfish algorithm, which makes extensive use of subkeys, depends on them. Before any encryption or decryption can occur, these subkeys are pre-calculated(fig 1, fig2).

-Subkey count is 18 keys (P array).
-16 rounds total
-There are 4 substitution boxes.

Following algorithm is used to convert normal text to cipher text:-

1.Divide plain text into two blocks L and R, each of
which should be 32 bits in size.

2.For i=1 -16
Li=Li ⊕ Pi
Ri= F(Li) ⊕ Ri
Swap L1 R

3.Unwind final switch
R= R ⊕ P17
L = L ⊕ P18

4. L and R are concatenated to create 64 bit cypher text.

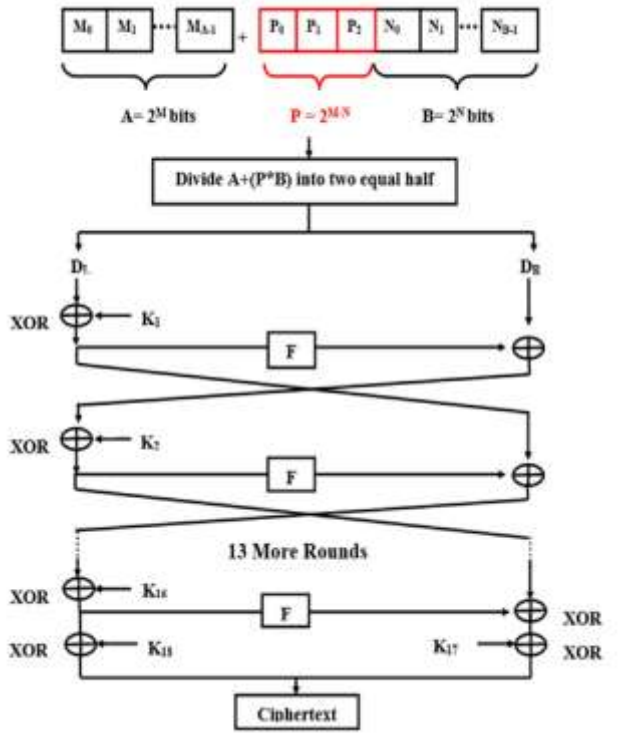

Fig1. Conversion from normal text to cipher text

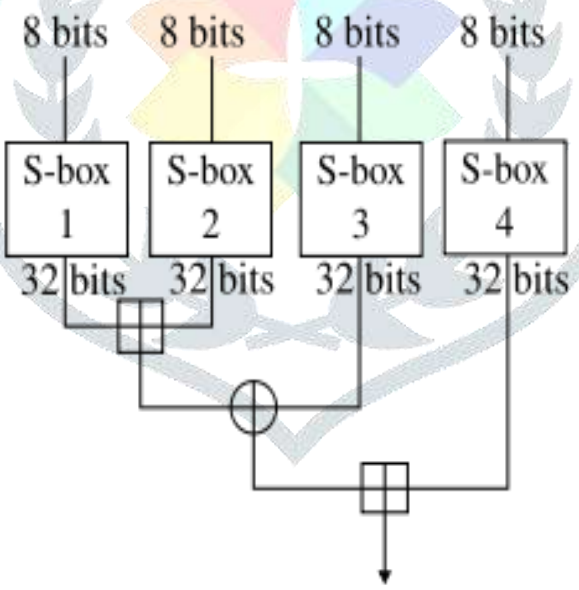

Fig2. no. of substitution boxes

3. Various Security Algorithm

I. Data Encryption Standard (DES)

The National Institute of Standards and Technology (NIST) later authorised the Data Encryption Standard (DES) algorithm, which was developed by an IBM team and is a symmetric-key block cypher.(fig 3) The procedure uses 48-bit keys to encrypt plain text, which is presented in 64-bit blocks. It is possible to defeat the DES algorithm using several technologies. The key used by the DES algorithm is 56 bits long. Using this key, the DES encrypts a block of 64-bit plain text after receiving it, creating a block of 64-bit cypher text.[8][9] Each of the several rounds that make up the DES process is referred to as a step. The quantity of rounds varies depending on the size of the key being used.

