Journal of Engineering Sciences
Vol 11, Issue 5,May/2020
ISSN NO:0377-9254

IoT Enabling and Device Level Security Using Lightweight Encryption Algorithms

D. SATYA BHAVANI [1], SOWMYA KALUSANI [2]
Asst. Professor [1], UG Scholar[2]
dsbhavani_cse@mgit.ac.in[1], sowmya.kalusani@gmail.com[2]
Department of Computer Science and Engineering[1,2]
Mahatma Gandhi Institute of Technology, Hyderabad, Telangana

ABSTRACT - Internet of Things (IoT) being a promising technology of the future is expected to connect billions of devices. Recent advancements in wireless technology have created an exponential rise in the number of connected devices leading to the internet of things (IoT) revolution. Large amounts of data is captured, processed and transmitted through the network by these embedded devices. Security of the data transmitted by these devices is a major area of concern in IoT. Many encryption algorithms have been proposed by the researchers in these years to ensure security of transmitted data through the IoT network. The devices which are a part of this network are generally smaller in size and consume less power. Conventional encryption algorithms are very complex in nature and require many rounds to encrypt, wasting the energy of these devices. Less complex algorithm, however, may compromise the desired integrity. In this paper we propose a lightweight encryption algorithm named as Secure IoT (SIT). It is a 64-bit block cipher and requires 64-bit key to encrypt the data. The architecture of the algorithm is a mixture of feistel structure and a uniform substitution-permutation network. Results show that the algorithm provides substantial security in just five encryption rounds.

Index Terms - IoT, Security, Encryption, SIT, Feistel, Substitution - Permutation.

I. INTRODUCTION

The Internet of Things(IoT) is described as a network of physical objects - "things"- that are embedded with

Sensors, software and other technologies for the Purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industry tools. In IoT, All things that are connected to the internet as a part can be put into three categories :

1. Things that collect information and send it .
2. Things that receive information and then act on it.
3. Things that collect and receive information.

As the broadband Internet services are easily accessible and its cost of connectivity is also reduced, more devices are getting connected to it. Such conditions are providing suitable growth of IoT.

A. Security in IoT

With rapid increase in the use of IoT applications, several security and privacy issues are observed. When nearly everything in the world being connected to each other, this issue will only become more noticeable and constant exposure will reveal the security flaws and weakness. Scalability factors and various restrictions on device capabilities also intend to convey that conventional cryptographic mechanisms, security protocols and protection mechanisms are insufficient. Security can be defined as an organized framework which consists of concepts , beliefs, principles, policies, procedures , techniques and measures required to protect individual system assets and as well as the whole system against any intentional or unintentional threat. The three core issues in the IoT network are privacy for humans, confidentiality of business process and third party dependability. In a IoT network, there are four interconnected interacting components which are as follows : 1) People 2) Objects 3) Software 4) Hardware

These components are bound to be confronted with security, privacy and open trust problems. IoT faces many active and passive attacks that may easily hinder its functionality and negate the benefits of using its services.

Passive Attacks - They are able to recover information from the network but do not impact its behavior.

Active Attack - They directly deter its service.

Threats can be classified into two categories ,external and internal threats.

External Threats - They originate from outside the network.

Internal Threats - They originate from within the network.

II. RELATED WORK

Francois-Xavier Standaert, Gilles Piret, Gael Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat [1] proposed an algorithm ICEBERG which is a fast involutional block cipher which is optimized for

reconfigurable hardware implementations. It uses 64-bit text blocks and 128-bit keys. All components allow efficient combinations of encryption and decryption. Hardware implementations of ICEBERG allow a change in the key at every clock cycle without any degradation in performance and its round keys are derived during the process of encryption and decryption eliminating the need to store the round keys. The resulting design offers better hardware efficiency than other 128-key-bit block ciphers. Resistance against side-channel cryptanalysis was considered as a design criteria for ICEBERG. Reconfigurable hardware devices usually enable encryption/decryption solutions which provide high performance for real-time applications that have multiple data streams. Video-processing is atypical context where high throughput has to be provided at low hardware cost. Although present encryption algorithms may provide very high encryption rates, it is often at the cost of expensive designs. ICEBERG is designed in a way which allows very efficient reconfigurable hardware implementations. An additional criteria for design was simplicity . It is scalable for different architectures (loop, unrolled, pipeline) and FPGA2 technologies.

