

IN SEARCH OF AN EFFICIENT LIGHTWEIGHT CRYPTOGRAPHY ALGORITHM FOR IOT DEVICES: A COMPARATIVE REVIEW

by

M.M Jubaid Hassan (170021019)

Ishtyaq Tahmid (170021023)

Shah Md. Sagar Chowdhury (170021045)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements
for the Degree of

**BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC
ENGINEERING**



Department of Electrical and Electronic Engineering

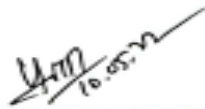
Islamic University of Technology (IUT)

Gazipur, Bangladesh

May 2022

IN SEARCH OF AN EFFICIENT LIGHTWEIGHT CRYPTOGRAPHY ALGORITHM FOR IOT DEVICES: A COMPARATIVE REVIEW

Approved by:



Handwritten signature of Dr. Nafiz Imtiaz Bin Hamid, dated 10.05.22.

Dr. Nafiz Imtiaz Bin Hamid

Supervisor and Assistant Professor,

Department of Electrical and Electronic Engineering,

Islamic University of Technology (IUT),
Boardbazar, Gazipur-1704.

Date: 10/05/2022

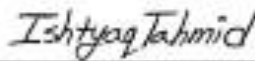
Declaration of Authorship

This is the attestation of the work done in the thesis paper by the students M.M Jubaid Hassan, Ishtyaq Tahmid and Shah Md. Sagar Chowdhury under the supervision of Dr. Nafiz Imtiaz Bin Hamid, Assistant Professor of Department of Electrical and Electronic Engineering (EEE), Islamic University of Technology (IUT).

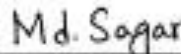
Authors



M.M Jubaid Hassan
ID-170021019



Ishtyaq Tahmid
ID-170021023



Shah Md. Sagar Chowdhury
ID-170021045

Table of Contents

List of Tables	i
List of Figures	ii
List of Acronyms	iii
Acknowledgements	iv
Abstract	v
1 Introduction	6
1.1 Basic Goal of Lightweight Encryption Algorithm	7
1.2 Different Aspects of IoT Data Security	8
1.3 Background and Motivation	9
2 Overview of algorithms	10
2.1 Outcomes of the algorithms	12
2.2 Performance Analysis	22
2.3 Security Analysis	28
3 Challenges	30
4 Conclusion	31
4.1 Future work	32
5 References	33

List of Tables

Table: 2.1	Correlation & entropy measurement of SIT	12
Table: 2.2	Comparison among SIT, Skipjack and RC5	20
Table: 2.3	Comparison among HIGHT, Skipjack, RC5 and SIT in case of operation time	23
Table: 2.4	Comparison among HIGHT, Skipjack, RC5 and SIT in case of CPU cycles	23
Table: 2.5	Comparison among eight algorithms in case of performance analysis	24
Table: 2.6	Comparison among five algorithms in case of RAM & ROM	26

List of Figures

Figure 2.1: Encryption & decryption of onion	13
Figure: 2.2: Histogram of entropy for onion(before and after encryption)	13
Figure: 2.3: Correlation for onion(before and after encryption)	14
Figure 2.4: Encryption & decryption of football	14
Figure: 2.5: Histogram of entropy for football(before and after encryption)	15
Figure: 2.6: Correlation for football(before and after encryption)	15
Figure 2.7: Result for Skipjack algorithm	16
Figure 2.8: Result for RC5 algorithm	17
Figure 2.9: Result for Hummingbird algorithm (23 benchmark function)	17
Figure 2.10: Fitness vs Iteration curve of Hummingbird algorithm (23 benchmark function)	18
Figure 2.11: Result for Hummingbird algorithm (50 benchmark function)	18
Figure 2.12: Fitness vs Iteration curve of Hummingbird algorithm (23 benchmark function)	19
Figure 2.13: Initial state of AES algorithm	19
Figure 2.14: Round key of AES algorithm	20
Figure 2.15: Final state of AES algorithm	20
Figure 2.16: Comparison among SIT, Skipjack and RC5	21
Figure 2.17: Comparison among HIGHT, Skipjack, RC5 & SIT	22
Figure 2.18: Comparison among eight algorithms in case of cycles	25
Figure 2.19: Comparison among six algorithms in case of RAM	26
Figure 2.20: Comparison among five algorithms in case of RAM and ROM	27

List of Acronyms

IoT	Internet of things
SIT	Secure Internet of Things
LWC	Lightweight Cryptography
AES	Advanced Encryption Standard
RC	Rivest Cipher
DES	Data Encryption Standard
RAM	Random Access Memory
ROM	Read Only Memory

Acknowledgements

Assalamualaikum. I would like to thank Allah for the opportunities that were provided to us. We consider ourselves lucky to be able to work on this thesis topic and find relevant data that will surely help the coming generation with their work on IoT platforms. We would like to recognize the efforts of the supervisor for being there through thick and thin and helping us understand how things are done. We would like to appreciate all the faculties and senior brothers who helped us complete this on time. Thank you.

2022

Abstract

The Internet of things (IoT) is one of the most promising technologies on the rise in this modern era. It is expected to connect billions of devices(18 billion by 2022 as per Ericsson forecast) within the coming years. These resource constrained devices are expected to deal with a massive chunk of data which can sometimes be coined as “sensitive” in nature. IoT devices were designed to be generally cheap and easily replaceable. As a result, they are treated as expendable devices which is the core cause for their huge resource constraints.

They often have limitations relating power, memory, processing speed etc. as they are not expected to serve the network after a limited amount of time. The lifetime of the network is highly dependent on the power efficiency of the devices and longevity(power supply) of the applied nodes. The easiest and most power efficient mode of communication through IoT nodes is to send data over plaintext. But communication via plaintext is extremely vulnerable and susceptible to all sorts of attacks. Confidentiality of transmitted data is of utmost importance. To provide security to the data, cryptographic techniques are commonly used.

Complication is, traditional encryption methods are very power hungry and often require large processing ability. That’s why lightweight and comparatively more power efficient algorithms are being developed as alternatives. PRESENT, HUMMINGBIRD, RC5, Skipjack, HIGHT and SIT are some of the algorithms used for IoT devices as alternatives to traditional algorithms. A comparative study between these algorithms has been presented here that focuses mostly on power efficiency. Pros and cons along with possible improvements are suggested. This will help in selecting the proper algorithm for power sensitive applications.

Chapter 1

Introduction

The IoT (internet of things) can fairly be called “A New Technology” still as Kevin Ashton coined the term himself in 1999. It has been mentioned to be the future of technology in multiple instances. It is becoming a well-known discourse in the research field with the implementation practically. In the IoT network, generally it is a M2M interconnection without human intervention. The connections are made and maintained using some standard protocols for sharing information through public networks. Connecting everything with the internet is the reason for achieving Big Data. It can be described as “things” which are embedded with various sophisticated chips, sensors, technologies and software.

The major aspects of an IoT network are its ability to be inexpensive, handy, resource constrained and able to convey necessary data without human intervention. Now, all this data being generated can be very sensitive in nature and sometimes this data can be susceptible to attacks of different sorts from attackers commonly termed as hackers. The primary target of our dissertation is to be able to provide a choice between security solutions for people willing to adapt/apply an IoT network in any environment of choice.

We have gone through several techniques for doing the same job and compared their outputs to find the best balance between security and resource polling rates. Engineers are meant to find the perfect level of acceptable tradeoffs and set of features that's acceptably delivered to the user end. Here we tried to do just that. Most of the relevant recent lightweight encryption algorithms are taken and analyzed to find the best in the category of striking the balance that we are talking about.