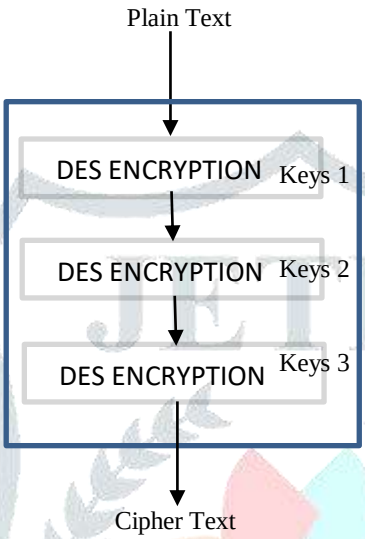

Fig3. DES conversion process

II. 3DES

An encryption cypher called 3DES was created by deriving from the original Data Encryption Standard (DES). Due to the emergence of it gained prominence in the late 1990s, but more secure algorithms like AES-256 and XChaCha20 have since displaced it. It is still used in some circumstances even though it will be deprecated in 2023. The same key is used for encryption and decryption because it is a symmetric key cypher. (fig 4)The network makes each of these methods essentially identical, resulting in a more straightforward procedure.[12] Although the block and key sizes of DES are 64 bits, the key only provides 56 bits of protection. 3DES was developed as a more secure substitute for DES because of its key length restrictions. Although 3 DES uses III separate keys and the DES algorithm III times, it is only considered secure when III different keys are used.

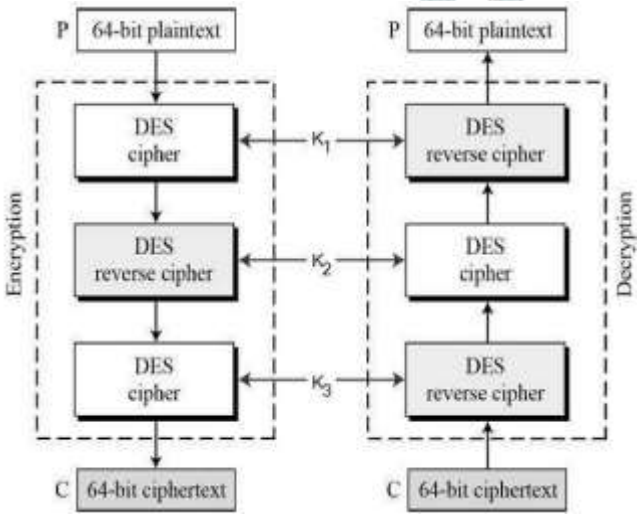

Fig4. 3DES Conversion process

III. Advanced Encryption Standard (AES)

The AES algorithm, also known as the Rijndal algorithm, is a symmetrical blocks cypher that uses keys with lengths of 128, 256 bits to convert plain text into ciphertext. As it is believed to be more secure, the AES algorithm is the established world standard. AES only ever employs bytes for computation, never bits. AES tends to a plaintext block's 128 bits as 16 bytes as a whole result.(fig5) These 16 bytes are organised into 4 strings and 4 rows for matrix processing. The quantity of rounds in AES may varies and is determined on the key size.[14] AES uses 10 series for 128-bit keys, and 14 series for 256-bit keys. Each of these series uses a unique 128-bit series key that is obtained from the first AES keys.