Advantages -

1.It allows efficient combinations of encryption/decryption.
2. It provides resistance against side-channel cryptanalysis.
3. Simplicity of design.
4. It is scalable for different architectures.
5. Parallel Nature.

Disadvantages -

1. Key expansion is critical.

Francois-Xavier Standaert, Gilles Piret, Neil Gershenfeld and Jean-Jacques Quisquater [2] proposed an algorithm SEA which is a scalable encryption algorithm targeted for small embedded applications. The plaintext size, key size and processor (or word) size are parameters of the design. SEA operates on various sizes of text, key and word. Its design is based on Feistel structure with a variable number of rounds, which is defined with respect to the following parameters:

1. n: plaintext size, key size.
2. b: processor (or word) size.
3. nb = n 2b : number of words per Feistel branch.
4. nr: number of block cipher rounds.

SEA provides a suitable solution for low-cost encryption/authentication within such networks. RFID's or any power/space-limited applications are similarly targeted. SEA is based on a limited number of elementary operations (selected for their availability in any processing device) denoted as follows: (1) bitwise XOR $\oplus$, (2) substitution box S, (3) word (left) rotation R and inverse word rotation R^{-1}, (4) bit rotation r, (5) addition mod 2b . The structure of SEA allows a faster evaluation of the cipher efficiency on any RISC machine. Its typical performances (encryption + decryption) for present key sizes and processors (e.g. 128-bit key, 1 Mhz 8-bit RISC) are in the range of a few milliseconds, using a few hundred bytes of ROM..

Advantages -

1. Extremely simple design.
2. Flexible in nature.
3.It combines encryption and decryption more efficiently.

Disadvantages -

1.Susceptible to related key attack or interpolation attack.

Deukjo Hong1, Jaechul Sung2, Seokhie Hong1, Jongin Lim1, Sangjin Lee1[3] proposed an algorithm HIGHT. It is a new block cipher with 64-bit block size and 128-bit key length. It provides low-resource hardware implementation, which is suitable for a ubiquitous computing device such as a sensor in USN or a RFID tag. HIGHT consists of simple operations security as a good encryption algorithm. The hardware implementation of HIGHT requires 3048 gates on 0.25 μm technology. HIGHT is a variant of Feistel network which has a 32-round iterative structure. The most prominent feature of HIGHT is that it consists of simple operations such as XOR, addition mod 28, and left bitwise rotation. So, it is said to be hardware-oriented rather than software-oriented.

Advantages -

1. It is useful for pervasive computing devices.
2. It consists of simple operations like XOR.
3.Suitable for low-cost, low-power, and ultra-light implementation.

Disadvantages -

1. Susceptible to saturation attack.

Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm[4] proposed a lightweight variant of DES called DESL , a new block cipher which is based on the classical DES design, but unlike DES it uses a single S-box that is repeated eight times. DESL is resistant against linear and differential cryptanalysis, and the Davies-Murphy attack . It is well suited for ultra constrained devices such as RFID tags. The main design ideas of the new cipher family, a variant of DES, are:

1. Using a serial hardware architecture which reduces the gate complexity.

2. Optionally apply key-whitening in order to increase the security of the cipher.
3. Replacing the 8 original S-Boxes by a single S-box which further reduces the gate complexity.

Advantages -

1.Provides resistance against certain types of common attacks
2. It has low power consumption and chip size is small.
3. Well suited for ultra-constrained devices such as RFID tags.

Disadvantages -

1. Does not allow any changes in feistel cipher structure.

Jian Guo1, Thomas Peyrin2,, Axel Poschmann2,and Matt Robshaw3 [5] implemented LED block cipher. It is a 64-bit block cipher with two different key sizes , 64- and 128-bit keys. The LED block cipher is simple to analyze and this allows us to precisely evaluate the necessary number of rounds to ensure proper security. The LED key is known for its simplicity and security. Since it is very simple to analyze, it allows us to directly derive a bound on the minimal number of active Sboxes, even in the scenario of related-key attacks. LED cipher is similar to AES , so one can directly reuse extensive work that has been done on the AES.

