جـــامـعـة الـــشـرق الأوسـط
MIDDLE EAST UNIVERSITY
Amman - Jordan

Implementation of proposed lightweight cryptosystem for use in Cloud Computing Security

تنفيذ نظام التشفير خفيف الوزن المقترح للاستخدام في أمن الحوسبة السحابية

Prepared By

Lubna Mutasem Al-Ramini

Supervisor

Prof. Hamza A. Al-Sewadi

A Thesis Submitted in Partial Fulfillment of the Requirements for the Master Degree in Computer Science

Department of Computer Science

Faculty of Information Technology

Middle East University

June 2018

Authorization

I, **Lubna Mutasem Mohammad AL-Ramini**, authorize Middle East University (MEU) to provide hard copies or soft copies of my thesis to libraries, establishments, and institutions upon request.

Name: Lubna Mutasem AL-Ramini

Date: 13-6-2018

Signature: 

Examination Committee's Decision

Examination Committee Decision

*This is to certify that the thesis entitled "**Implementation of Proposed Light Weight Cryptosystem for Use in Cloud Computing Security**" was successfully defended and approved on 13-06-2018.*

Examination Committee Members	***Signature***

Prof. Hamza Abbas. Al-Sewadi (Supervisor and Chairman)
Professor, Department of Computer Science
Middle East University (MEU)

Hamza A Al
13-6-2018

Dr. Mohammed A. Fadhil Al-Husainy (Internal Committee Member)
Associate Professor, Department of Computer Science
Middle East University (MEU)

٢٠١٨/٦/١٣

Dr. Mohammad A. Shkoukani (External Committee Member)
Associate Professor, Department of Computer Science
Applied Science Private University (ASU)

٢٠١٨/٦/١٣

Acknowledgment

I want to thank ALLAH for His blessings and facilitate all necessary to accomplish my work in full.

I am pleased to extend my thanks and gratitude to my supervisor **Prof. Hamza A. Al-Sewadi** for his efforts in this thesis through his valuable guidance and follow-up to me. Without him I would not have finished this thesis.

I must also thank all relatives and friends for their moral support during my academic journey.

Finally, my deep gratitude to my parents, for their patience and encouraging throughout the interval of studying and researching and to achieve my dream

Dedication

To my only sister, my friend, Life-long companion, my daughter I did not have, **Maysaa Mutasem AL-Ramini**.

I dedicate my effort

Table of Contents

List of Tables

List of Figures

List of Symbols

Symbol	Description
ADD	Logical Addition
CST	Constant
CT	Cipher-Text
F0, F1	Rotate function
K	Secret Key
N	Number of rounds
p	Sub-block
PT	Plain-Text
SK	Sub-key
SUB	Logical Subtraction
WK	Work-key
XOR	Exclusive or

List of Abbreviations

Abbreviations	Meaning
AES	Advanced Encryption Standard
API	Application Programming Interface
CC	Cloud Computing
CPU	Control processing unit
CSA	Cloud Security Alliance
DES	Data Encryption Standard
ECC	Elliptic Curve Cryptography
IDEA	International Data Encryption Algorithm
IT	Information Technology
LED	Light Encryption Device
LW	Light Weight
PES	Proposed Encryption Standard
PII	Personally Identifiable Information
RFID	Radio-frequency identification
SHSED	Simple and Highly Secure Encryption-Decryption
SPN	substitution–permutation network

Implementation of proposed lightweight cryptosystem for use in Cloud Computing Security

Prepared By

Lubna Mutasem Al-Ramini

Supervisor

Prof. Hamza A. Al-Sewadi

Abstract

In the course of the past 30 years, data has become pivotal to all aspects of human life. Data generated, captured, and replicated are increasing in size and expanding applications. The proliferation of fast wireless networks has encouraged data storage within the cloud. So, protecting data from attackers has become urgent to maintain its security and confidentiality, need for security and privacy technologies, systems, and processes to address it.

This thesis proposes a simple and highly secure encryption decryption (SHSED) algorithm, that can be used for cloud computing based applications. It is inspired by the international data encryption algorithm (IDEA) that is developed by Lai and Massey, (1991). It achieves the Shannon's concept of diffusion and confusion by the involvement of logical operations, such as XORing, addition, and subtraction in addition to byte shifting. It is also characterized by the flexibility in the secret key length and the number of rounds. Experimental results have demonstrated powerful security level and a clear improvement in the encryption execution time measurements and security strength as compared with cryptosystems widely used in cloud computing.

Key Words: Cryptography, cloud computing security, and lightweight algorithm.

تنفيذ نظام التشفير خفيف الوزن المقترح للاستخدام في أمن الحوسبة السحابية

إعداد

لبنى معتصم الراميني

إشراف

الأستاذ الدكتور حمزة السوادي

المُلخص

على مدار الثلاثين عاماً الماضية، أصبحت البيانات بالغة الأهمية بالنسبة لجميع جوانب الحياة البشرية، فالبيانات التي يتم توليدها وتسجيلها وتكرارها تتزايد ويتسع مجال التطبيقات التي تتناولها. وقد أدى انتشار الشبكات اللاسلكية السريعة إلى تشجيع تخزين البيانات داخل السحابة، لذا أصبحت حماية البيانات من المهاجمين أمراً ملحاً للحفاظ على أمنها وسريتها، والحاجة إلى تقنيات وأنظمة أمن وخصوصية لمعالجة هذه المشكلة.

تقترح هذه الأطروحة خوارزمية ذات تقنية بسيطة وعالية الامان وخفيفة الوزن للتشفير وفك التشفير (يطلق عليها SHSED) ويمكن استخدامها للتطبيقات القائمة على الحوسبة السحابية، وهي مستوحاة من خوارزمية تشفير البيانات الدولية (IDEA) التي طورت من قبل لاي وماسيه (Lai and Massey, 1991).

وتحقق هذه الخوارزمية مبدأ شانون للانتشار والارباك من خلال تنفيذ عدد من العمليات المنطقية كالجمع والطرح وبوابان XOR اضافةً إلى عمليات تزحيف البايتات. كما تمتاز الخوارزمية بالمرونة في طول المفتاح السري وعدد الدورات. وقد أظهرت النتائج التجريبية للخوارزمية المقترحة مستوى أمان قوي وتحسناً واضحاً في قياسات وقت تنفيذ التشفير وقوة الأمان مقارنة مع نظم التشفير المستخدمة على نطاق واسع في الحوسبة السحابية.

الكلمات المفتاحية: علم التشفير، أمن الحوسبة السحابية، خوارزمية التشفير خفيفة الوزن.