While going through this comparative study, we have also tried to bring in blockchain technology to provide security to these IoT devices. But after a certain level of study and work,

we came to a conclusion that there are better and more practical ways to introduce encryption in this platform than block chain. So we dropped that idea and went with the existing modern lightweight algorithms to get a better outcome. For sure blockchain is much more secure but the resource consumption and application complexity just doesn't make practical sense when applied. Sometimes the seemingly better choices are practically unrealizable.

1.1 Basic Goal of Lightweight Encryption Algorithm

The operation of IoT devices is subject to a lot of cyber attacks as generally the data being transmitted via nodes placed in an IoT environment is vulnerable and can be sensitive in nature. Providing proper security to these data is of utmost importance and our primary goal is to do just that. To accomplish this task, there are a few ways that can be considered. Out of those methods, encryption of the data being transmitted is arguably the most secure and acceptable procedure that is being adopted.

There are a lot of algorithms for encryption during data transmission. But all the general algorithms are generally focused on security irrespective of it being energy and resource hungry. Problem is the nodes for IoT networks are often designed in a way where they are resource constrained by design as they have to be cheap, affordable and often expendable. The batteries used for these devices are small and cannot provide bulk power on demand. To introduce a layer of encryption in the transmission of data, we have to think of methods that are providing good security but don't cost too much power or memory.

Processing power is also kept in check. All these aspects are satisfied by Lightweight algorithms as they were designed keeping these things in mind. The primary goal of Lightweight encryption algorithm is to provide a certain level of acceptable security that might vary depending on application environment and type of data with a sufficient level of power and memory consumption while doing so. In this book, we have compared some of the top algorithms to see which fits best for low resource applications for IoT devices. This comparative study will help us choose the perfect algorithm catered towards the specific job being asked from them.

1.2 Different Aspects of IoT Data Security

The research delves into providing security for data being transferred from node to node within an established IoT network. There are three common layer architectures in IoT applications. They are the physical layer, communication layer and application layer. Physical layer, also known as Perception layer, is the layer which has sensors that sense and gather information about the surrounding environment. Communication layer transmits information using physical layer.

It is used as a media for wired, wireless, fiber optics, 2G, 3G, 4G, short range communication through public networks. The mode of application is identified by the application layer. It ensures the issues related to confidentiality, data integrity and authenticity which is a major component for maintaining security and privacy of IoT networks. Data can be tracked by hackers and the information can be leaked.

So end to end security is desirable. Confidentiality can be achieved through Encryption/Decryption. Additionally, data integrity is also important. The data cannot be changed on its intermediate states between source and destination. While traveling through mediums, data can be susceptible to attack by hackers. Data authentication is the next big thing.

The communicating entities must be capable of authenticating each other to ensure that the communication is held between the claimed entities. There are several algorithms like PRESENT, HUMMINGBIRD, RC5, Skipjack, HIGHT and SIT which help networks with data security via cryptography. Operation speed, power consumption and storage are some of the vital concerns. SIT is an algorithm which has a 64 bit clock cipher and needs a 64 bit key to encrypt the data.

For WSN where security is required, memory efficient cryptographic algorithms are needed. Where availability is needed, an energy efficient cryptographic algorithm is required. Another architecture for security is TinySec, which provides Skipjack and RC5 algorithms. Another algorithm is HIGHT for ubiquitous 8 bit computing devices. So according to the environment, users can choose the suitable algorithm for implementation.

1.3 Background and Motivation

The research presented in this dissertation aims at finding the most optimal way to provide security to data transmitted from node to node over an IoT network via encryption that is achieved through a resource efficient technique.

When we started looking for a method to establish a secure network using the IoT platform, we started working with blockchain and our goal was to develop an algorithm that will pose a good balance between power efficiency and data encryption for confidentiality. After a lot of time and effort, we had to come to a conclusion that it wouldn't be possible to produce an algorithm within the allocated time frame and resources we were exposed to.

Then we shifted our focus on finding the most efficient solutions that are already presented and we thought of comparing those for summarizing and focusing the data towards energy efficiency and security achievements of algorithms. This presentation is a brainchild of a desire to provide people with the best solution in terms of algorithm when designing an IoT network using resource constrained nodes. We tried to include the recent and popular algorithms only.

Chapter 2

Overview of algorithms

There are various types of lightweight cryptographic algorithm for IoT devices to ensure the security.

SIT Algorithm:

This is a hybrid encryption process that be formed of encryption rounds, where individual round is formed on some mathematical functions that create confusion as well as diffusion. Better security is made sure by increment in number of rounds. Though it increase the consumption of constrained energy.[2] The suggested algorithm is constricted to five rounds. To generate enough confusion and diffusion of data for oppose the attacks, the algorithm uses the feistel network of substitution diffusion function.

RC5 Algorithm:

This algorithm has a variable block size varying between 32, 64 and 128bits. It also allows for variable key size from 0 to 2040 bits. The no. of rounds can be fixed between 0 to 255. But when it was originally implemented, the block size was 64-bits with a 128-bit key and 12-rounds. This method is susceptible to differential attack if we implement 12-round RC5 with 64-bit bricks. To prevent that condition, 18 or more rounds are suggested.

Skipjack Algorithm:

It's an 80-bit key used for a 64-bit data block encryption or decryption. It utilizes unbalanced Feistel network with 32 rounds. This method is susceptible to attacks using impossible differential cryptanalysis. Exhaustive key search is also plausible as the length of key is relatively short.

HIGHT Algorithm:

It is a 64-bit block length and 128-bit of key length algorithm which is better for low cost, low power and ultra-light implementation. It is a faster and more efficient algorithm than AES.

This algorithm is susceptible to differential attack, Boomerang attack in case of 13-round application with 2^{62} plaintext. Saturation attack can be done on 16-round HIGHT effectively. Saturation attack is more effective than differential attack in most cases.

AES Algorithm:

AES (Advanced Encryption standard) is a good example of SPN based algorithm, which is standardized by NIST, performs on 128-bit block with 128, 192 and 256-bit key variants.

Hummingbird Algorithm:

It is an ultra-lightweight algorithm, introduces a hybrid structure (block and stream). It takes 16-bit input with a 256-bit key to perform 20 iterations.

PRINCE Algorithm:

It is both hardware and software efficient lightweight algorithm which performs on 64-bit input using a 128-bit key for 12 times.

Fantomas Algorithm:

It is a linear cryptanalysis of round reduced block cipher. It belongs to the family of bitslice ciphers. Fantomas is a technique that can be applied to both algorithms/linear cryptanalysis. The proposal for linear cryptanalysis is to construct a linear characteristic.

The relation between plain text and ciphertext bits can be described with it. The relationship can hold with probability 0.5 for a secure cipher.

2.1 Outcomes of the algorithms

SIT Algorithm

Table: 2.1 Correlation & entropy measurement of SIT

Image	Correlation Initial	Correlation Encrypted	Entropy Initial	Entropy Encrypted
Panda	0.9811	0.0003	7.4938	7.9968
Onion	0.9875	0.0020	7.3414	7.9971
football	0.9616	0.0023	6.6839	7.9973
baboon	0.8198	0.0041	7.2316	7.9973

Here we can see that, from the SIT algorithm, the correlation and entropy measurement can be determined after running the process. The process shows us the measurement of before and after encryption results. From the table we can see that before encryption initially the correlation for four things(i.e, Panda, Onion, football, baboon) are quite high.