Advantages -

1. It is simple to analyze.
2. Allows us to evaluate the necessary no. of rounds to ensure proper security.
3.It is resistant to classical attacks and related key attacks.

Disadvantages-

1. Software implementations of LED are not known.
2. The security level can be adjusted to the required but has low throughput.
3. Vulnerable to biclique cryptanalysis.

Daniel Engels, Markku-Juhani O. Saarinen, Peter Schweitzer, and Eric M. Smith [6] Hummingbird-2 is an encryption algorithm with a 128-bit secret key and a 64-bit initialization vector. Hummingbird-2 produces an authentication tag for each message processed. Hummingbird-2 has been targeted for low-end microcontrollers and for hardware implementation in lightweight devices such as RFID tags and wireless sensors. Hummingbird-2 is an authenticating encryption algorithm that has been designed particularly for resource-constrained devices such as RFID tags, wireless sensors, smart meters and industrial controllers. Hummingbird-2 can be implemented with minimal hardware or software footprint and is therefore suitable for providing security in low-cost ubiquitous devices. The Hummingbird-2 does not directly fall into either traditional stream cipher or block cipher categories as it inherits its properties from both categories. Hummingbird-2 has been designed to provide resistance to all the standard attacks against all block ciphers and stream ciphers for ex. differential and linear cryptanalysis, structure attacks and various algebraic attacks. Hummingbird-2 also has an added advantage of being resistant to chosen-IV attacks.

Advantages -

1. Resistant to all standard attacks against block and stream ciphers.
2. Consumes less power.

Disadvantages -

1.Does not provide resistance to various cryptanalysis attacks including the related key attack.

Sreeja Rajesh, Varghese Paul , Varun G. Menon , and Mohammad R. Khosravi [7] proposed TEA algorithm . It uses Feistel structure with 64 rounds or 32 cycles where one cycle consists of two rounds. The size of plaintext is 64 bits The recommended key size is 128 bits. Recent advancements in the wireless technology have created an exponential rise in the number of connected devices leading to the internet of things (IoT) revolution. Large amounts of data are captured, processed and transmitted through the network by these embedded devices. Security of the transmitted data is a major area of concern in IoT networks.

Numerous encryption algorithms have been proposed by researchers in these years to ensure the security of the data transmitted through the IoT network. Tiny encryption algorithm (TEA) is the utilizes less memory and is easy to implement on both software and hardware platforms. The major drawback of TEA is it uses the same key in all the rounds of encryption , which yields a reduced security evident from avalanche effect of the algorithm . The time required for encrypting and decrypting a text is high, leading to lower efficiency in IoT networks with embedded devices.

Advantages -

1. Ease of implementation.
2. Less memory utilization compared to all other symmetric encryptions.

Disadvantages -

1. Usage of same key in all rounds of encryption leads to reduced security.
2. Time taken for encryption and decryption is high, leading to reduced efficiency.

III. SYSTEM DESIGN

The SIT algorithm falls into the category of symmetric

key algorithms. The algorithm works on the 64 bit data block and 64 bit encryption key .SIT uses a combination of feistel and substitution-permutation network, that allows similar steps for encryption and decryption, decreasing overall complexity of the system by reducing number of required computational steps. The SIT algorithm consists of five encryption rounds which require five distinct keys. 64 bit key is taken as input from the user and is expanded to generate keys for each round. The data is encrypted using the expanded keys .

A. Key Expansion

The key expansion process, plays a vital role in the process of encryption and decryption. The key expansion (shown in Fig No.1) block in SIT is required to generate five unique keys. The user is prompted to give 64-bit number as input, which is broken down into 4-bit chunks and after substitution and diffusion and these chunks are fed into the *f*-function block. The key expansion block works on the following steps:

1.The cipher key of size 64-bit is broken down in 16 segments of 4-bits each. Four *f* function blocks work on the data of 16-bit and the initial substitution is performed using the following equation .

$$Kb_jf = P_{i=1}^{4} Kc_{(i-1)+j} \quad (1)$$

where Kc shows the cipher key input given by the user. For first four rounds i = 1-4, the expanded keys are calculated using the permutations of the bits of the initially provided cipher key. In the next round the 16-bits of are passed to the *f*-function to further expand the key using : $Kajf = f(Kbjf)$ (2). The transformation used in the *f*-function is based on the P and Q tables presented in Fig No. 1. The P and Q tables perform linear and non-linear transformations resulting in confusion and diffusion.