Chapter one

Introduction

1.1 Introduction

Cloud computing is a way to increment the capacity or add capabilities dynamically without investing in novel infrastructure, manage experiment personnel, or certify new software. It extends Information Technology's (IT) traditional capabilities. In the last few years, cloud computing has full-grown from being a promising business concept to one of the steadfast ontogenesis circular segment of the IT laboriousness. But as more and more teaching on individuals and corporation are stead in the cloud, concerns are beginning to grow about equitable how safe an environment is? Despite of all the hype enclosure, the cloud entertain customers are still reluctant to deploy their employment in the cloud. Security is one of the major delivery which shorten the growth of cloud number and complications with data privacy and data safety continue to plague the market. The advent of an advanced model should not scheme with the required functionalities and capabilities present in the current model. A new model targeting at improving features of an existing model must not risk or threaten other important features of the current model. The architecture of cloud poses such a threat to the security of the existing technologies when deployed in a cloud environment **(Priya jaiswal, Randeep kaur, Ashok Verma,2014).**

As cloud computing is achieving increased popularity, concerns are being voiced about the security issues introduced through the adoption of this new model. The effectiveness and efficiency of traditional protection mechanisms are being reconsidered, as the characteristics of this innovative deployment model, differ widely from those of traditional architectures **(Randeep Kaur and Supriya Kinger, 2014).**

In this research work, the researcher attempts to demystify the unique security challenges introduced in a cloud environment and clarify issues from a security perspective.

The new light weight cryptosystem will be referred to as “Simple and Highly Secure Encryption_Decryption algorithm (SHSED)It is inspired by international data encryption algorithm (IDEA) **(Xuejia Lai and James L. Massey, 1991)**. This proposed cryptosystem is anticipated to provide speed in execution time and powerful level of data security. Moreover, comparison will be conducted with other reported hybrid schemes.

1.2 Cryptography

Encryption has a special importance in information security science, which is the heart of information security because of its confidentiality. The use of encryption - through history – was to share messages that cannot be read by anyone other than the person intended to receive the message. Digital encryption technology has expanded beyond simple confidential messages; encryption can be used for more complex purposes, such as verifying the message author or surfing the Internet anonymously using the Tor network. Under certain circumstances, encryption can be automatic and simple. Encryption is the way to protect your valuable information, such as documents, images, or electronic transactions on the Internet, from unwanted people to prevent them from being accessed or changed. Encryption uses a "mathematical formula" code, and a key to convert read data "plaintext" to a format that others cannot understand "encrypted text", or “ciphertext”.

Cipher is a generic cryptographic recipe, and the encrypting key (secret key) makes encrypted data unique, only those who know this key can decrypt the encrypted massage. Keys are usually a long string of numbers protected by common authentication mechanisms such as passwords, symbols, or biometrics, like fingerprints or palm prints **(Vinita Keer, Dr. Syed Imran Ali, Neeraj Sharma, (2016).**

Today, encryption technology has a prominent place among science. Its practical applications have varied to include the diplomatic, military, security, commercial, economic, media, banking and informatics fields. It should be noted that the Arabs used the term "blindness" as a process of converting clear text into an incomprehensible text using a specific method.

Hence encryption is the conversion of information from a readable state to a completely opaque state, like a utilitarian puzzle that does not add information to the reader.

There are four main objectives behind the use of cryptography:

- ✓ Confidentiality or Privacy: Confidentiality is a service used to store the content of information from all persons except those who have been selected to be informed.

- ✓ Integrity: A service that is used to save information from changes (such as delete, add or modify) by unauthorized persons to make these changes.

- ✓ Authentication: A service used to establish the identity of the customer with the data (authorized).

- ✓ Non-repudiation: a service used to prevent a person from denying the reception of a message or denying the sending of a message (Akashdeep, GVB, Vinay, Hanumat, 2016)

Therefore, in cryptography the use of mathematics to convert plaintext message PT into an unreadable ciphertext CT, during encryption process E at the sender side, while reconverting ciphertext back to plaintext by decryption process D. Both processes use secret keys, such as K1 and K2, as shown in figure 1.1 (Akashdeep, GVB, Vinay,

Hanumat, 2016). It must be stated the K1 equals to K2 in the case of symmetric systems while they are different in the case of asymmetric systems, as will be explained later on in this thesis.

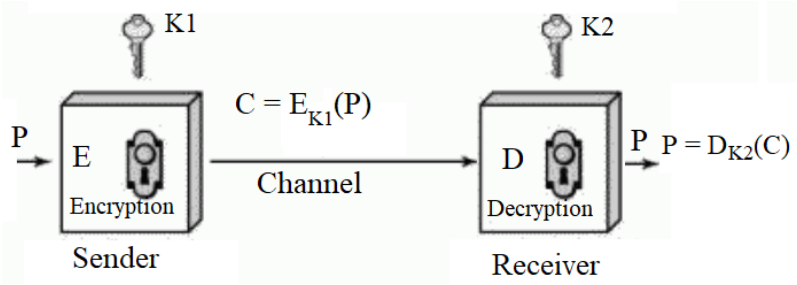

Figure 1.1: Encryption and Decryption processes

1.2.1 Security Algorithms

Security algorithms are generally categorized by relying on the following, as illustrated in figure 1.2 (Akashdeep, GVB, Vinay, Hanumat, 2016).

- How plaintext is converted into ciphertext
- How plaintext is processed
- How keys are used

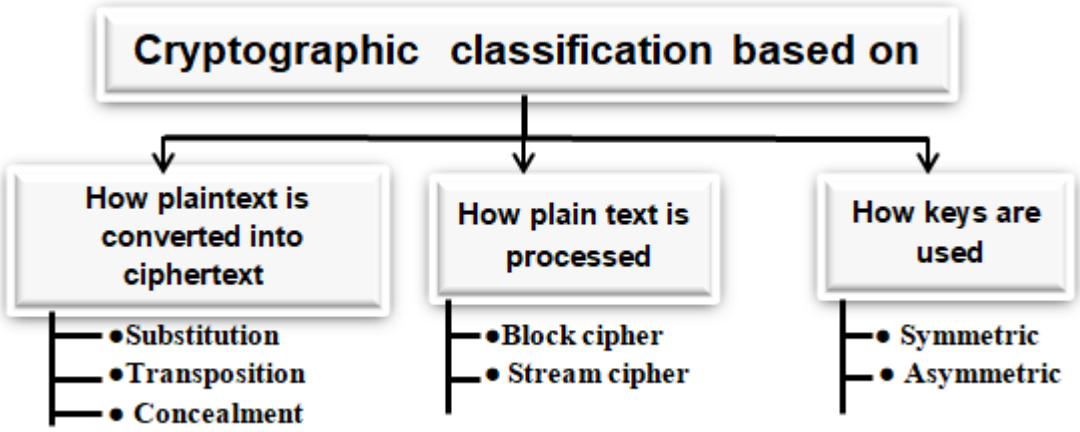

Figure 1.2 : Classification of Algorithms (Akashdeep, GVB, Vinay, Hanumat, 2016)

1.2.2 A symmetric algorithm

In asymmetric algorithms, a pair of related key are used; one key for encryption called the Public key, say K1 and the other is different but inter-related key used for decryption called the Private keys, say K2, when performing transformation of plain text into cipher text. Examble of asymmetric algorithms popularly used are Elliptic Curve Cryptography (ECC), Diffie-Hellman, and Rivest, Shamir, and Adlman (RSA) algorithms.