This shows that the correlation of these things are quite good. Whereas after the encryption it is seen that the correlation is almost zero. Which shows that the correlation is almost null which results in very good encryption, signifying almost no similarity between the original and encrypted image resulting in high encryption.

Now for entropy, we can see that before encryption the entropy varies. But after encryption it is quite uniform for the four things which results in good encryption. The uniform distribution of intensities after the encryption is an indication of desired security. The entropy measurement can be done from

Entropy = Intensity x probability of intensity

So the measurements of correlation and entropy shows that the encryption is done well.

Some resulted figures:

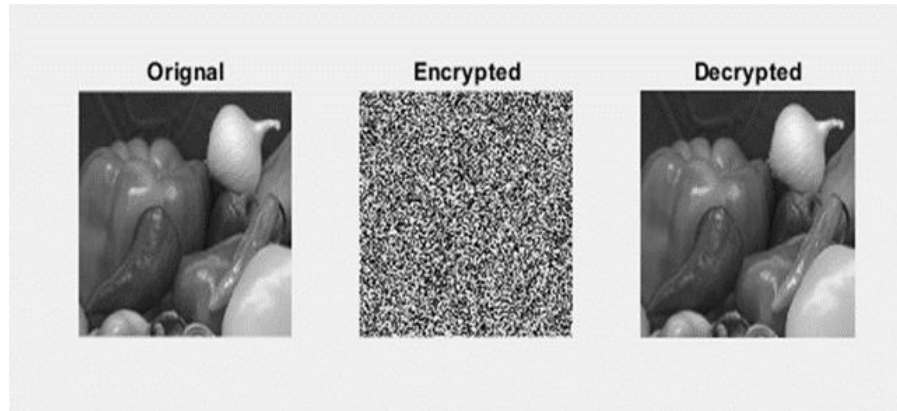


Figure 2.1: Encryption & decryption of onion

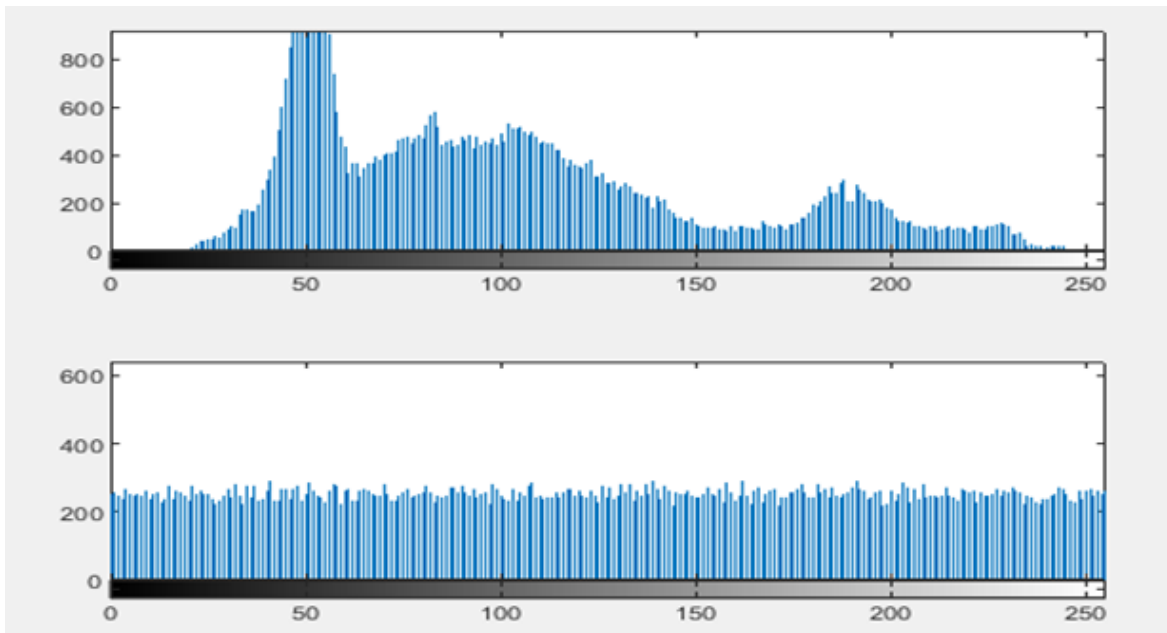


Figure: 2.2: Histogram of entropy for onion(before and after encryption)

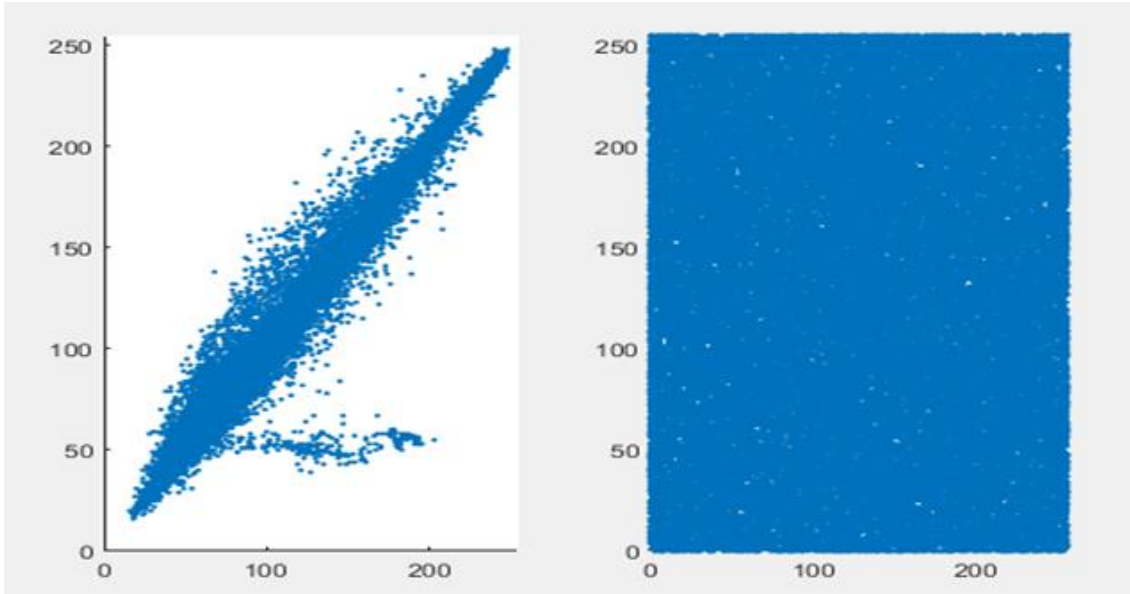


Figure 2.3: Correlation for onion(before and after encryption)