Table 1. Summary of P and Q

Kc_j	P-Values $P(Kc_j)$	Q-Values $Q(Kc_j)$	Kc_j	P-Values $P(Kc_j)$	Q-Values $Q(Kc_j)$
0	3	0	8	D	F
1	F	E	9	A	0
2	E	5	A	9	4
3	0	6	B	6	D
4	5	A	C	7	7
5	4	2	D	8	B
6	B	3	E	2	1
7	C	C	F	1	8

Fig No. 1: Key Expansion

4. The fifth key is obtained by performing an XOR operation between the four round keys K1,K2,K3 and K4 as expressed in equation (3).

$$K5 = K1 \oplus K2 \oplus K3 \oplus K4 \quad (3)$$

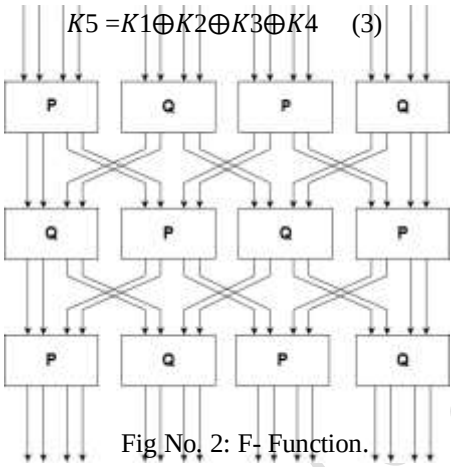


Fig No. 2: F- Function.

F-function is comprised of P and Q tables. These tables perform linear and non-linear transformations resulting in confusion and diffusion as illustrated in Fig No. 2. The Key Expansion block is shown in Fig No. 3.

The output of each f-function block is arranged in 4×4 matrix named Km shown below.

$$Km_1 = \begin{bmatrix} Ka_1f_1 & Ka_1f_2 & Ka_1f_3 & Ka_1f_4 \\ Ka_1f_5 & Ka_1f_6 & Ka_1f_7 & Ka_1f_8 \\ Ka_1f_9 & Ka_1f_{10} & Ka_1f_{11} & Ka_1f_{12} \\ Ka_1f_{13} & Ka_1f_{14} & Ka_1f_{15} & Ka_1f_{16} \end{bmatrix}$$

$$Km_2 = \begin{bmatrix} Ka_2f_1 & Ka_2f_2 & Ka_2f_3 & Ka_2f_4 \\ Ka_2f_5 & Ka_2f_6 & Ka_2f_7 & Ka_2f_8 \\ Ka_2f_9 & Ka_2f_{10} & Ka_2f_{11} & Ka_2f_{12} \\ Ka_2f_{13} & Ka_2f_{14} & Ka_2f_{15} & Ka_2f_{16} \end{bmatrix}$$

$$Km_3 = \begin{bmatrix} Ka_3f_1 & Ka_3f_2 & Ka_3f_3 & Ka_3f_4 \\ Ka_3f_5 & Ka_3f_6 & Ka_3f_7 & Ka_3f_8 \\ Ka_3f_9 & Ka_3f_{10} & Ka_3f_{11} & Ka_3f_{12} \\ Ka_3f_{13} & Ka_3f_{14} & Ka_3f_{15} & Ka_3f_{16} \end{bmatrix}$$

$$Km_4 = \begin{bmatrix} Ka_4f_1 & Ka_4f_2 & Ka_4f_3 & Ka_4f_4 \\ Ka_4f_5 & Ka_4f_6 & Ka_4f_7 & Ka_4f_8 \\ Ka_4f_9 & Ka_4f_{10} & Ka_4f_{11} & Ka_4f_{12} \\ Ka_4f_{13} & Ka_4f_{14} & Ka_4f_{15} & Ka_4f_{16} \end{bmatrix}$$