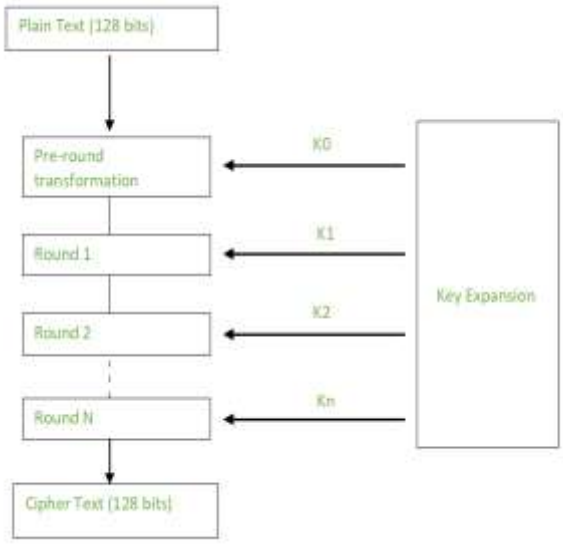

Fig.5 AES Conversion process

IV. Rivest Cipher4 (RC4)

RC4 is a symmetric stream cypher technique used for data encryption and decryption. It is often referred to as R Cipher 4 , Ron's Code 4.

The RC4 technique creates a pseudorandom stream of bytes using a variable-length key (between 1 and 256 bytes), [15]which the ciphertext is produced by XORing with the plaintext(fig 6). Both encryption and decryption employ the same key. RC4 is a well-liked encryption method because it is quick and easy to use, and it also resists some common assaults. It has been discovered that it has some flaws and vulnerabilities, particularly when used with short keys.

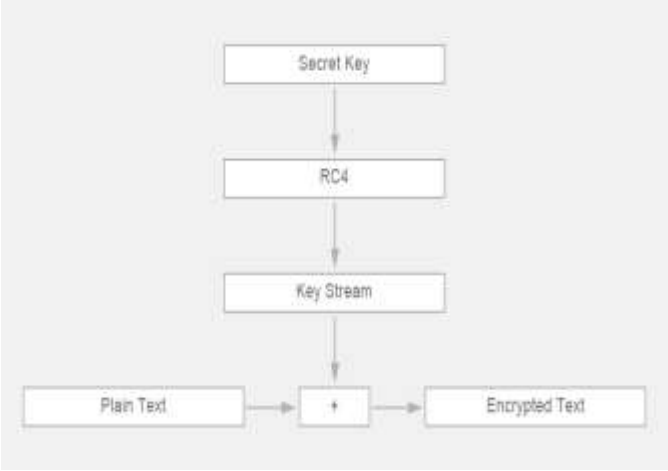

Fig6. RC4 Conversion process

V. Rivest Cipher6 (RC6)

Ron Rivest created the symmetric block cypher technique known as RC6 (Rivest Cipher 6) in 1998. It is a variation of the RC5 algorithm and, in terms of its composition and guiding principles, is comparable to AES (Advanced Encryption Standard).RC6 encrypts and decrypts data in fixed-size blocks, just as other block cyphers. It works with 32-bit words and has a variable-length key (between 0 and 2040 bits). Block sizes of 128, 192, or 256 bits are supported by RC6.In each cycle of the RC6 method, the input block is subjected to an amalgam of substitution, permutation,[16] and modular arithmetic operation utilising the key. The key size and block size affect how many rounds are there.