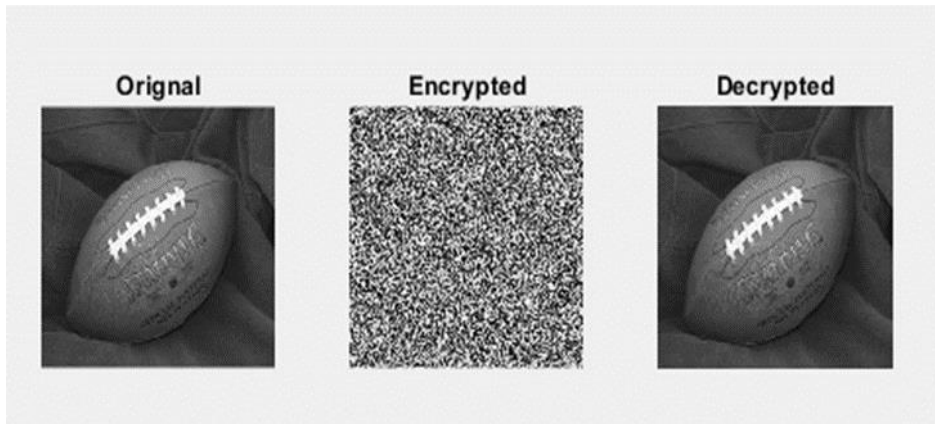


Figure 2.4: Encryption & decryption of football

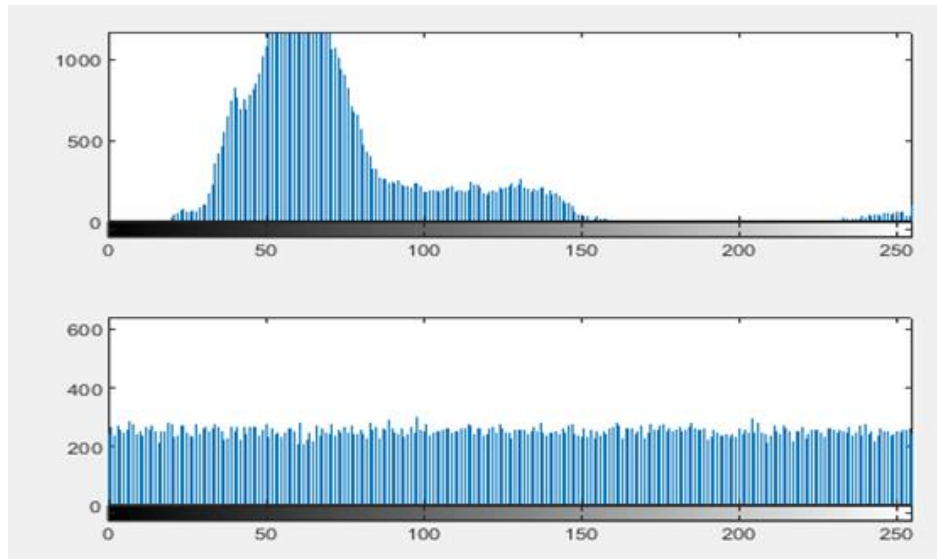


Figure: 2.5: Histogram of entropy for football(before and after encryption)

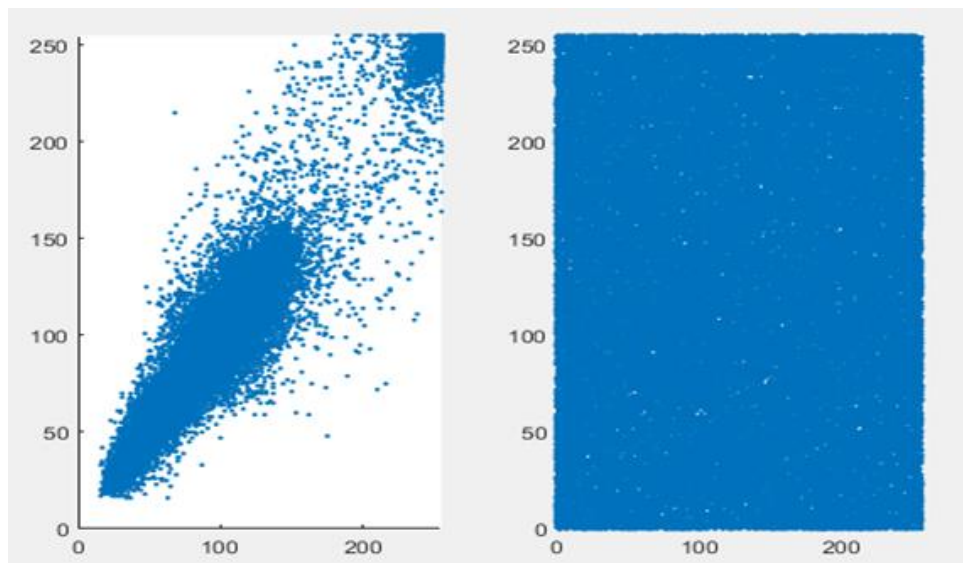


Figure: 2.6: Correlation for football(before and after encryption)

Figure: 2.1 is the encryption and decryption picture. After encryption the picture is unreadable. And after decryption the picture looks like the original one.

In figure:2.2 it is seen that after encryption the intensity is quite uniform which results in good encryption as uniform intensities are desired.

The figure: 2.3 shows us the correlation. As it is seen that before encryption the points are quite linear. Which shows that the correlation is good.

But after encryption the points are scattered, not maintaining a linear relationship. So here the correlation is almost zero and encryption is good.

Similar things can be explained from figure: 2.4, figure: 2.5 and figure: 2.6 as well.

Skipjack Algorithm:

```
d=3
Public key is (3,15)
Private key is (3,15)
Enter the message: IUT
ASCII equivalent of message
    73    85    84

The encrypted message is
    14   14   14
The decrypted mes in ASCII is
    73   85   84
The decrypted message is: IUT
```

Figure 2.7: Result for Skipjack algorithm

From the figure: 2.7 we can see the result for the Skipjack algorithm. In this figure we can see that there is public key and private key. Public key is for shared key whereas private key is hidden. As we can see if we enter any message(i.e, IUT) it will convert it to ASCII equivalent first.

Then it will encrypt the message. After encryption it will decrypt it and finally the decrypted message is shown(i.e, IUT) which is the same message as the expected message.

RC5 Algorithm

```
RC5 Encryption Algorithm Test Code  
Plaintext: 2215633647 1339455216  
CryptedText: 726107159 3043073298  
DecryptedText: 2215633647 1339455216  
***Decryption process is success.***
```

Figure 2.8: Result for RC5 algorithm

In the RC5 encryption algorithm, from the figure: 2.8, the result shows us the plan text first converted to encrypted text and after that it decrypted and we found exactly the same text as the plan text. So by this it can be said that the decryption process is done successfully.

Hummingbird Algorithm (23 Benchmark Functions)

```
FunIndex=1  
The best fitness is: 2.8154e-302
```

Figure 2.9: Result for Hummingbird algorithm (23 benchmark function)

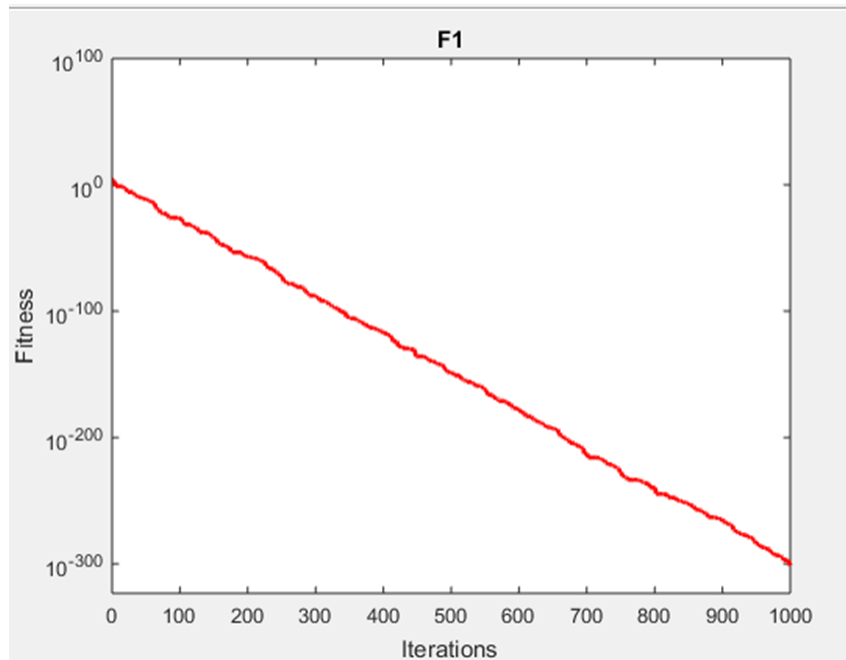


Figure 2.10: Fitness vs Iteration curve of Hummingbird algorithm (23 benchmark function)

In figure: 2.10 we can see the fitness vs iterations curve for Hummingbird algorithm for 23 benchmark functions. Here we can see that the the more the iterations the decrement of the fitness.