To obtain round keys, K1, K2, K3 and K4 the matrices are transformed into four arrays of 16 bits that are called round keys(Kr). The arrangement of these bits is shown in equations (7), (8), (9) and (10).

$$K1 = a_4 + a_3 + a_2 + a_1 + a_5 + a_6 + a_7 + a_8 + a_{12} + a_{11} + a_{10} + a_9 + a_{13} + a_{14} + a_{15} + a_{16} \quad (7)$$

$$K2 = b_1 + b_5 + b_9 + b_{13} + b_{14} + b_{10} + b_6 + b_2 + b_3 + b_7 + b_{11} + b_{15} + b_{16} + b_{12} + b_8 + b_4 \quad (8)$$

$$K3 = c_1 + c_2 + c_3 + c_4 + c_8 + c_7 + c_6 + c_5 + c_9 + c_{10} + c_{11} + c_{12} + c_{16} + c_{15} + c_{14} + c_{13} \quad (9)$$

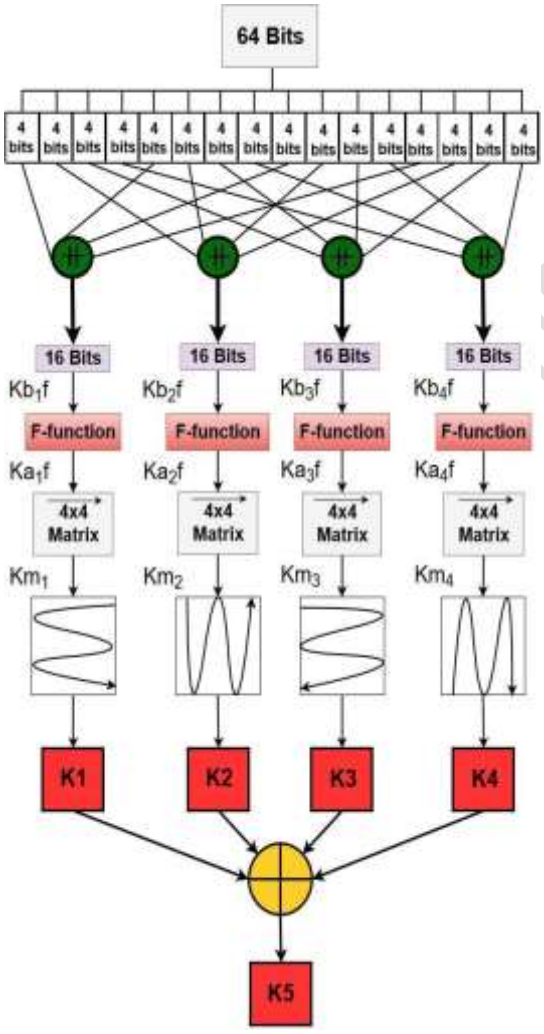
$$K4 = d_{13} + d_9 + d_5 + d_1 + d_2 + d_6 + d_{10} + d_{14} + d_{15} + d_{11} + d_7 + d_3 + d_4 + d_8 + d_{12} + d_{16} \quad (10)$$

- An *XOR* operation is performed among the four round keys to obtain the fifth key as shown in equation (11).

$$K5 = \bigoplus_{i=1}^{4} Ki \quad (11)$$

The Key Expansion block is shown in Fig No.3.

Fig No.3: Key Expansion Block



B. Encryption

The encryption process shown in Fig. 4, starts after obtaining all round keys. Data diffusion and confusion is attained by shift and logical operations. 16-bit segments are extracted from the 64-bit data which are represented by $Px0{-}15$, $Px16{-}31$, $Px32{-}47$ and $Px48{-}63$. To increase the confusion in cipher text , text swapping in performed at each round. Bitwise *XNOR* operation is performed between the cipher key $K1$ & $Px0{-}15$ and $K4$ & $Px48{-}63$ to obtain $RO11$ and $RO14$respectively in the first round. The result of *XNOR* operation is fed as input to the $\boldsymbol{f}$-function generating $Efl1$ and $Efr1$.The $\boldsymbol{f}$-function used in encryption process is the same as that of the key expansion process. Bitwise *XOR* is then applied between $Efl1$ & $Px32{-}47$ and $Efr1$ & $Px16{-}31$ to obtain $RO12$ and $RO13$ respectively.

