LIGHT WEIGHT ENCRYPTION FOR PORTABLE DEVICES

A Project Report Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Sai Manoj Konidana

(Roll No. 2017BCS0030)

to

DEPARTEMENT OF COMPUTER SCIENCE

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

KOTTAYAM-686635, INDIA

April 2021

DECLARATION

I, **Sai Manoj Konidana** (**Roll No: 2017BCS0030**), hereby declare that, this report entitled

" "LIGHT WEIGHT ENCRYPTION FOR PORTABLE DEVICES

" submitted to Indian Institute of Information Technology Kottayam towards partial requirement of Bachelor of Technology in Computer Science and Engineering is an original work carried out by me under the supervision of Dr.Arun Cyril Jose and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold the academic ethics and honesty. Whenever an external information or statement or result is used then, that have been duly acknowledged and cited.

Kottayam-686635 **Sai Manoj Konidana**

April 2021

CERTIFICATE

This is to certify that the work contained in this project report entitled "LIGHT WEIGHT ENCRYPTION FOR PORTABLE DEVICES " submitted by Sai Manoj Konidana (Roll No: 2017BCS0030) to Indian Institute of Information Technology Kottayam towards partial requirement of Bachelor of Technology in IIIT kottayam has been carried out by him under my supervision and that it has not been submitted elsewhere for the award of any degree.

Kottayam-686635
April 2021

Dr.Arun Cyril Jose
Project Supervisor

ABSTRACT

With the increased usage of portable and resource constrained devices like IoT systems ,automotive systems and other wireless devices, the need for providing increased security to the data these devices operate on while managing the limited amount of resources like power(consumption), storage space etc. has become challenging.Thus a special set of algorithms called the "light weight algorithms" have been introduced to keep the resource overhead in check in terms of both hardware usage and ease of implementation.In this report we analyse several lightweight cryptographic algorithms for providing encryption on resource constrained devices and propose an encryption architecture that would be optimal in terms of both robust security and resource management.

Contents

Chapter 1

Introduction

Resource-constrained devices have many application areas like automotive systems, smart parking, sensor networks, weather forecast, healthcare, distributive control systems, IoT, cyber-physical systems, smart cities, smart grid, etc.

Devices in IoT are extremely open to assaults, as they remain unsupervised for long time there is a chance of physical attack on its components.Also it is simple to attack because of wireless communication medium. The constituents bear low competency in terms of energy and computational capability. If conventional security algorithms are used which require computations, their performance will be wasted. For example, IoT is used for monitoring purposes generates substantial amount of data, so their integrity and authentication are a matters of concerns.

1.1 LIGHTWEIGHT ALGORITHMS :

Algorithms used for encryption must be tailored for implementation in constrained environments including RFID tags, sensors, contact less smart cards, health-care devices etc while providing adequate security. From the view of the implementation properties, the lightweight algorithms are superior to conventional cryptographic ones :

1.In hardware implementations, chip size and energy consumption.

2.In software implementations, the smaller code and RAM size.

A considerable number of new light weight algorithms and modified versions of existing encryption algorithms have been introduced lately.

Some of them include :

1.AES (Advanced Encryption Standard)

2.DESL (A light weight implementation of the conventional Data Encryption Standard)

3.Hight algorithm

4.Grain algorithm

5.Trivium lightweight algorithm

6.Humming bird encryption algorithm

7.SIT algorithm.

Each of these algorithms emphasize on different aspects like data integrity, confidentiality ,authority at different levels of the system like the hardware, middle ware and the user interface.

Each of them have their own advantages like high level of confusion and diffusion in data post encryption, low hardware usage, self-encryption, ease of implementation etc. In the upcoming section we will analyse these various algorithms and their existing implementations and extract some critical points to looked over.