As IoT is an energy constrains device, for larger iterations its fitness decreases as we can see this in the figure. In figure: 2.9 we found the best fitness for this amount of benchmark function is $2.8154e-302$.

Hummingbird Algorithm (50 benchmark functions)

```
FunIndex=1
The best fitness is: -5
```

Figure 2.11: Result for Hummingbird algorithm (50 benchmark function)

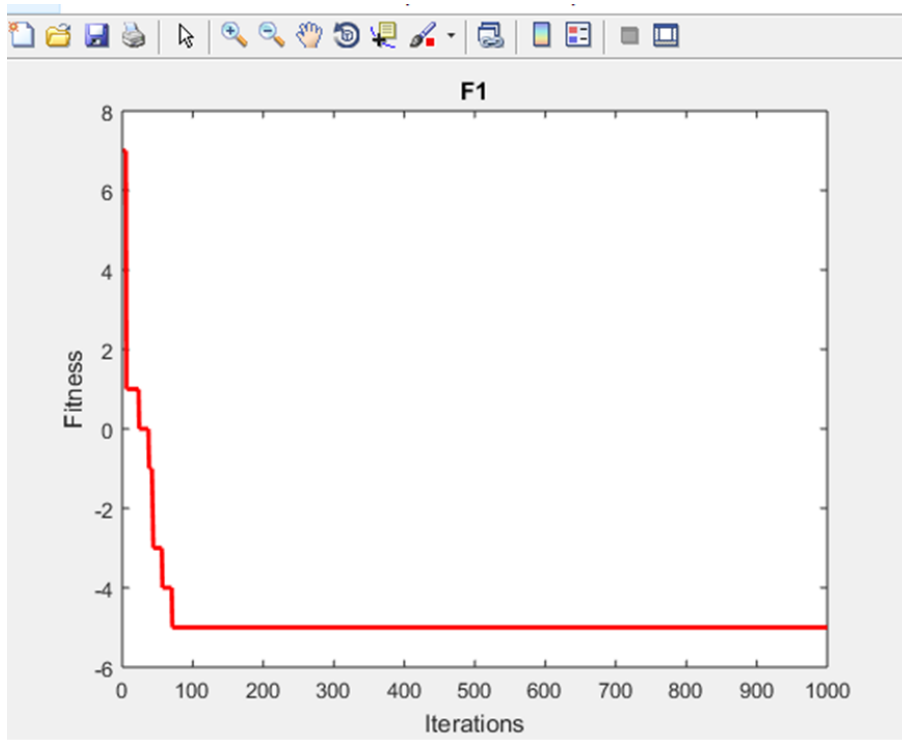


Figure 2.12: Fitness vs Iteration curve of Hummingbird algorithm (23 benchmark function)

This is also for Hummingbird algorithm but for 50 benchmark functions. The concept of fitness vs iterations is similar here also. Here in figure: 2.12 it shows the fitness vs iterations graph. In figure: 2.11, it shows the best fitness -5.

AES Algorithm

```
Initial state :      00 44 88 cc
                   11 55 99 dd
                   22 66 aa ee
                   33 77 bb ff
```

Figure 2.13: Initial state of AES algorithm

```

Round key :           00 04 08 0c
                    01 05 09 0d
                    02 06 0a 0e
                    03 07 0b 0f

```

Figure 2.14: Round key of AES algorithm

```

Final state :        00 44 88 cc
                    11 55 99 dd
                    22 66 aa ee
                    33 77 bb ff

```

Figure 2.15: Final state of AES algorithm

In the results of AES algorithm it is seen from the figure: 2.13 that it is the initial state. Which is encrypted and a round key(figure: 2.14) is found. And after decryption the final state (figure: 2.15) is found. Here, it is clearly visible that the final state is similar to the initial state. So the encryption and decryption is done perfectly here.

Table: 2.2 Comparison among SIT, Skipjack and RC5

	SIT	Skipjack	RC5
RAM	22	328	72
Cycles	3006	17390	70700
Code Size	826	5230	3288
Key size	64	80	128

Table: 2.2 shows us the comparison among SIT, Skipjack and RC5 algorithms for RAM, cycles, code size and key size. From the comparison it is seen that, RAM is less required for SIT, then for RC5 and then for Skipjack.

For cycles, SIT requires less cycles, then Skipjack and then RC5 which requires more cycles compared to the other ones. For code size, SIT algorithm needs lesser code size, then RC5 and the more code size is needed for Skipjack. The requirement for key size is lesser for SIT, then Skipjack and then RC5.

So compared to all the four categories, SIT algorithm is more efficient than other two algorithms.

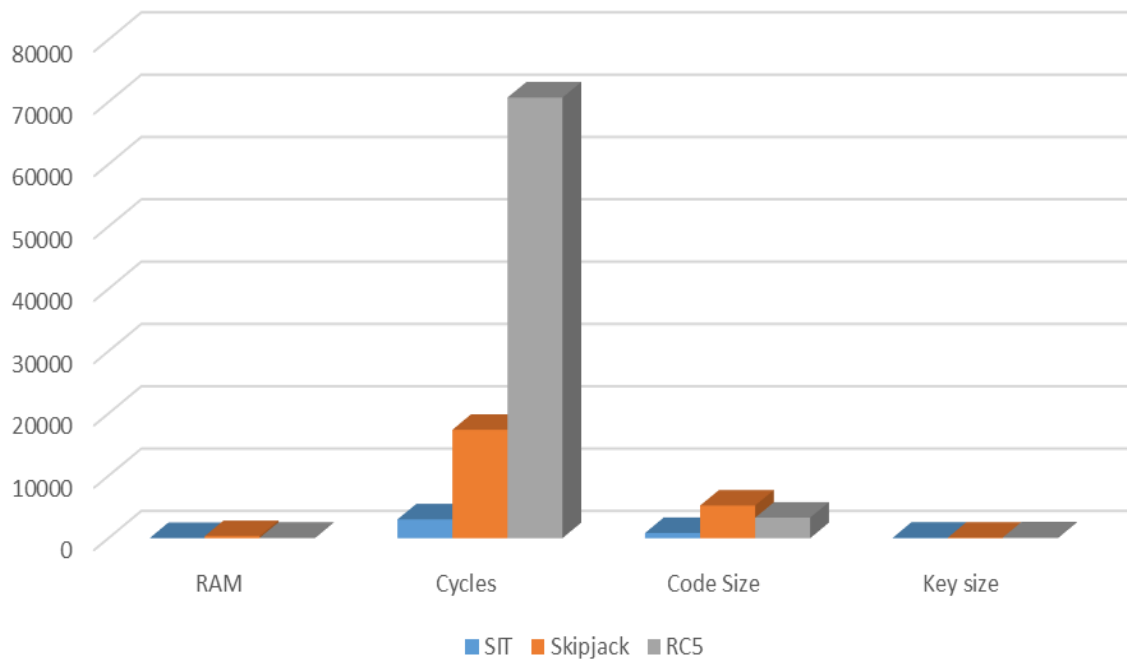


Figure 2.16: Comparison among SIT, Skipjack and RC5

In this figure: 2.16 it shows bar chart of the previous table. The y-axis shows the requirements. Here, it is visible that RAM, cycles, code size and key size is lesser for SIT algorithm. For RC5 the cycles are in high requirement. And other categories are varies in larger or lower requirements for the Skipjack and RC5 algorithms.