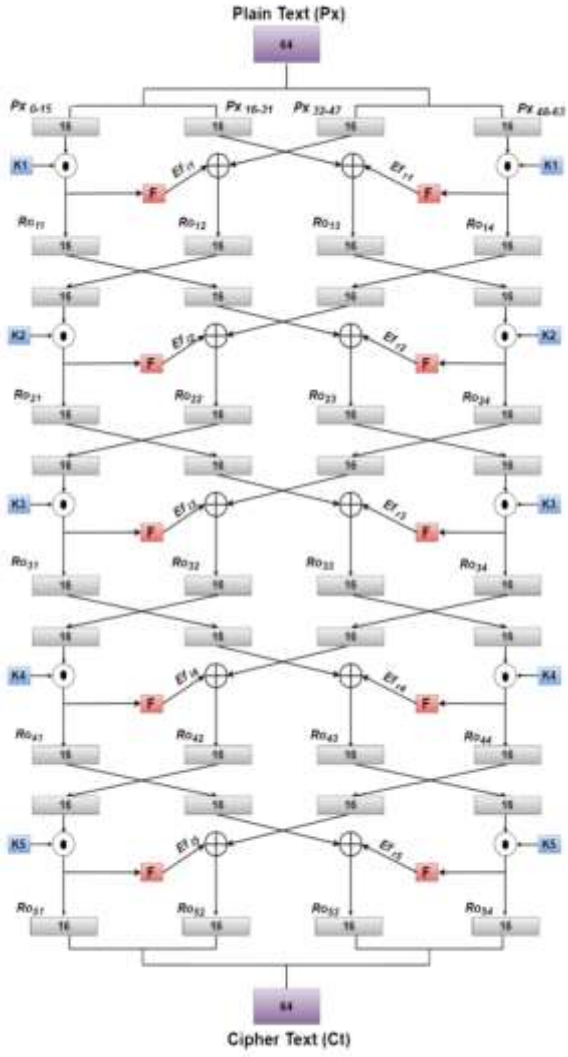


Fig No. 4 : Encryption Process.

$$Roj,i = \{$$
$$Pxj, \odot Kj \; ; i = 1,4$$
$$Pxj,+1 \oplus Eflj \; ; i = 2 \quad (4)$$
$$Pxj,-1 \oplus Efrj \; ; i = 3$$

The transformation is made in such a way that for the

next round of encryption , $RO11$ will become $Px16{-}31$, $RO12$ will become $Px0{-}15$, $RO13$ will become $Px48{-}63$ and $RO14$ will become $Px32{-}47$. The transformed data segments again encrypted using equation with the second key generated from the key expansion. The process is continued for all the five round keys. The final round results are concatenated together to extract the Cipher Text (Ct) given by Ct = Concatenate ($R51$, $R52$, $R53$, $R54$) (5) .

C. Decryption

The decryption process is the reverse of the encryption process .The first round of encryption is performed using K_5 . The same process is continued with keys K4,K3,K2 and K1 to obtain the plaintext.

IV. RESULTS AND DISCUSSIONS

The Key Expansion Algorithm takes 64 bit input from the user and return five unique keys (K1,K2,K3,K4 and K5) which are used for encryption. For example : The following 64 bit input 0000001100101000000000110010100000000011001010000000001100101000 when given as input to key expansion algorithm , the round keys generated are shown below in Fig. 5.

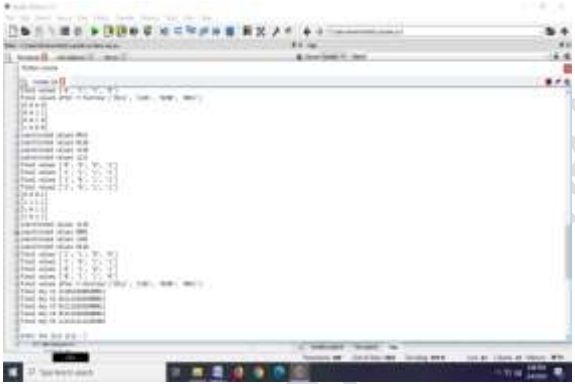


Fig No.5 : Round Keys.