1.2.3 Symmetric algorithm

Symmetric algorithms involve a single shared secret key to encrypt as well as decrypt data, i.e. K1=K2. Examble of asymmetric algorithms popularly used are Data Encryption Standard (DES and 3DES), Advance Encryption Standard (AES), and BLOWFISH (Akashdeep, GVB, Vinay, Hanumat, 2016).

1.3 Problem Statement

Documents of personal data and information that are usually dealt with and need to be secured corporate financial data, Personally Identifiable Information (PII), medical records, etc. This problem is even more serious and of great concerns to important establishments and organizations, such as national security, military, agriculture, intelligence agencies, etc. Several security systems are available to protect data and information on the computer clouds. They include encryption, data hiding, and authentication. Some of these systems are light and simple having weak security may be hardly enough for not so sensitive data, while others are strong enough but they are heavy and time consuming.

The proliferation of fast wireless networks has encouraged data storage within the cloud, Protecting data from attackers has become urgent to maintain its security and confidentiality, need for security and privacy technologies, systems, and processes to address it.

Security in the Cloud is now the main challenge in Cloud Computing. Due to lack of understanding and proper application, there have been lot of speculations for many organizations to use services of Cloud computing as data is stored at any physical location outside their own control. This facility has raised various security questions like privacy, confidentiality, integrity etc. and demanded a trusted environment where data confidentiality can be maintained. Thus, we need to determine the perfect blend of security using different techniques to provide the most efficient authentication, confidentiality and integrity of data over network.

1.4 Research Questions

Some questions that illustrate the problem discussed in this research:

1- How to develop a lightweight and secure encryption system that can compete with known cryptography systems.

2- Will the proposed lightweight cryptosystem overcome security concerns such as confidentiality and integrity, and have efficient processing execution speed.

3- Would the performance of the proposed lightweight cryptosystem be effective and comparable with other reported lightweight systems.

1.5 Goal and Objectives

The goals of the research lightweight cryptosystem are to achieve the following:

1- An acceptable fast and fair level of security for simple, less important data and information, such as personal day to day information.

2- A high level of security that protect confidential and extremely important data during transfer over the communication link as well as at storage somewhere in the computer cloud, where the client has no direct control or knowledge of the where about it resides.

1.6 Motivation

As using the cloud computing is cheap, handy, and provide available services anywhere and anytime, it is irrational not to benefit from it just because some people are worried about their information. Therefore, it is the researcher intention to encourage computer users to benefit from such service by working on security or increase privacy and

its feeling for user on the computer cloud. It is thought that if even the simple and non-important information are coded in certain level, such work will encourage user to put more faith in the cloud.

Designing a light weight cryptosystem will not put much time complexity on the internet, nor on the manipulation and storage of data and information in the virtual machines.

1.7 Contribution and Significance of the Research

The importance of research can be summarized as the design of a new encryption system which is Simple and Highly Secure Encryption_Decryption (SHSED), It is inspired by International Data Encryption Algorithm (IDEA) presented by Lai and Massey, (1991) in order to improve the security level and execution time measurements to use for data storage in cloud computing.

1.8 Scope of the Study

Implementing the design of lightweight cryptosystem to reach powerful security level and improvement in the encryption execution time measurements to be used for data storage in cloud computing.

1.9 Thesis Outlines

The Thesis is organized in five chapters. The contents of these chapters are given in the following order.

Chapter One: Addressed an introduction to the thesis with some definitions of cryptography in general. It presents the problem statement, research question, motivations, scope, research contribution, and limitations.

Chapter Two: Addresses first the theoretical Background for cloud computing and the description of the related cryptographic systems, namely AES, DES, IDEA, and LED. Then the related work in both cloud computing and lightweight cryptographic systems are reviewed.

Chapter Three: this chapter lists the detailed design of the suggested lightweight cryptographic system. It includes all related materials such as the block diagrams, the algorithm steps for both encryption and decryption processes, and the key generation algorithm.

Chapter Four: It includes the coding, implementing, and testing of the proposed cryptosystem. The obtained results will be analyzed and discussed. The comparison with other systems will be included in this chapter, too.

Chapter Five: this chapter summarizes the work and its obtained results, it gives the conclusion of the work conducted in this thesis first. Then some ideas and thoughts of the possible future work that can result in a progress of the proposed work are followed.

Chapter Two

Theoretical Background and Related Work

2.1 Introduction

This chapter includes some introductory background on cloud computing first then outlines the current security threats to the data and information hosted on the cloud as well as these data that are transmitted over the network. The lightweight cryptography and some of the related cryptographic systems will be briefly defined and explained. Then most related work will be summarized.

2.2 Cloud Computing

A huge collection of virtualized and scalable computer resources that have the capability to accommodate various application and provide different services that are needed by users. It has the “pay only for use” policy, where users pay for the cost of using the resources. Computing Clouds can be considered as a set of connected networks that supply services enabled a quality of service, cheap, when needed, scalable, accessed anywhere, at any time, and can be used for individual or groups (Shilpashree Srinivasamurthy, and David Q. Liu, 2010).

Cloud computing architecture generally consists of Servers, Storage Databases, Services, Applications, and Computer network, as depicted in figure 2.1. Any user outside the cloud can connect to the computer cloud in order to get the required services, which might be using the available computing power of the processors, storing any amount of data and information, or sharing data with other intended users. Obviously, the use of the cloud computing is available on-demand and therefore its cost will be as required.

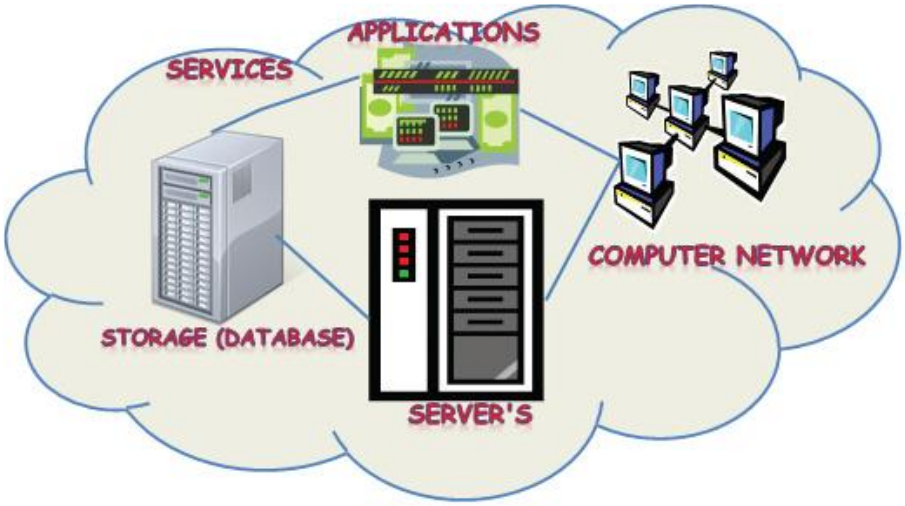

Figure 2.1: Cloud Computing (S. Srinivasamurthy, and D. Q. Liu, 2010)

2.2.1 Cloud computing Characteristics

In more general term, it can be said that cloud computing exhibit the following characteristics (Peter Mell and Tim Grance, 2009).