Chapter 2

LITERARY SURVEY

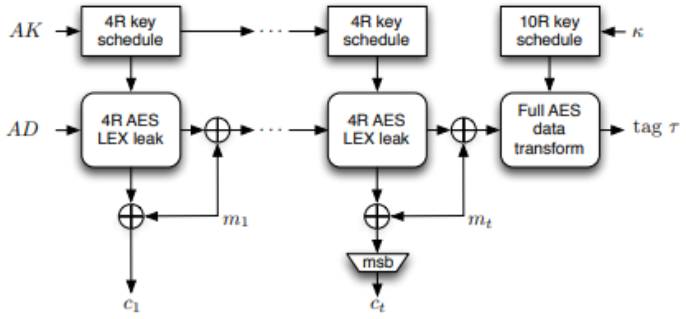
In this section, We are going to take an analysis on 6 different algorithms , their architectures and their cons.

2.1 AES Based Lightweight Authenticated Encryption : [6]

2.1.1 Description

This paper proposes a lightweight encryption algorithm based on AES called ALE (Authenticated Lightweight Encryption) being efficient in hardware and software aspects. It is a single-pass algorithm that uses a nonce and preserves the memory alignment of data.The architecture of ALE brings together some conventional ideas of Pelican, LEX and ASC-1 as a resource constrained algorithm. This algorithm uses Pelican keys in all rounds for forming

the authentication tag and transfers blocks of bits of the state in each round in a LEX way for encryption and decryption. It has a 256-bit internal state hidden internally that is formed from both key and an arbitrary numerical parameter.



2.1.2 Advantages :

It consumes only a maximum of 3 kge of area in the ASIC hardware, which is less than 100 ge overhead when compared to the conventional AES in it's smallest implementation possible. ALE consumes only half the size of the conventional AES and ASC-1.

In the aspect of speed of implementation for averaged sized inputs, ALE is two and a half times faster than AES and four and a half times faster than ASC-1 in it's smallest implementation.

2.1.3 cons :

1.It uses too simple algebraic structure.

2.Every block is always encrypted in the same way.

3.Hard to implement with software.

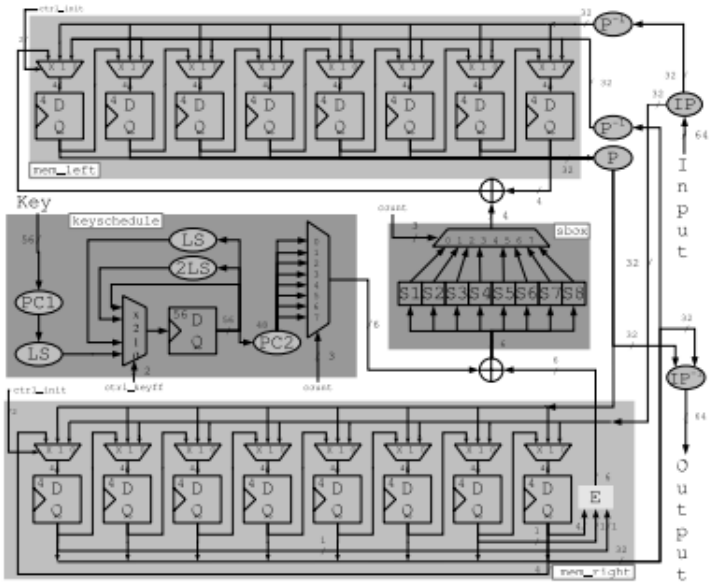
2.2 Light-Weight Cryptography Algorithm for RFID : [7]

DESL is a lightweight variant for the conventional DES algorithm where the difference lies in the F function. We substitute the conventional DES s-boxes with only one but stronger s-box, that is repeated eight times. And we remove the initial permutation and the inverse permutation, because they do not provide any additional cryptographic robustness, but at the same time requires area for hardware wiring.

The design of our DESL algorithm is exactly the same as for the DES algorithm, except for initial permutation and the inverse permutation wiring and the s-box module. The new s-box module hasonly one S-box. This S-box function neither requires the count control signal or a multiplexor, for saving resources by not using around 192 transistors.

2.2.1 pros :

1.It takes 144 clock cycles to encrypt one 64-bit block of plaintext. For one encryption at 100 kHz the average power consumption is 0.89 µA and the throughput reaches 5.55 KB/s.



2.2.2 cons :

1.It is broken easily using brute-force search.