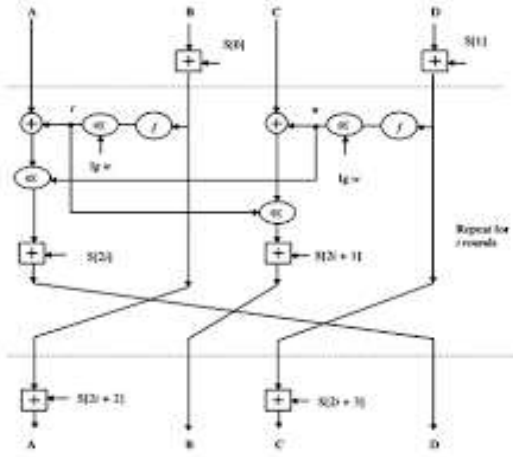

Fig6. RC6 Conversion process

4. Comparison of Algorithms

Table1.Comparison table

❖ **DES**	
Keys Length	64 bits
Block Length	64 bits
Round	16
Level of Security	Not Secure Enough
Encryption Speed	Very Slow
❖ **3DES**	
Keys Length	112 or 118 bits
Block Length	64 bits
Round	48
Level of Security	Adequate Security
Encryption Speed	Very Slow
❖ **AES**	
Keys Length	128,192,256 bits
Block Length	128 bits
Round	10,12,14
Level of Security	Excellent Security
Encryption Speed	Faster
❖ **RC4**	
Keys Length	1-256 bits
Block Length	40-2048 bits
Round	1
Level of Security	Adequate Security
Encryption Speed	Faster
❖ **RC6**	
Keys Length	128-256 bits
Block Length	128 bits
Round	20
Level of Security	Enough Security
Encryption Speed	Average
❖ **BLOWFISH**	
Keys Length	32-248 bits
Block Length	64 bits
Round	16
Level of Security	Highly Secured
Encryption Speed	Very Fast

5. Testing blowfish algorithm in various bits

Table2. various bits

5.1 Using 64 bits

1. Choose a secret key with at least 128 bits (or 16 bytes).We'll use the key 0x2B7E151628AED2A6 ABF7158809CF4F3C for this example.
2. Use the secret key to start the Blowfish cypher.
3. Split the 64-bit plaintext into two 32-bit chunks,
0x12345678 on the left and 0x90abcdef on the rig
ht(right half).
4. Use the Blowfish algorithm to apply an initial key-dependent permutation to each half.
5. Encrypt the two halves of the plaintext 16 times,
switching between the left and right halves each time.
The following steps are included in each round:

Applying these procedures with the secret key 0x2B7E51628AED2A6ABF7158809CF4F3C to the 64-bit plain text 0x 1234567890 abcdef yields the following results:
1.The first secret key is 0x2B7E151628AED2A 6ABF7158809CF4F3C.
2. Use the secret key to start the Blowfish cypher.
3. Right half: 0x90abcdef, left half: 0x12345678.
4. The left half of the initial key-dependent permutations is 0x4F0138C8, while the right half is 0xF9F4C4B8.
5. 16 encryption rounds:
Round 1: The round key at this time is 0x6B056E18759F5CCA.
Right half: 0xF9F4C4B8, left half: 0x335BD687.
Round 2: 0x5EEDB1A07DD1A3AA is the Current round key.
Right half: 0x0D6D75E6, left half: 0x9C4B4D31. \s...
Round 16: 0xB8503C3DC33D3027 is the current round key.
Right half: 0x1E39D7AB, left half: 0x8B0FA81F.
Final swap: Right half: 0x8B0FA81F, left half: 0x1E39D7AB.

5.2 Using 128 bits

Applying these procedures with the secret key 0x2B7E151628AED2A6ABF7158809CF4F3C to the 128-bit plaintext yields the following results:

1.The first secret key is

0x2B7E151628AED2A6ABF7158809CF4F3C.
2. Use the secret key to start the Blowfish cypher.
3. Split the four 32-bit blocks of the 128-bit with the following values for each: block1 = 0x12345678, block2 = 0x90abcdef, block3 = 0xfedcba09, and block4 = 0x87654321.
4. Initial key-dependent permutations for blocks 1, 2, 3, and 4 are 0xB8D14A79, 0x5C8F81F5, 0x27D8CD75, and 0x8351D95A.
5. 16 encryption rounds:
Round 1: The round key at this time is 0x243F6A88D6A61D2.
Blocks 1 through 4 have the values 0x3CAB747B,
1C32445B, 2BCFDBE4, and 6E62C6DE.
Round 2: The round key is currently 0x6A3B50F152637BDB.
block1 is equal to 0x8DD144C, block2 to 0x7C76B201, block3 to 0x9685D1F5, and block4 to 0x1E5B8BC9.
Round 16: 0xCBCA5FEF94341A27 is the current round key.
Blocks 1 through 4 have the values 0xAF7B460C, 0xC920AC1A, 0x6342A9A9,
and 0xA0C0DDE6.
In order to create the 128-bit ciphertext 0xA298DE9709B907A858091A485DF6A, the128-bitplaintext 0x1234567890abcdeffedcba0987654321 is encrypted. utilising the secret key

0x2B7E151628AED2A6ABF7158CF4F3C using the Blowfish method.