- **On-demand self-service:** A consumer or the user can unilaterally provision computing capabilities.
- **Broad network access: Av**ailability over the entire network and its access are managed through standard mechanisms in order to encourage the use of the cloud by various platforms.
- **Pooling of resources:** Multiple users from various areas and with virtual resource that are automatically and continuously assigned or changed according to user's demand.
- **Rapid elasticity.** Use of resources and services are changing automatically, to either increase or decrease, i.e. may instantly release or quickly engaged.

- **Measured service:** Computer clouds check and control the use of resource by using special measuring capabilities and may stop or allow at appropriate criteria.

2.2.2 Computer Cloud deployments

Four available models of computer clouds are deployed that can be adopted, namely (Jasleen Kaur, Dr. Sushil Garg , 2015):

- **Private**: this cloud is deployed, observed, and managed for a certain particular distance area but connected through internet to other areas, however, it is only for private branch-branch connection.
- **Public**: such cloud is available to any users, such as, Google-Drive service, and Micro-soft One-Drive. It is useful to public as it provides less cost compared with cost required for own facilities.
- **Hybrid**: the connection and availability of private and public clouds are possible for interchanging services between them in order to offer services for both connected parties.
- **Community**: Huge infrastructure is used in this cloud. It includes but not limited to government organizations that give services for computing and information storage to the computing community.

2.2.3 Cloud Computing Service Models

Basically, cloud computing implementation can be categorized in three different levels or services as shown in figure 2.2, and explained in the following (Aws Naser Jaber, and Mohamad Fadli Bin Zolkipli, 2013).

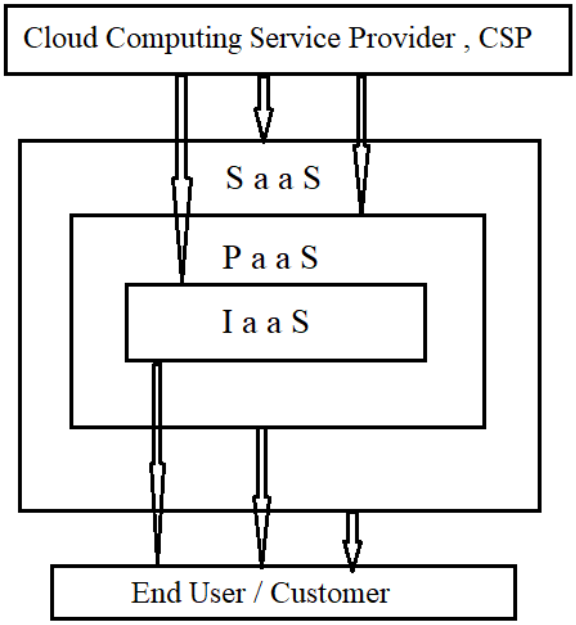

Figure 2.2: Cloud computing service models (Jaber et. al., 2013)

- **Software as a Service (SaaS):**

users buy the privilege to inter the computer cloud and acquire certain application or service hosted their (Vaquero L. M., L. Rodero-Merino, J. Caceres, and M. Lindner, 2008). Some enterprises such as Microsoft is increasing its involvement in such business. There is some web Apps for MS-Office 2010 accessible to Office volume licensing customers and Office Web App subscribers through its cloud-based online services.

- **Platform as a Service (PaaS):**

Users or customers buy access to cloud platforms to implement their own computer programs and applications (Antonova A., E. Gourova, and N. Roumen, 2011). The access to the cloud and the operating system is managed by the cloud service supplier, who may place some constraints on the service as required.

- **Infrastructure as a Service (IaaS):**

It represents the core of the cloud computing system, where users may be able to control and manage system processes, applications, storage, and network connectivity (Vaquero et. al., 2008) and do not merely maintain the cloud infrastructure.

2.3 Cloud Computing Security

Since 2000, cloud computing has begun to emerge, with some well-known companies such as Amazon and Google starting to employ the services provided by them. Since then there has been concern about everything that requires control over security and protection on the cloud (S. Srinivasamurthy, and D. Q. Liu, 2010) and (Deyan Chen, Hong Zhao, 2012). Cloud Security Alliance (CSA) discovered the top seven threats affecting cloud computing, which can be summarized below:

- Cloud Computing Abuse : Misuse of cloud computing by spreading viruses and malware to thousands of devices by sneaking into the public cloud, CSA recommends to lessen this threat by stricter initial registration and validation, using enhance credit card, fraud monitoring, comprehensive checking and controlling public blacklists.

- Insecure Application Programming Interfaces: The need for secure authentication and control of connection to application software interfaces. The recommendations is to propose analysis of the security model of the cloud provider and to have strong controls to allow connection with encryption.
- Malicious Insiders: Some malicious security threats slowdown from the inside, CSA recommend enforcing strict supply chain management, specifying human resource requirements, and requiring transparency into overall information security.
- Shared Technical Vulnerabilities, One of the problems facing a service provider when sharing this technology is its lack of environmental suitability, solutions is to implement security best practices, monitor environment for unauthorized users.
- Loss/Leakage: Loss of data by deleting it without backups or loss of encryption keys and so on, from unauthorized access and no possible access to this data, CSA suggests a strong access control using API and by develop powerful keys, protect the data integrity and implementing powerful security system.
- Account, Service & Traffic Hijacking: Attacks that lead to loss of accounts and denial of service. To counter these attacks, prohibition of sharing account credentials between users and services and understand cloud provider security policies are suggested
- Unknown Risk: The absence of a complete profile that outlines the types of risks that lead to loss of security and confidentiality, CSA suggested disclosure of applicable logs and monitoring on necessary information (Shilpashree Srinivasamurthy, and David Q. Liu, 2010).

Cloud security is a broad and evolving domain that includes network security, information security and computer security and refers to a wide range of policies and

controls to protect data, applications and computing infrastructure. So many areas related to cloud computing security concerns have been developed by the Cloud Security Alliance. These security areas may be put together with more services provisioning module that includes all security issues for infrastructure including network, host and application level as depicted in the cloud computing security architecture, shown in figure 2.3 (Deyan Chen , Hong Zhao , 2012).