2.3 Implementation of 'HIGHT' Encryption Algorithm on Microcontroller [8]

Hight algorithm is implemented for encrypting data in devices that have low power consumption, low cost etc.This algorithm contains a 128 bit key to encrypt 64 bit block of data.entire operations involved in this encryption are 8-bit oriented for 32 rounded feistel structure.

There are 4 primary steps involved in Hight algorithm :

1.Key schedule

2.Initial transform

3.Round function

4.Final transformation

2.3.1 pros :

1.Both encryption and decryption processes have similar implementation with just the order of blocks inverted.

2.It is more hardware oriented algorithm than a software one , hence making itself more applicable wireless resource constrained devices like RFID tags etc.

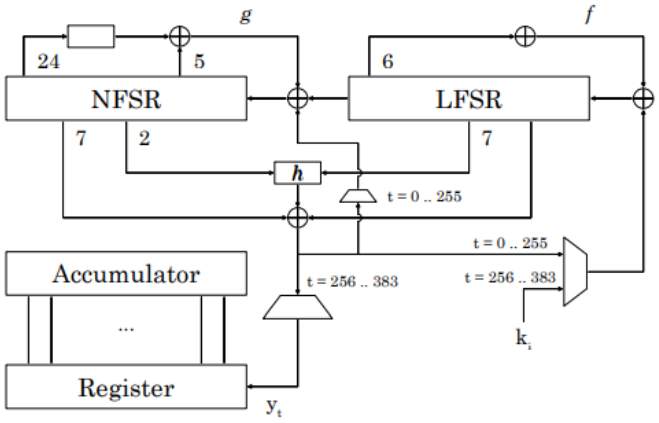
2.3.2 cons :

Slowness of encryption: an entire block must be accumulated before encryption or decryption can begin.

2.4 Grain-128 - A lightweight stream cipher [11]

Grain is a stream cipher algorithm built such that the hardware implementation is much more simple and requires relatively less chip area.

Any stream cipher consists of 2 phases :



Phase 1: Initialization of the internal state using a secret key and plain text. After that state keeps updating iteratively and key bits are generated.

The main components of the stream cipher are 2 eighty shift registers.Out of them one has a linear feedback and the other a non-linear feedback . The key size is specified with 80 bits and an initial value of 64 bits is being input.

2.4.1 Cons :

As it takes less chip size, it Is prone to cascading failures.

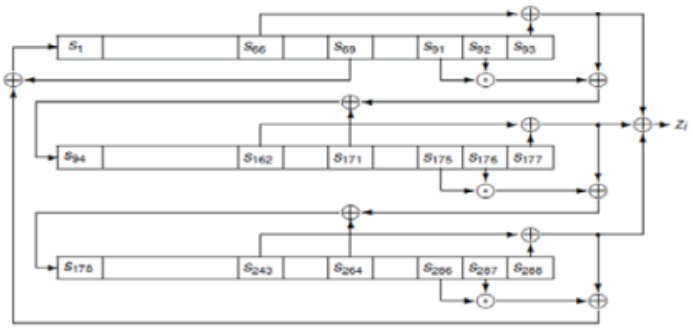
2.5 IoT Security: Performance Evaluation of Grain and Trivium - Lightweight Stream Ciphers [9]

2.5.1 Grain cipher :

The Grain cipher is a stream cipher algorithm that uses a fixed 80-bit key and 64-bit input value. It's key stream is generated by a combination of a linear feedback shift register and a non-linear feedback shift register and 3 primitive polynomial functions that determine the feedback and output bits. The system initially loads the plain text and key into the feedback registers respectively, and then starts to generate usable key stream after 160 rounds of iterating the system which includes feeding back the output back into the registers. It can generate a maximum of pow(2,80) bits of unique key.

2.5.2 Trivium stream cipher :

The Trivium cipher uses a 80 bit key and 80 bit input. The key is generated using a system that contains a 288-bit register, where particular registers are read and fed back into the system while it is cycled in a circular pattern. The key is loaded in the first 93 bits and the input is added in the next 84 bits. The entire system is cycled 1144 times before key is available. The system can produce up to pow(2,64) bits of unique key.



2.5.3 cons :

1.The main disadvantage is the high memory consumption for all the 288 states

2.The running time is inherently exponential in the number of variables.