2.2 Performance Analysis

Memory efficiency:

Memory consists of ROM and RAM. Program implementation is done by RAM. Sensor node memory is limited and needs energy to store information. Higher efficiency is paramount.

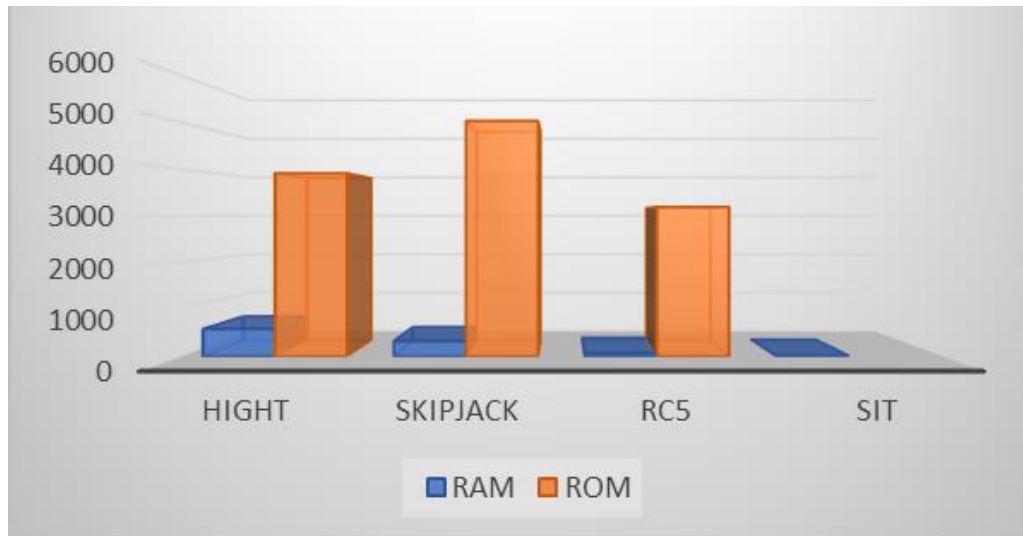


Figure 2.17: Comparison among HIGHT, Skipjack, RC5 & SIT

This figure: 2.17 talks about the memory needed by RC5, skipjack, HIGHT, SIT. It is seen that RAM and ROM requirement is lower for SIT algorithm. RAM requirement is highest for Skipjack algorithm whereas ROM requirement is highest for HIGHT algorithm among these four types of algorithms. So, SIT algorithm is more efficient here.

Operation Time

Average of estimated values were determined by consistently running encryption and decryption process. RC5, Skipjack, Hight uses simpler process like XOR, bitwise rotation, addition.

Table: 2.3 Comparison among HIGHT, Skipjack, RC5 and SIT in case of operation time

Algorithm	Operation Time
HIGHT	7.413 s
Skipjack	7.299 s
RC5	7.264 s
SIT	6.258 s

Table: 2.3 shows the comparison for four types of algorithms (HIGHT, Skipjack, RC5 and SIT) in terms of operation time. Here it shows that the highest operation time is needed for HIGHT algorithm whereas the lowest time is needed for SIT algorithm. So here SIT algorithm is more efficient compared to the rest of the three algorithms.

Energy Efficiency

Energy consumed per byte is calculated. CPU cycle was estimated by PowerTOSSIM. Power consumption of RC5, Skipjack, SIT and Hight is given in chart.

Table: 2.4 Comparison among HIGHT, Skipjack, RC5 and SIT in case of CPU cycles

Algorithms	CPU cycle
HIGHT	64,355
Skipjack	17,390
RC5	70,700
SIT	3,006

Table: 2.4 shows that HIGHT algorithm requires the highest CPU cycle, then RC5, then Skipjack and the lowest requirement is for SIT algorithm. As it is seen, the most efficient cryptographic process here is SIT algorithm.

Table: 2.5 Comparison among eight algorithms in case of performance analysis

	SIT	Skipjack	RC5	HIGHT	Hummin gbird	AES	Prince	Fantomas
RAM	22	328	72	584	82	-	-	78
Cycles	3006	17390	70700	64355	4637	4192	3614	3646
Code Size	826	5230	3288	5672	-	-	-	-
Key size	64	80	128	128	128	128	128	128
Block size	64	64	64	64	16	128	64	128
ROM	-	5020	3188	3906	1822	918	1108	1920

Table: 2.5 shows the comparison of performance analysis of eight algorithms (SIT, Skipjack,, RC5, HIGHT, Hummingbird, AES, PRINCE and Fantomas) in terms of RAM, cycles, code size, key size, block size and ROM.

Among them the highest requirement for RAM and Code size belongs to the HIGHT algorithm. RC5 requires the highest cycles requirement. Without SIT and Skipjack algorithms the rest of the algorithms require higher key size. AES and Fantomas algorithms require the highest block size. Skipjack requires the highest ROM among these algorithms. On the other hand SIT algorithm needs the least amount of requirements in case of RAM, cycles, code size, block size, key size. The ROM requirement for SIT was not found because it needs hardware implementation.

Overall it is seen that SIT algorithm is more efficient compared to the rest of the algorithms in case of performance analysis.

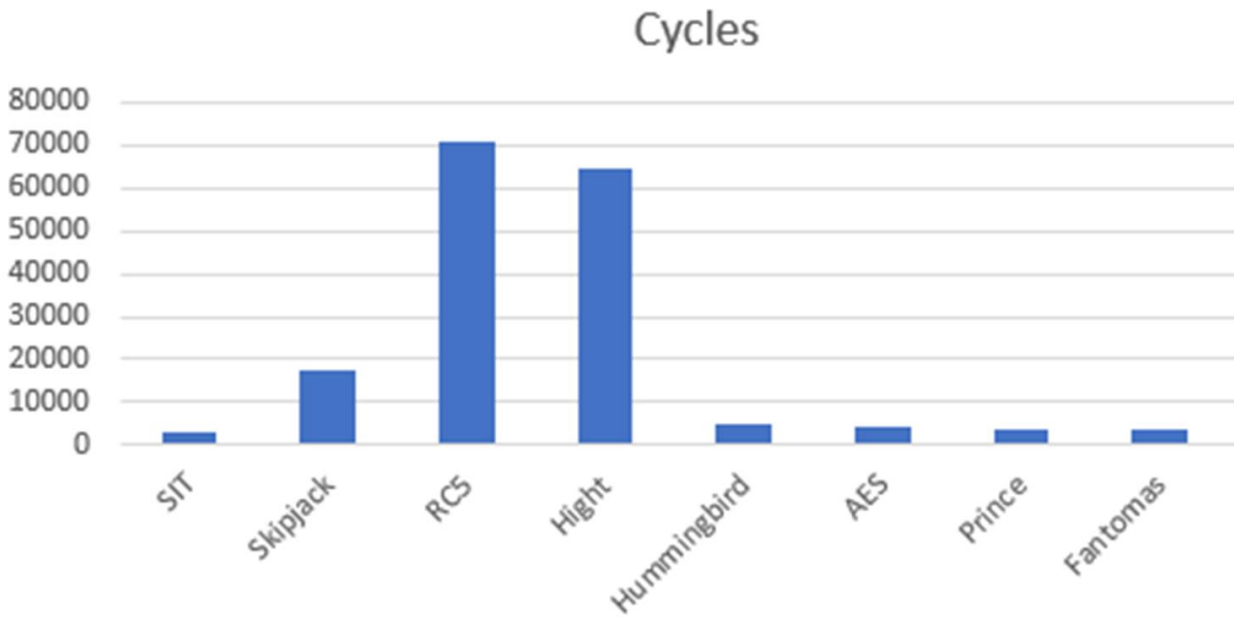


Figure 2.18: Comparison among eight algorithms in case of cycles

Figure: 2.18 shows the comparison of cycles requirements for the eight algorithms (SIT, Skipjack, RC5, HIGHT, Hummingbird, AES, PRINCE and Fantomas).