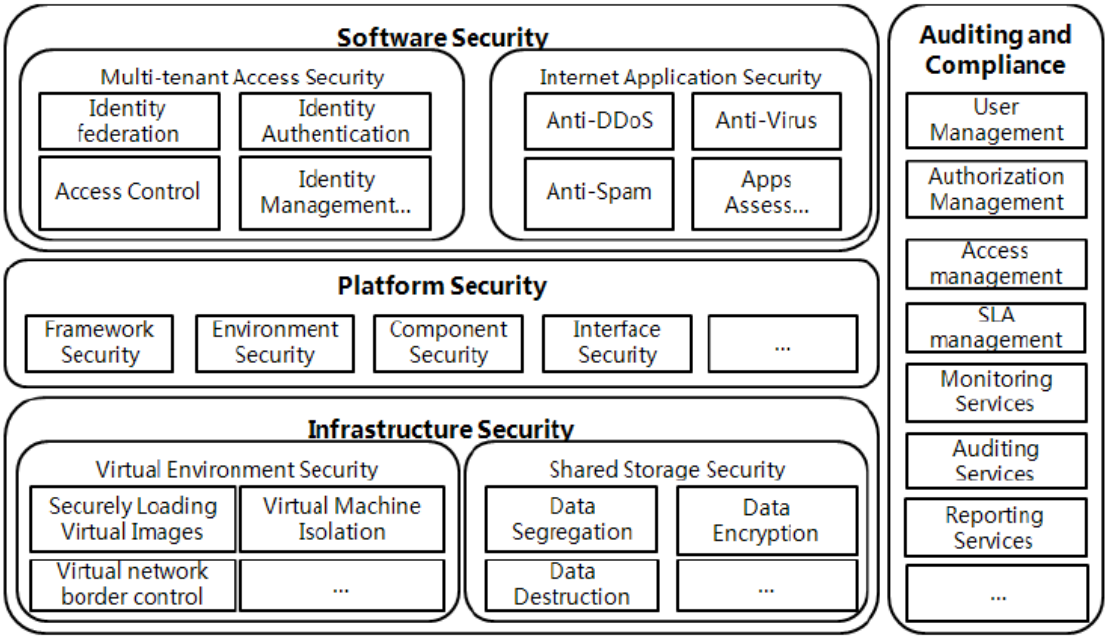

Figure 2.3: cloud computing security architecture (Deyan Chen, Hong Zhao, 2012)

2.4 Light weight cryptography

Over the past few years, several light weight encryption alternatives have been proposed that aim to develop algorithms for use in devices that are unable to provide most existing codes and do not have sufficient resources such as memory size, power consumption and execution time, connectivity hardware

and software, rather than famous and heavy encryption methods such as AES and DES cryptography.

Lightwieght encryption is a switch between lightness and security to reach high levels of security using only small computing power, For example Radio-frequency identification (RFID) tags and smart cards, sensors (Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., ... & Vikkelsoe, C. 2007)

2.5 Hybrid technique

Hybrid Encryption is a combination of two or more cryptographic systems in order to take advantage of their strengths in cloud computing (Vinita Keer, Dr. Syed Imran Ali, Prof. Neeraj Sharma, 2016) such as well-known encryption algorithms AES, DES, 3DES, RSA, Blowfish …etc.

2.6 data classification

A new approach to prevent cloud operators and others from accessing important data stored on the cloud by splitting data into two types; sensitive and normal partitions before encrypting and sending them to storage within the cloud ,

(Li, Y., Gai, K., Qiu, L., Qiu, M., & Zhao, H. 2017). Figure 2.4 shows Architecture of data classification

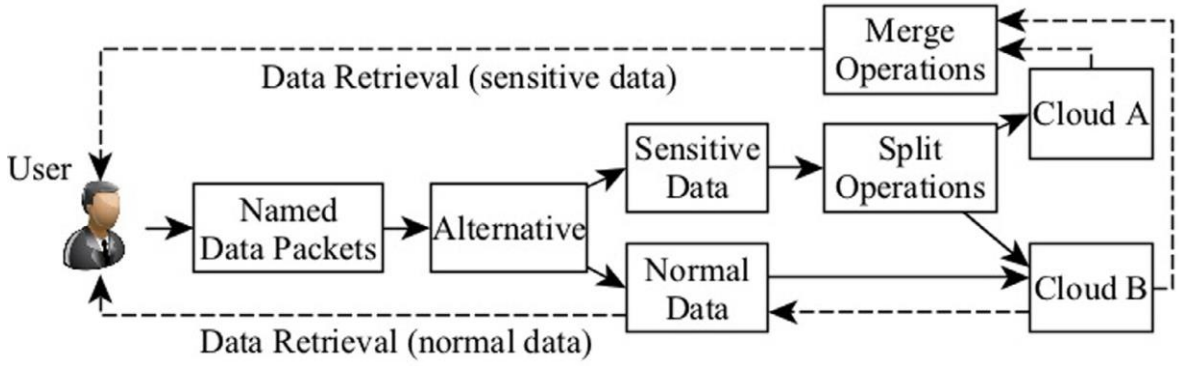

Figure 2.4: Architecture of data classification application (Li, Gai, Qiu, Qiu, & Zhao, 2017)

Data classification is the process of identifying data element with respect to its value, Data classification properties by access control content of data and data storage.(Rizwana Shaikh and Dr. M. Sasikumar, 2015)

2.7 relevant cryptographic system background

This section gives short description for two of the most popular data encryption techniques, namely AES and DES first, then IDEA cryptosystem will be described as it is relevant to the proposed lightweight cryptosystem as it is inspired by it. Then finally one of the newly designed lightweight cryptosystems, named LED included.

2.7.1 AES

Advanced Encryption Standard (AES) was developed as a replacement for Data Encryption Standard (DES), and this was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow. Figure (2.5) shows the structure of AES method, this method has the following features:

- Symmetric block cipher key
- 128-bit data, 128/192/256-bit keys , with 10, 12, and 14 rounds.
- Faster than Triple-DES
- Provide full specification and design details

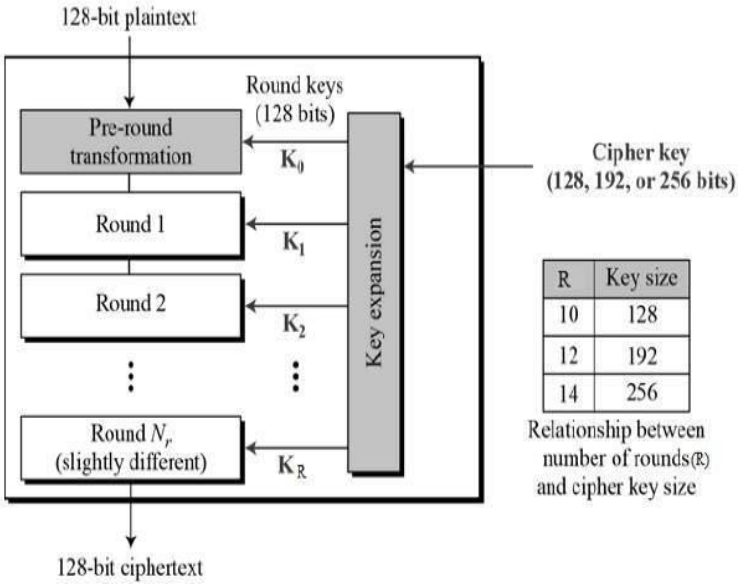

Figure 2.5: Structure of AES method

2.7.2 DES

The Data Encryption Standard (DES)(Shakeeba S. Khan, Prof.R.R. Tuteja, 2015) is a symmetric- key block cipher published in January 1977 by (NIST). the encryption of DES takes a 64-bit plaintext and 48- bit cipher key is used for both encryption and decryption. The encryption process is made of sixteen Feistel rounds , each round consists of substitution , transposition , and exponentiation, as shown in figure 2.6.