2.6 The Hummingbird-2 Lightweight Authenticated Encryption Algorithm [5]

Humming bird-2 does not come under the category of block cipher or stream cipher, but is having the properties of both. the block size of Humming bird-2 is 16 bit, which is suitable for RFID devices or wireless sensors because it handles only small messages.

Humming bird-2 optionally produces an authentication tag for each message. The key size of The Humming bird-2 is 128 bit. Its internal state R, with size 128 bit, is initialized using 64 bit Initialization Vector IV. Accessing of these variables is done as vectors of

16 bit words. The operations in Humming bird-2 are exclusive OR, addition modulo 65536 and non-linear mixing function f(x) which are performed on 16 bit words.

2.6.1 Cons :

This algorithm is well suited for use in passive RFID systems due to its low power consumption, which minimally impacts range but not in active devices where the overhead becomes too high.

After observing all the cons of the algorithms mentioned above, we propose the following architecture in the upcoming section.

Chapter 3

Proposed Architecture :

It is a variation of Feistel and SP architectures, using their properties to make the best of both to provide desirable security while managing the resources and keeping computation complexities as minimal as possible.

The algorithm has two parts:

1.Key expansion (Scheduling)

2.Encryption

A 64 bit key is input by user, divided into 4 blocks, supplied into F-functions, arranged in 4X4 matrices and five unique keys that are generated from the expansion process using linear and non-linear transformations are input into XNOR operations for 5 continuous rounds of encryption.

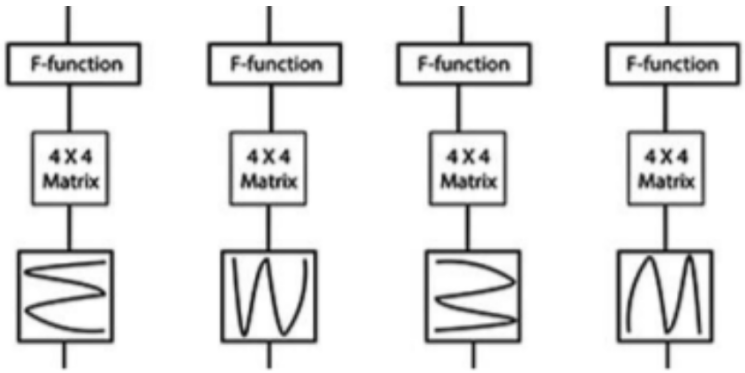


Figure 3.1: [1]

3.1 Key Expansion :

A 64-bit key is input from user or can be extracted from the stream of input data itself (Self-Encryption).

64 bit key is split into 4 16 bit parts and each of them are undergone into an initial permutation operation. Then each 16 bit sub key is subjected to a non-linear bit shuffling operation where each of their bits are shifted and shuffled in accordance with two permutation blocks.

After performing bit shuffling for 3 layers, the intermediate state is subjected to a final permutation operation. Now each of these 16 bit keys serve as 4 different keys for the first 4 rounds of the later encryption process. The 5th and final key is obtained by performing an XOR operation of the 4 initial keys.