6. The following key-dependent permutations are the final ones: Left half: 0x879B90A5, right half: 0x3D9EE9C1. 7. Concatenation: 0x879B90A53D9EE9C1 is the ciphertext. Thus,the64-bitplaintext 0x1234567890abcdef was encrypted using the Blowfish technique and the secret key '0x2B7.	6. Change blocks 2 and 3 to 0xAF7B460C and 0x6342A9, respectively. 7.Change the second and third ciphertext blocks: Blocks 1 through 4 have the values 0xAF7B460C,0x6342A9A9, 0xC920AC1A, and 0xA0C0DDE6. 8.Key-dependent final permutations: Blocks 1 through 4 have the values 0xA298DE97, 0x09B907A8, 0x58091A48, and 0x5A05DF6A. 9.The final 128-bit ciphertext is created by concatenating the four 32-bit blocks of the ciphertext: 0xA298DE9709B907A858091A485A05DF6A

5.3 Using 256 bits

To use Blowfish to encrypt 256-bit plaintext, we must split it into 4 blocks of 64 bits each since Blowfish only supports 64-bit blocks. Then, using Blowfish, we can encrypt every block independently.
Eight 32-bit subkeys are created from the 64-bit key:

5.4 Using 448 bits

We can't directly encrypt a 448-bit plaintext using Blowfish because it only supports 64-bit blocks of plaintext. Using a mode of operation One standard technique for encrypting larger messages using Blowfish is Cypher Block Chaining (CBC) or Counter (CTR) mode. Using an initialization vector (IV) that is 128-bit wide and a key that produced at random, we may encrypt a 448-bit plaintext using Blowfish as follows:

6. Result

` The symmetric-key encryption method known as Blowfish has been around for a while and is regarded as secure. It is quicker than several other widely used encryption algorithms and strikes a fair mix between security and effectiveness.

` In general, it is desirable to have a bigger block length for encryption because it makes some sorts of attacks more challenging. The application- specific
requirements and the available resources will determine the block length, though.
Although Blowfish can handle blocks up to 448 bits in length, the most used block length is 64 bits. This is due to the fact that many applications need to be able to encrypt and decrypt messages that are only a few kilobytes in length, and 64-bit blocks are adequate for this.
It could be essential to utilise an encryption mode like Cipher Block Chaining (CBC) or Counter (CTR) mode if longer messages need to be encrypted. In these settings, the plaintext is broken up into blocks and the Blowfish method is used to individually encrypt each block.
The selection of block length is dependent on the particular needs of the programme and the resources that are available, however both 64-bit and 128-bit block lengths are frequently employed with Blowfish.

7. Conclusion

The symmetric-key block cypher was developed by B. Schneier. encryption method known as Blowfish . It is frequently employed in applications where a quick and effective encryption technique with a configurable key length is needed. Blowfish's adaptability in allowing varied key lengths, which may be between 32 bits to 448 bits—is one of its primary advantages. This qualifies it for a variety of security needs and applications.[18] Blowfish can also be utilised in contexts with limited resources and is quite simple to implement.

The selection of an encryption method ultimately depends on the particular security requirements of the application, even though Blowfish is a well-known encryption technique that has been extensively utilised and researched over the years. Applications that need better security guarantees or where more data needs to be secured should take into account more advanced and secure encryption techniques.

The choice of block length depends on the program's specific requirements and the available resources, however both 64-bit and 128-bit block lengths are often used with Blowfish. Blowfish is a well-known encryption technique that has undergone extensive testing and research.[19] It nevertheless makes a decent choice for applications that need a quick and effective encryption technique with a configurable key length despite its drawbacks. For applications that demand higher security guarantees or where bigger amounts of data need to be encrypted, newer encryption methods should be taken into account.