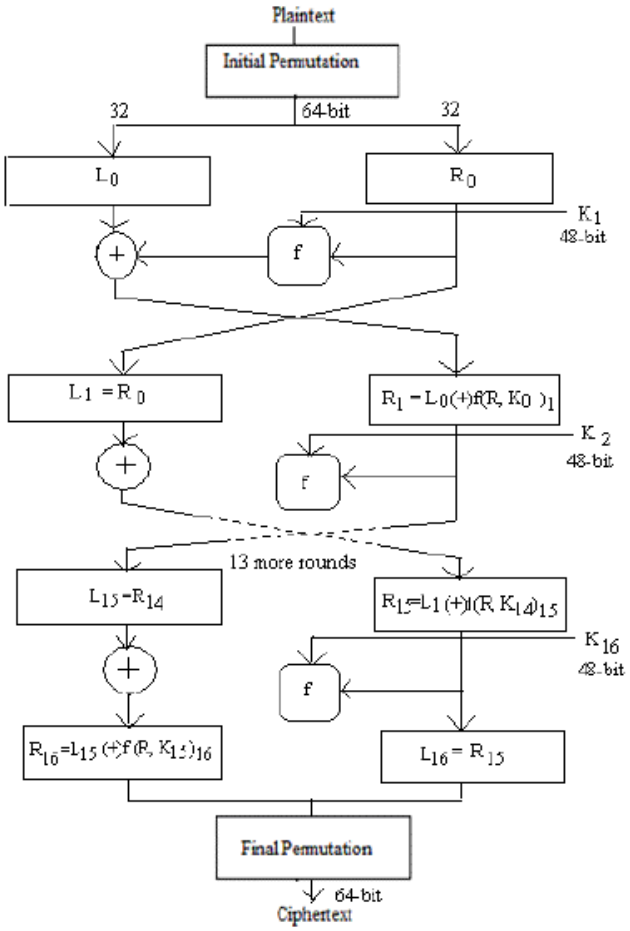

Figure 2.6: encryption DES (Shakeeba S. Khan, and R.R. Tuteja, 2015)

2.7.3 IDEA

The International Data Encryption Algorithm (IDEA) is a symmetric-key, block cipher. It was published in 1991 by Lai, Massey, and Murphy. IDEA is a modification of an earlier cryptographic system called Proposed Encryption Standard (PES), that was published in 1990 by Lai and Massy , it was designed as a replacement for the Data Encryption Standard (DES). Then the name (PES) is changed to IDEA in 1992 (Xuejia Lai and James L. Massey, 1991). The encryption of IDEA takes a 64-bit plaintext and 56- bit cipher key is used for both encryption and decryption. The encryption is one of mixing

operation of different algebraic groups XOR , Addition and multiplication , as shown in figure 2.7.

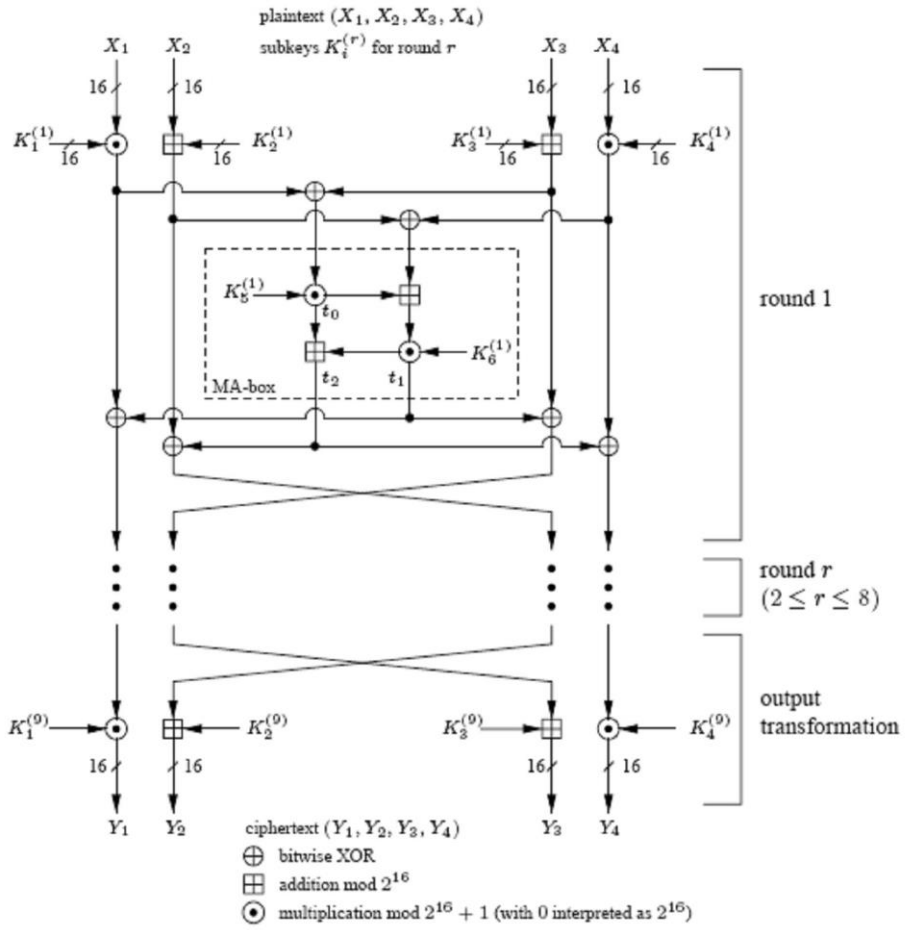

Figure 2.7: IDEA Encryption Structure (Xuejia Lai and James L. Massey, 1991)

2.7.4 LED

Lightweight Encryption Device is a substitution–permutation network structure (SPN) based on AES algorithm. It is made of steps of XORing the key in four rounds with AES style sub cells, shift rows and mix columns . as shown in figure 2.8 .