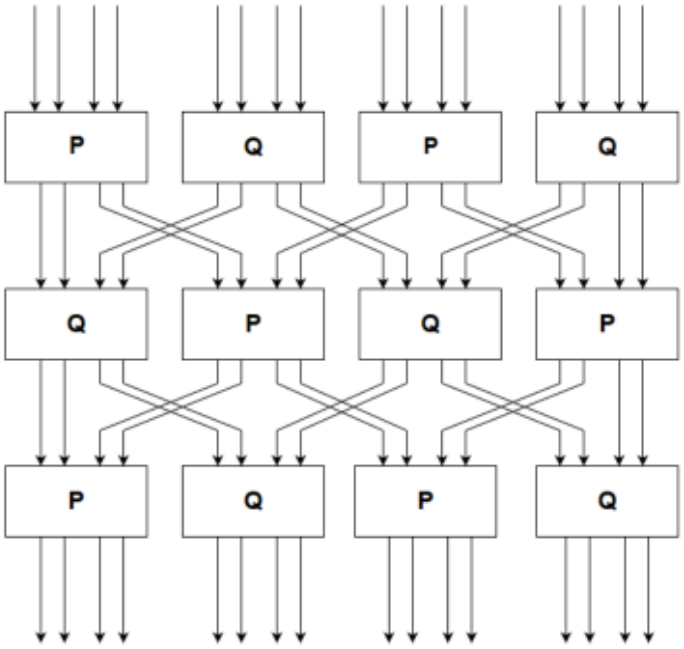


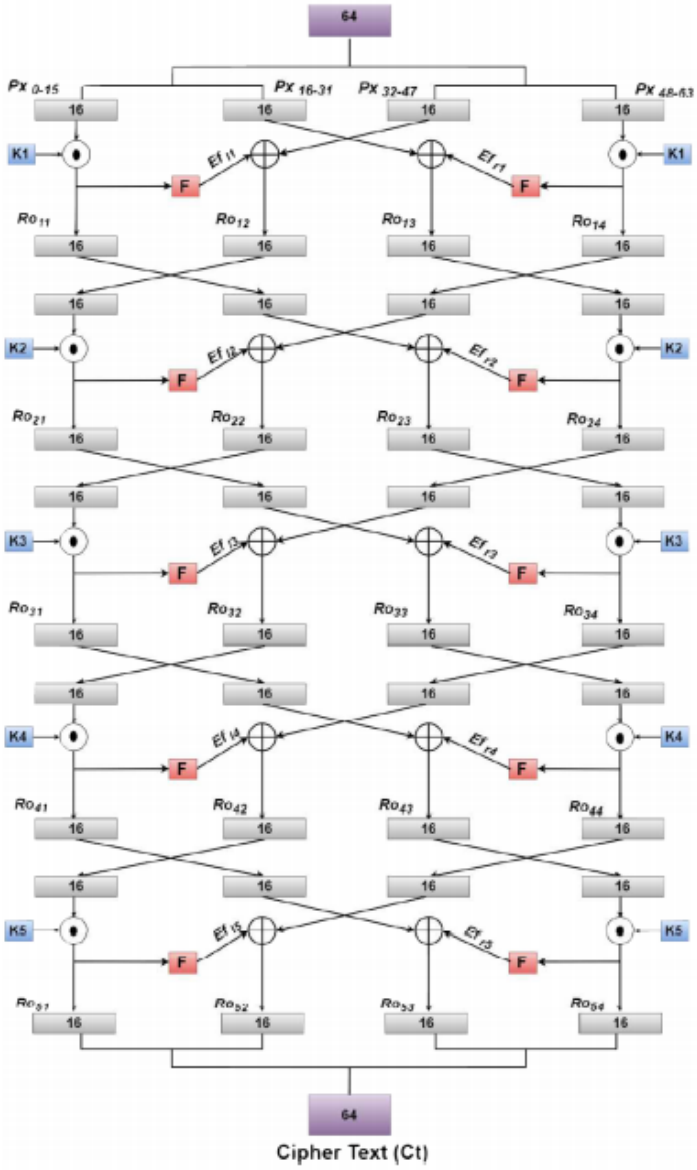
Figure 3.2: [4]

3.2 Encryption :

Again 64 bit input is divided into 4 16 bit parts and each of them undergo initial permutation. Post that, each of the 16 bit input parts are XNORed with the corresponding round key and shifted and shuffled for each round of encryption.

After 5 successful round of encryption , we finally obtain a robust,secure cipher text.

To create more complex cipher with higher confusion and diffusion, operations like crossover between the 16 bit parts and mutation (inherited from genetic algorithm) can be used.