The y-axis of the barchart is the requirement values. This shows that the highest requirement for cycles is RC5 algorithm. And the lowest requirement is for SIT algorithm.

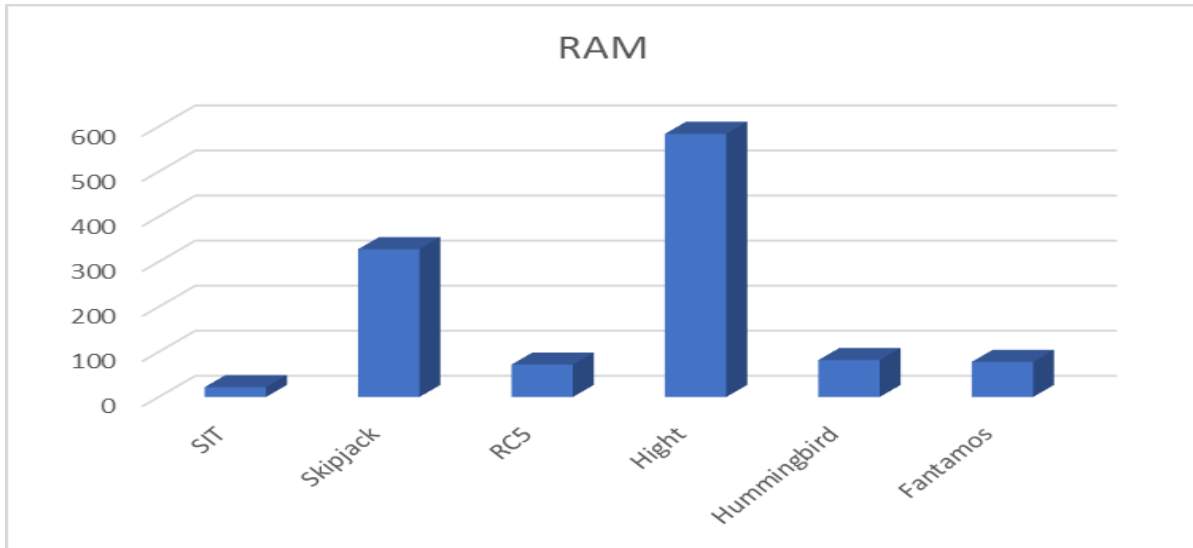


Figure 2.19: Comparison among six algorithms in case of RAM

According to the above figure: 2.19 it can be seen Hight algorithm consumes greater RAM compared to others whereas SIT Algorithm consumes least amount of RAM making it more efficient and resource friendly.

Table: 2.6 Comparison among Present, AES, CLEFIA, DES and Klein algorithms in case of RAM & ROM

	Present	AES	CLEFIA	DES	Klein
RAM	1384	2016	1256	4680	1256
ROM	3200	3716	4708	10628	2472

Table: 2.6 shows some more cryptographic algorithms. So comparing these five types of algorithms (Present, AES, CLEFIA, DES and Klein) it is cleared from the table that DES algorithms requires the highest RAM and ROM values. So, this algorithm is the least efficient among the five algorithms.

And the lowest requirement of RAM and ROM is for Klein algorithm. Though CLEFIA has the same requirement for RAM but it requires higher ROM than Klein. So after analysing the table it can be said that the most efficient one among these five categories of algorithms is Klein algorithm.

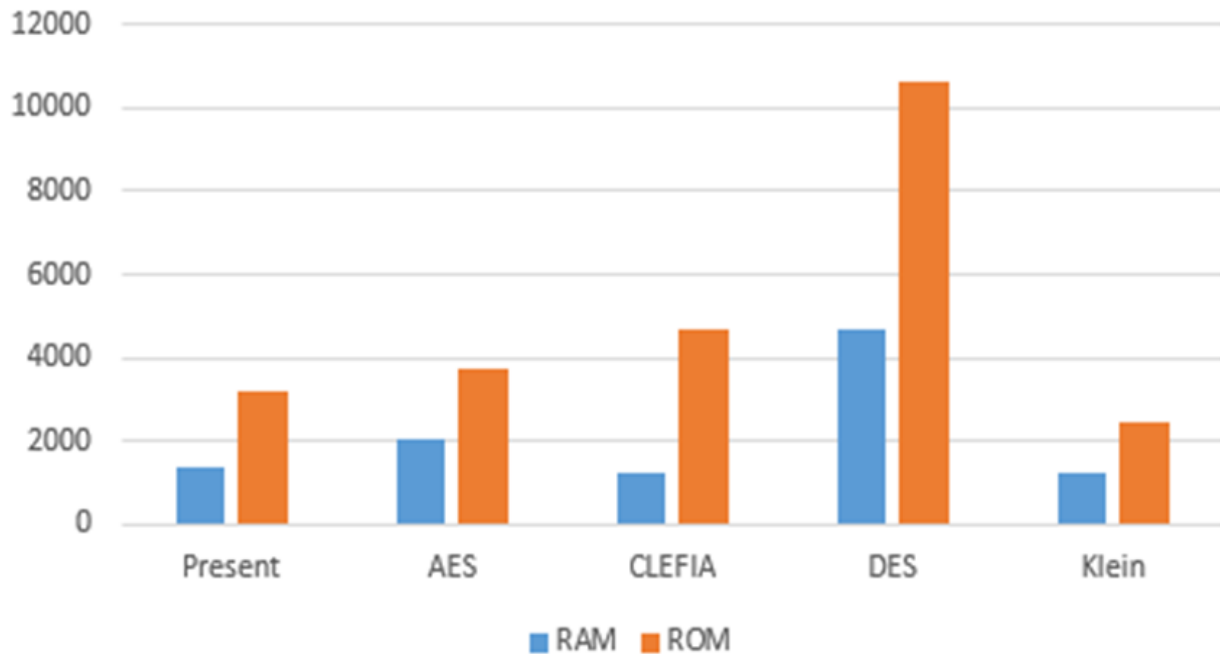


Figure 2.20: Comparison among five algorithms in case of RAM and ROM

Figure: 2.20 shows the bar chart of requirements of RAM and ROM for the five categories of algorithms (Present, AES, CLEFIA, DES and Klein). Y-axis shows the requirements quantity.

From the bar chart it is visible that the DES algorithm requires the highest amount of RAM and ROM whereas Klein algorithm requires the lowest amount of RAM and ROM compared to these five kind of algorithms (Present, AES, CLEFIA, DES and Klein).

2.3 Security Analysis of some LWC Algorithms

Present-GRP Algorithm:

Brute force is quite a popular kind of attack procedure. It is resource intensive and time consuming as it tries every possible key configuration possible. This attack can be dominated by increasing key size but with increasing key size, the complication related to calculations increases exponentially. This algorithm provides a good balance where it meets the safety margin expected from a lightweight algorithm while staying comparatively safe from brute force attacks.