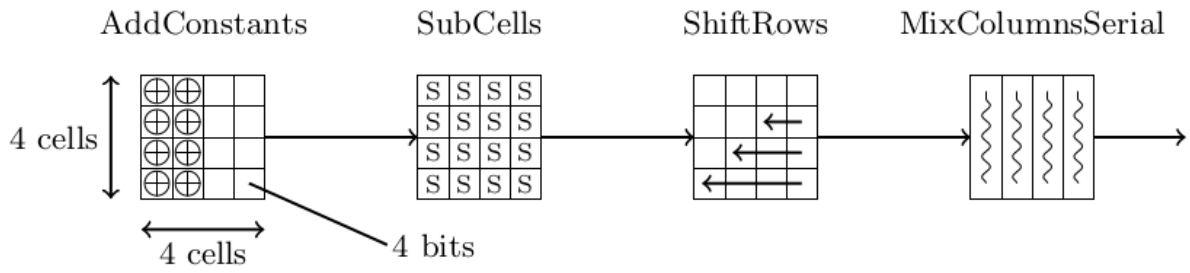

Figure 2.8: The LED round functions (Bogdanov, Knudsen, Leander, Paar, Poschmann, Robshaw, ... & Vikkelsoe, 2007)

LED is lightweight block cipher alternative, use in Radio-frequency identification (RFID) tag deployments, sensor networks, and the small embedded applications which have limited resources, (Bogdanov, Knudsen, Leander, Paar, Poschmann, Robshaw, ... & Vikkelsoe, 2007)

2.8 Comparison

After reviewing previous studies related to the existing cryptosystem scheme, a parameters comparison between these cryptography algorithms are conducted and summarized as given in Table 2.1

Table 2.1 : Parameters comparison of symmetric cryptography algorithms

	AES	DES	LED	IDEA
Developed	**2001**	**1981**	**2011**	**1992**
Block Size (bit)	**128 (bit)**	**64 (bit)**	**64 or 128 (bit**	**64(bit)**
Key Length (bit)	**128, 192 or 256 (bit)**	**56 (bit)**	**64 or 128 (bit)**	**128 (bit)**
Number Of Rounds	**10 or 12 or 14**	**16**	**4**	**1-8**
Security rate	**secure**	**Proven inadequate**	**secure**	**secure**
Possible key	**2^{128} , 2^{192} Or 2^{256}**	**2^{56} bits**	**2^{64} or 2^{128} bits**	**2^{128} bits**

2.9 Related WORK

This section consists of two sub-sections, one will presents related work about the cloud computing security and the other presents related work to the available lightweight cryptographic systems.

2.9.1 Work Related to Cloud Computing Security

So many researchers have dealt with the problem of cloud computing security, however, the most recent and related papers to the project in this study will be mentioned here after, as follows:

(Deyan Chen and Hong Zhao ,(2012)) concentrated on the security of cloud computing incidents, such as the loss of Amazon's storage service, and severe leak of user information in Google Docs. They listed security issues related to network level, host level, and application level. They discussed privacy protection issues associated with cloud computing through the stages of the data life cycle: such as data generation, transfer, use, share, storage, archival and destruction of data.

(Omar K. Jasim, Safia Abbas, El-Sayed M. El-Hprbaty and Abdel-Badeeh M.Salem, (2013)) presented a comparison between the symmetric algorithm (e.g. DES, 3DES, and AES), and asymmetric algorithms (e.g. RSA) in terms of Key Length ,Rounds, Block size, Security rate, Execution Time. On the basis of the results, the algorithms implemented are more efficient on cloud environment.

(Rachna Arora and Anshu Parashar,(2013)) defined cloud computing as a set of devices, storage, networks, interfaces, and services that provide the means, computing power, and services on demand. They also mentioned the advantages of traditional cloud computing from agility to low cost of access, device autonomy, location and scalability. Some security issues and concerns with some solutions were considered such as loss of cloud physical security and secure data transfer solution ensuring data integrity during transport, storage and retrieval. They suggested a solution to separate the data, and the client who must control the encryption / decryption keys. Data card industry must provide data records to security managers and regulators, and the solution will control user access. They also explained the symmetric Advanced Encryption Standard (AES), Data Encryption Standard (DES), Blowfish algorithm, and Asymmetric cryptographic algorithm (RSA).

They made a comparison between the above-mentioned coding methods in terms of the following: Characteristics, Platform, used key size, scalability, initial vector size, security, data encryption capacity, authentication type, memory usage, required maximum time, and execution time.

(Mandeep Kaur, Manish Mahajan, (2013)) proposed the use of hybrid security methods to improve the protection of data stored on cloud computing such as : AES , DSA , Blowfish , RSA , Eclipse IDA .This paper focused on not using a third party to encrypt client data, but giving the customer the authority to determine ways to encrypt his data.

Priya jaiswal, Randeep kaur, Ashok Verma,(2014) introduced a definition for cloud computing as a large network providing all kinds of facilities required by the user such as operating systems, software applications, cloud data storage, sources. They also suggested some important privacy protection measures for hardware component such as random access memory (RAM) for the 2009 cloud computing service.

(Randeep Kaur and Supriya Kinger, (2014)) presented some of the challenges that cloud computing users are facing such as: security and privacy, lack of standards and continuous evolvement. They highlighted some benefits to cloud service providers such as: cost savings, scalability, flexibility, reliability, and mobile accessibility. The authors explained the methods used in security and traditional privacy and compared the algorithms used to protect data stored on the cloud.

(Lo'ai Tawalbeh, Nour S. Darwazeh, Raad S. Al-Qassas and Fahed Al-Dosari, (2015)) proposed a cloud computing model based on data classification to reduce overhead and processing time, The data were divided into three layers: basic level for encrypting the general type of data, confidential level for data with medium confidentiality degree, and highly confidential levels to handles the most important data.

(Jasleen Kaur, Dr. Sushil Garg, (2015)) proposed the use of hybrid security methods to improve the protection of data stored on cloud computing such as : RSA algorithm , RSA Digital Signature , Encryption/Decryption with Blowfish algorithm, Fiestel, and XOR operations algorithms.

(Shakeeba S. Khan, Prof.R.R. Tuteja, (2015)) explained the symmetric Data Encryption Standard (DES) and the asymmetric cryptographic algorithm (RSA). And they explained how to merge two different algorithms, such as DES and RSA in order to eliminate the security challenges of Cloud Storage.

(Shikha Rani, Shanky Rani, (2016)) listed and conducted a survey of previous studies devoted to data protection on the cloud. They proposed a method of hybrid security encryption using Blowfish and MD5 to provide enhancing security on the cloud server.

(Akashdeep Bhardwaj, GVB Subrahmanyam, Vinay Avasthi, Hanumat Sastry, (2016)) reviewed classification of algorithms by classified broadly as: Symmetric Algorithms , Asymmetric Algorithms, Signature Algorithms, and Hash Algorithms. Their

classification was done after answering the questions: how plaintext is converted into ciphertext?, How keys are used?, And How plain text is processed?