64 0's are taken as input (shown in Fig.6) by Encryption Algorithm and the resultant cipher text is shown below in Fig.7.

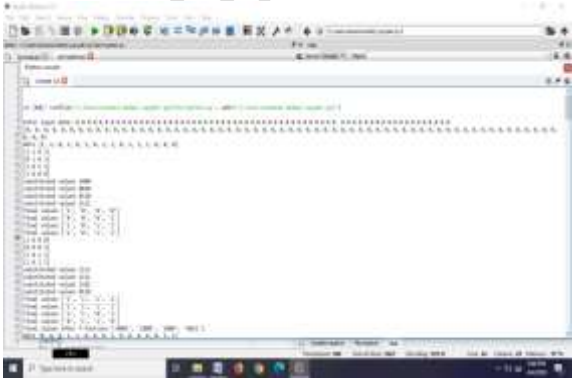


Fig No. 6 : 64 bit input to Encryption algorithm.

Fig. 7 : Cipher Text generated by Encryption Algorithm.

V. CONCLUSION AND FUTURE SCOPE

A. CONCLUSION

Block ciphers are basic components for designing applications. A number of cryptanalysts have stated that there exist numerous attacks on ciphers. Therefore ciphers must provide good resistance. As internet services have become cheap and easily accessible, The number of devices getting connected to a network are gradually increasing. As a result huge amount of data is generated and transmitted over the network. So the success of IoT is strongly dependent on the security and privacy. Since the devices connected in an IoT network are low energy devices , conventional cryptographic algorithms which drain the energy cannot be used to provide security. The lightweight encryption algorithms come into picture in such a scenario. IoT being emerging field requires lightweight cipher designs having rich encryption standards, robust architecture, less complexity, less execution time, lower power consumption, low resource utilization and good resistance against possible attacks. The SIT algorithm described in this paper, is best suited for providing security to low energy constrained devices in a IoT network. SIT , uses a combination of Feistel structure and Substitution- Permutation(SP) networks provides security to the devices connected in a network with only 5 rounds of encryption , where each round uses a unique key . The Avalanche test of the algorithm shows that single bit change in key or plaintext results in change of 26 bits in the resultant cipher text.

B. FUTURE SCOPE

For future research, the implementation of the algorithm on hardware and software in various computation and network environment is under consideration. To enhance the performance according to different hardware platforms , the algorithm can be optimized. Hardware like FPGA performs the parallel

execution of the code and the implementation of the algorithm on an FPGA is expected to provide high throughput. The scalability of algorithm can be exploited for better security and performance by changing the number of rounds or the architecture to support different key length. IoT application to transmit sensor data securely can be achieved .

REFERENCES

[1] Francois-Xavier Standaert, Gilles Piret, Gael Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat, ICEBERG : An Involutional Cipher Efficient for Block Encryption in Reconfigurable Hardware,2004.

[2] Francois-Xavier Standaert, Gilles Piret, Neil Gershenfeld and Jean-Jacques Quisquater, SEA: A Scalable Encryption Algorithm for Small Embedded Applications,2006.

[3] Deukjo Hong1, Jaechul Sung2, Seokhie Hong1, Jongin Lim1, Sangjin Lee1, HIGHT: A New Block Cipher Suitable for Low-Resource Device,2006.

[4] Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm, New Lightweight DES Variants,2007.

[5] Jian Guo1, Thomas Peyrin2,, Axel Poschmann2,and Matt Robshaw3, The LED Block Cipher,2011.

[6] Daniel Engels, Markku-Juhani O. Saarinen, Peter Schweitzer, and Eric M. Smith, The Hummingbird-2 Lightweight Authenticated Encryption Algorithm,2012.

[7] Sreeja Rajesh, Varghese Paul , Varun G. Menon , and Mohammad R. Khosravi, A Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices,2019.

[8] Muhammad Usman_, Irfan Ahmedy, M. Imran Aslamy, Shujaat Khan_ and Usman Ali Shah , SIT: A Lightweight Encryption Algorithm for Secure Internet of Things,2017.