GRP algorithm shows an avalanche effect if there's a small change in input which is desirable to resist differential kinds of attacks. Another most common attack is linear attack. Present-GRP Algorithm is immune to linear attack due presence of its S-Box. Algebraic attacks are unlikely to present a threat to PRESENT-GRP as the PRESENT algorithm is immune to such attacks.

RC5 Algorithm:

This Algorithm is vulnerable to differential attacks if implemented in 12-round RC5 with 64-bit bricks. To prevent that situation, 18 or more rounds are preferred.

Skipjack Algorithm:

It uses an 80-bit key used for a 64-bit data block encryption and decryption. It utilizes an unbalanced Feistel network with 32 rounds. This method is susceptible to attacks using impossible differential cryptanalysis. Exhaustive key search is also recommended as the length of key is relatively short over there.

HIGHT Algorithm:

This algorithm is also vulnerable to differential attack. Boomerang attack in case of 13-round application with 262 plaintext. Saturation attack can be done on 16-round HIGHT much effectively. Saturation attack is more effective than differential attack in majority of the cases.

SIT : Linear & Differential Cryptanalysis:

The F-function used here can withstand differential attacks due to it treating every bit in a similar manner in case of round transformation. The input-output correlation is kept very large to prevent it from linear attacks.

Weak keys: The proposed algorithm first XORs the key given and then feeds it to F-function. All non-linearity is constant in the F-function afterwards.

Related Keys: The proposed algorithm is designed to have fast and non-linear diffusion of cipher keys which prevents slow diffusion and symmetry based attacks.

Interpolation Attacks: This attacks are practically unreasonable for the proposed algorithm due to its diffusion layer and expression of the S-box.

SQUARE Attack:

For this attack to come to fruition, 2^8 key guesses is required which will produce 2^{16} S-Box lookups, making the attack practically irrelevant

Chapter 3

Challenges

Iot (internet of things) network possesses higher risks to various threats from cyber attacks causing hindrance in proper functionality of the system. Iot devices can be kept unsupervised for a long time increasing its risks for physical attacks. As all the communication occurs through wireless mediums so chances of eavesdropping are also high increasing its probability of cyber attacks. Components of IOT have lower capability in terms of energy and in terms of computational capability. Conventionally expensive computationally security algorithms will create a barrier on the performance of limited energy devices.

IOT composes of three components- Hardware, Middleware, Presentation. Middleware gives storage and computational elements. Middleware solutions are best for assisting a sensor node take decision on the most important data for processing.

In IOT the sensor nodes are taken as the internet nodes so the authentication process becomes more significant. Various cryptographic algorithms are there but their implementation in IOT is not feasible. Hence Secure IOT (SIT) comes here to help. It can deal with the security and resource allocation challenges for the network

Main challenge occurs during optimization of all three parameters and maintaining proper balance between them. Main components for optimization are cost, performance and security. If we increase the key size ultimately it degrades the algorithm performance. Our main target is not to compromise security hence more emphasis on less computing power , memory consumption, less physical area is ideal.

Creating random sub-keys from given keys for all rounds is a challenge. Decreasing rounds not hampering the overall security is challenging as well. Hence our goal is to create a lightweight cryptography algorithm with correctly balancing parameters like cost, performance and security.

Chapter 4

Conclusion

Very soon the Internet of Things will be an integral part of our daily lives. Plethora of energy constrained devices and sensors will continuously be communicating with each other, the security of which must not be compromised at all. To ensure this a lightweight security algorithm was proposed in this paper named as SIT. The implementation shows promising outcomes making the algorithm a suitable algorithm to be adopted in IoT applications.

As there is an exponential growth in the number of IoT devices in various fields, IoT security is one of the main concerns. Hence, there is a need for a lightweight algorithm with trade-offs amongst cost and performance and security.

For resource-constrained IoT devices, lightweight cryptography is a suitable way to secure communication by encrypting the data. The cost, performance and security are compared, and further research gaps were shown.

Moreover, new attacks are reported with the growth of new LWC algorithms which is an unavoidable and never-ending procedure. The war between cybersecurity experts and attackers always opens a window of new opportunities for modern research in the field of cybersecurity, mostly lightweight cryptography.

Since they are resource constraint devices so a lightweight cryptography algorithm is crucial which is capable to maintain good balance between cost, performance and security. Latency and energy consumption of IOT devices should be minimal as they are smaller in physical area as well.

We tried to compare some efficient cryptography methods through various parameters, these algorithms should always be improved with time in order to wipe out the possibility of any cyber-attacks.

4.1 Future Work

Implementing the algorithms on hardware and software in various computation and network environments are also under consideration. In addition, the algorithm can be optimized in order to enhance the performance according to requirements of different hardware platforms.

The scalability of algorithms can be exploited for stronger security and performance by changing the number of rounds or the architecture to support different key lengths.

Security and overall performance can be improved by changing its capacity to support different key lengths and become resistant to any kind of cyber or physical attacks.

Improving the balance between cost, performance and security will also be considerations for future research. An algorithm with the right blend of three main characteristics namely, cost, performance and security is ideal.

Chapter 5

References

- [1] Usman M, Ahmed I, Aslam MI, Khan S, Shah UA. SIT: a lightweight encryption algorithm for secure internet of things. arXiv preprint arXiv:1704.08688. 2017 Apr 27.
- [2] R. Chandramouli, S. Bapatla, K. Subbalakshmi, and R. Uma, "Battery power-aware encryption," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 2, pp. 162–180, 2006.
- [3] Usman M, Ahmed I, Aslam MI, Khan S, Shah UA. SIT: a lightweight encryption algorithm for secure internet of things. arXiv preprint arXiv:1704.08688. 2017 Apr 27.
- [4] Koo WK, Lee H, Kim YH, Lee DH. Implementation and analysis of new lightweight cryptographic algorithm suitable for wireless sensor networks. In 2008 International Conference on Information Security and Assurance (ISA 2008) 2008 Apr 24 (pp. 73-76). IEEE.
- [5] M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of Things (IoT): Research, Simulators, and Testbeds," *IEEE Internet of Things Journal*, pp. 1–1, 2017.
- [6] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography," National Institute of Standards and Technology, Gaithersburg, MD, NIST IR 8114, Mar. 2017.
- [7] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions," *J Ambient Intell Human Comput*, pp. 1–18, May 2017.
- [8] "Advanced Encryption Standard," [Online]. Available: <http://www.nist.gov/toolkits/aes/>.
- [9] "PRESENT-C," [Online]. Available: <https://github.com/bozhu/PRESENT-C>.
- [10] Beg A, Al-Kharobi T, Al-Nasser A. Performance Evaluation and Review of Lightweight Cryptography in an Internet-of-Things Environment. In 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS) 2019 May 1 (pp. 1-6). IEEE.
- [11] N. Pub, "197: Advanced encryption standard (AES)," *Federal Inf. Process. Standards*, vol. 197, no. 441, p. 0311, 2001.

[12] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: A very compact and a threshold implementation of AES," in *Proc. 30th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, May 2011, pp. 69_88.

[13] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith, "Hummingbird: Ultra-lightweight cryptography for resource-constrained devices," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2010, pp. 3_18.

[14] M. R. Albrecht, B. Driessen, E. B. Kavun, G. Leander, C. Paar, and T. Yalçın, "Block ciphers_focus on the linear layer (feat. pride)," in *Proc. Annu. Cryptol. Conf.* Cham, Switzerland: Springer, 2014, pp. 57_76.