64
Px 0-15
Px 16-31
Px 32-47
Px 48-63
16
K1
K2
K3
K4
K5
F
Ro11
Ro12
Ro13
Ro14
Ro21
Ro22
Ro23
Ro24
Ro31
Ro32
Ro33
Ro34
Ro41
Ro42
Ro43
Ro44
Ro51
Ro52
Ro53
Ro54
64
Cipher Text (Ct)

Chapter 4

RESULTS AND DISCUSSION

4.1 EVALUATING THE CIPHER

Key Sensitivity :

Key sensitivity ensures that the cipher must not decipher to original data if the key has even a bit difference from the actual key. The amount of change occurred in the cipher text by the change of one bit of the key is evaluated by Avalanche test. According to Strict Avalanche Criterion, the test is to be perfect if 50

4.2 EXECUTION CYCLE

Most fundamental parameter for the evaluation of algorithm performance is the amount of cycle to perform encoding and decoding a particular data. The proposed algorithm developed for resource-constraint devices in mind must consume minimal cycle and should

offer desired security. Execution cycle and power consumption can be correlated, in which case minimizing the cycle also tends to reduce the power consumption.

4.3 MEMORY UTILIZATION

Limitation of memory is one of the major challenges for resource-constraint devices. Memory can be measured the number of registers and the number of bytes of RAM and ROM that are used. ROM is used to store the program code and fixed data such as S-boxes and hardcore round keys, while RAM is used to store the computational values. The proposed algorithm uses small amount of rounds that suitable and favorable for its deployments in resource-constraint devices.

Chapter 5

CONCLUSION

Portable devices have become a basic component of our day by day lives. Various vitality obliged gadgets and sensors will persistently be speaking with one another the security of which must not be undermined. For this reason, above lightweight security calculation is proposed.

Chapter 6

References

[1] SIT: A Lightweight Encryption Algorithm for Secure Internet of Things Muhammad Usman , Irfan Ahmed† , M. Imran Aslam† , Shujaat Khan and Usman Ali Shah† Faculty of Engineering Science and Technology (FEST), Iqra University, Defence View, Karachi-75500, Pakistan.

[2] Saddam Hossain2 , Hasan Imam Shoun3 , Dr. Mohammod Abul Kashem4 Department of CSE Dhaka University of Engineering Technology, Gazipur Gazipur, Bangladesh.

[3] Cryptography for Resource Constrained devices: A Survey Jacob John Dept. of Computer Engineering Sinhgad Institute of Technology Pune, India.

[4] ENHANCED SIT ALGORITHM FOR EMBEDDED SYSTEMS Hemala N1, Satheesh T2 1PG Scholar Nandha Engineering College, Erode, Tamil Nadu, India 2Professor, Dept. of EEE, Nandha Engineering College, Erode, Tamil Nadu, India.

[5] The Hummingbird-2 Lightweight Authenticated Encryption Algorithm Daniel Engels, Markku-Juhani O. Saarinen, Peter Schweitzer, and Eric M. Smith REVERE SECURITY 4500 Westgrove Drive, Suite 335, Addison, TX 75001, USA

[6] ALE: AES-Based Lightweight Authenticated EncryptionAndrey Bogdanov, Florian Mendel, Francesco Regazzoni, Vincent Rijmen and Elmar Tischhauser, Technical University of Denmark.

[7] A. Poschmann, G. Leander, K. Schramm and C. Paar, "New Light-Weight Crypto Algorithms for RFID," 2007 IEEE International Symposium on Circuits and Systems, New Orleans, LA, USA, 2007, pp. 1843-1846, doi: 10.1109/ISCAS.2007.378273.

[8] J. Aguilar, S. Sierra and E. Jacinto, "Implementation of 'HIGHT' encryption algorithm on microcontroller," 2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Santiago, Chile, 2015, pp. 937-942, doi: 10.1109/Chilecon.2015.7404685.

[9] IoT Security: Performance Evaluation of Grain, MICKEY, and Trivium - Lightweight Stream Ciphers Levent Ertaul, Arnold Woodall, CSU East Bay, Hayward, CA, USA.

[10] The MAC function Pelican 2.0 Joan Daemen1 and Vincent Rijmen, STMicroelectronics Belgium, joan.daemen@st.com, KU Leuven iMinds (Belgium).

[11] Grain-128AEAD - A lightweight AEAD streamcipher, Martin Hell, Lund University, Sweden.,Thomas Johansson, Lund University, Sweden Willi Meier, FHNW, Switzerland.,Jonathan S¨onnerup,

Lund University, Sweden Hirotaka Yoshida, AIST, Japan.