The security of the system ultimately depends on other aspects, such as key management and implementation specifics, and no encryption algorithm can guarantee 100% security. As a result, selecting an encryption technique is only one element of an overall security plan for RFID devices.[20]

To sum up, Blowfish can be a solid option for protecting RFID systems, but other elements like key management, implementation specifics, and overall security strategy should also be carefully taken into account to provide the highest level of security.

8. REFERENCES

1. B. Schneier, 1994. Chapter 14 of Applied Cryptography: Protocol, Algorithm, and Source Code , 2nd Edition.
2. B. Schneier (1993), A New 64-Bit Block Cypher with Variable-Length Key is Described (Blowfish). Proceedings of the faster Software Encryption Workshop at Cambridge.
3. J Kelsey, B Schnier, David Wagnr, and C Hall wrote "Blowfish: A Cryptographic Algorithm".
4. Author: Eli Biham; Publisher: Springer-V Berlin and Heidelberg GmbH & Co. KG; "Fast Software Encryption: IV International Workshops" .
5. HAsaeda, Line Guo, L Hao Yue, the Internet of Thing Journal, Vol. 1,Oct 2014; Data Cloud: Developing Data-Centric Services for the Community
6. Internet of Things Trust Management Mechanism by Lizet, WJingp, and S Bin was published in China Communication Magazine in Feb 2014
7. From Research and Innovation to Market Deployment: The Internet of Thing O.Vermesan, 2010 River Publishers' Communication Series
8. Internet of Things: Applications, Research Challenges, and Future Directions, D. Morandi and S. Sicari, Journal, Elsevier, April 2012.
9. International Journal of Computer Networks article titled "Towards Internet of Things: Survey and Future Vision," Volume. 5, 2013.
10. Yihe Liu, Quandeng Gou, and IoT Security System Design and Implementation, International Conference on Green Computing
11. Stephen B. Miles' "RFID Technology and Applications": This book gives a general review of RFID technology, covering its concepts, uses, and difficulties,2020.
12. by Lei Chen, Hui Li, and Xiaolan Zhang, "Handbook of RFID Security: Fundamentals, Applications, and Challenges": This book discusses a variety of RFID security topics, such as attacks, protocols, and defences.
13. Harvey Lehpamer's "RFID Design Principles": A practical manual for constructing RFID systems, including antenna design, power management, and system integration, is provided in this book. Edited by Jalel Ben-Othman, Abdelhamid Mellouk, and Mohamed Essaaidi, "RFID Systems: Research Trends and Challenges": This book discusses current issues in RFID research, such as localisation, privacy, and energy efficiency.
14. By Khalid A. Alrawi and Abdul Sattar Abdul Jabbar, "Implementation and Evaluation of a Secure RFID System Using Blowfish Algorithm": This study describes the implementation and assessment of a secure RFID system that encrypts and decrypts data using the Blowfish algorithm.
15. Jianguo Liu and Qingyuan Wang's "Design and Implementation of a Secure RFID System on Blowfish Algorithm".
16. A Compact Radio Frequency Authentication Protocol Blowfish Algorithm" by Zhengping Jin and Wei Wu, and its effectiveness is assessed.

17. By Zhiming Liu and Jia Liu, "A Secure and Lightweight RFID Authentication Protocol Based on Blowfish Algorithm": This study analyses the security and effectiveness of a proposed secure and compact RFID authentication mechanism based on the Blowfish algorithm.
18. By Zhiming Liu and Jia Liu, "A Secure and Lightweight RFID Authentication Protocol Based on Blowfish Algorithm": This study analyses the security and effectiveness of a proposed secure and compact RFID authentication mechanism based on the Blowfish algorithm.
19. LCasto ,S Foso,Éco Polytechnic Material “A DEEP DIVE INTO RFID TECHNOLOGY”
20. "The RFID Guidebook (R 8)", S DCS & Label Worldwide, 2022