(Vinita Keer, Dr. Syed Imran Ali, Neeraj Sharma, (2016)) studied some of the most popular cloud service providers, such as Microsoft (Azure and One Drive), and Google (Google Drive). They investigated cryptographic algorithms that mostly used in cloud computing: Modern Cryptography (e.g. AES, RC6, DES, 3DES, and BLOWFISH), Searchable Encryption, Homomorphic Encryption, and Attribute based Encryption. They also developed a definition of hybrid encryption as combining two or more cryptographic methods to take advantage of the power of each method to protect data on the cloud.

(Yibin Li , Keke Gai , Longfei Qiu , Meikang Qiu , Hui Zhao,(2016)) proposed the use of an intelligent encryption approach to protect the storage of data distributed on the cloud independently for fear that the operators of the cloud computing might have some influence on the cloud computing security.

(Jean Raphael Ngnie Sighom , Pin Zhang and Lin You,(2017)) made a comparison of IDAs, SHA-512, 3DES, and AES-256 consists of encoding and decoding data on premise. This algorithms achieves far a greater degree of security and also better performance for small and large data files.

2.9.2 Work Related to Lightweight Cryptographic Systems

So many lightweight symmetric cryptosystems have been developed and reported for appropriate applications, such as LED. HIGHT, LBlock, DESL, CLEFIA, PRESENT, TWINE, RECTANGLE, SIT, etc.

Hong D., J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee, (2006) reported a 64 bits block size lightweight cryptosystem having 128 bits key, iterated in 32 rounds, and having two types of operations; XOR operation combined with left or right rotations. Its design was suited for hardware implementation on ubiquitous devices, such as wireless sensor nodes and RFID tags, having almost the same chip size as AES but works much faster. Its was tested for security using differential attack giving results slightly less than the exhaustive search.

Shirai T., K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, (2007) reported a symmetric block cipher developed called CLEFIA-128, it is developed by Sony and designed to be suitable for both hardware and software, encrypting data block of 128 bits under 128 bits key length and 28 rounds with Feistel structure. Many versions of CLEFIA were reported with 192 or 256 bits key lengths running 22 and 26 rounds, respectively. CLEFIA implements 2 different S-boxes of 8 bits, followed by a diffusion matrix multiplication inspired from the AES Mix Columns operation. Using the impossible differential attack against CLEFIA reduced to 12 rounds for a 128 bits key with 2119 encryptions. Surely the execution time increases for longer keys.

Leander G., C. Paar, A. Poschmann, and K. Schramm, (2007) developed two versions of lightweight Data Encryption systems, i.e. DESL and DESXL. DESL used a

single S-Box instead of different ones with no initial and final permutations, on the other side, in DESXL, a whitening step is used improve the security by using a key of 184 bits length. No attack has been exhibited against DESL and DESXL as claimed by them.

Bogdanov A., L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. (2007) presented an ultra-light block cipher named "PRESENT'. It implements substitution-permutation network (SPN) structure with a block size of 64 bits and 80 or 128 bits key size running 31 rounds with multiple uses of 4 bits S-box. This system is also optimized for hardware implementation.

Cannire C., Orr Dunkelman, and M. Kneevi, (2009) also reported an efficient hardware-oriented block ciphers that process block size of 32, 48, or 64 bits with 80 bits key length. Two types were suggested, namely KATAN and KTANTAN differing only in the key scheduling, they use two Boolean functions with no shifting, running for 254 rounds in total. Both algorithms, KATAN and KTANTAN have a serialized structure.

Wenling Wu and Lei Zhang, (2011) presented a lightweight cipher called "LBLOCK". It also has SPN structure and efficiently implemented in both software and hardware. LBLOCK is designed with block size of 64 bits having a a key of 80 bits and run for 32 rounds.

Guo J., T. Peyrin, A. Poschmann, and M. J. B. Robshaw, (2011) suggested a lightweight block cipher called LED. It presents a reasonable performance efficiency for software implementation. It encrypts 64 bits blocks with different key sizes. They are 64 bits, 80 bits, 96 bits and 128 bits in length. The same S-box used for PRESENT cipher are used here in the execution of LED cipher system.

Gong Z., S. Nikova, and Y. W. Law, (2012) suggested lightweight cryptosystem also of SPN technique. It is also fit for software implementation as they claim. Having legacy sensor platforms, and also it is suitable for hardware application. It implements 4 bits/16 S-boxes for substitution, and Rotate-4 bits and Mix-4 bits for permutation. It is called “KLEIN” ciphers with 64 bits block using key of different length, namely 64, 80, or 96 bits running for 12 or 16 or 20 rounds, respectively.

Suzaki T., K. Minematsu, S. Morioka, and E. Kobayashi, (2013) reported a generalized Feistel structure with multi-platform cryptosystem called “TWINE”. It is claimed that it has extremely-small hardware size, with efficient on embedded software. It is of 64 bits block size running 36 rounds with key of either 80 or 128 bits length, and each round involves a nonlinear substitution layer with 4-bits S-boxes and 4 bits block permutation layer.

Zhang W., Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, (2014) reported cryptosystem called “RECTANGLE” which designed for 64 bits block size with key length of either 80 or 128 bits, running only 25 round. It is an ultra-lightweight bit-slice block cipher, which is found suitable for multiple platforms achieving highly competitive software performance and requires very low area in hardware.

2.10 Summary

In this chapter, various studies related to:

- Cloud computing security in general was reviewed, analyzed and summarized, using different cryptography systems of encryption, and comparisons are made in terms the advantages, disadvantages and goals of this methods.
- Hybrid security methods were mentioned to improve the protection of data stored on cloud computing environment.
- Lightweight symmetric cryptosystems which are designed for both hardware and software were reviewed, and comparisons were made in terms of structures, key lengths, and number of rounds.

Chapter Three

The suggested lightweight cryptosystem

3.1 Introduction

This chapter proposes a new lightweight encryption/decryption algorithm that is suggested to be used for cloud computing security. The algorithm is simple and highly secure encryption_decryption (will be referred to as SHSED). It is inspired by the International data encryption algorithm (IDEA).

3.2 Proposed Technique

The proposed technique in this research work suggests the newly designed lightweight encryption algorithm (SHSED) that is based on (IDEA) cryptosystem. It is suggested to be used for data storage on cloud computing.

This chapter will include the design and implementation of SHSED, then its comparison with other cryptographic algorithms such as the widely used AES and DES algorithms that are used in cloud computing together with lightweight algorithms. Then examples are included and results were output.

3.3 Modified Crypto System

The main idea of the SHSED cryptography is to used 64-bit size block, 256-bit key length and 7-bit constants variables CST, besides the number of rounds ranges from 1 to 31 rounds. Passing through initial and final state in each of the encryption and the decryption processes. The procedure followed in this algorithm implements mixed operations of different algebraic groups, namely XOR and Addition operations. Figure 3.1

shows a general overview of the encryption process, together with the key scheduling step. It accepts the plaintext data and the encryption key prior to produce the ciphertext data.

The algorithm uses two types of keys; work keys (WK) and sub-keys (Sk). A detailed descriptions of these keys will be given later in this chapter.

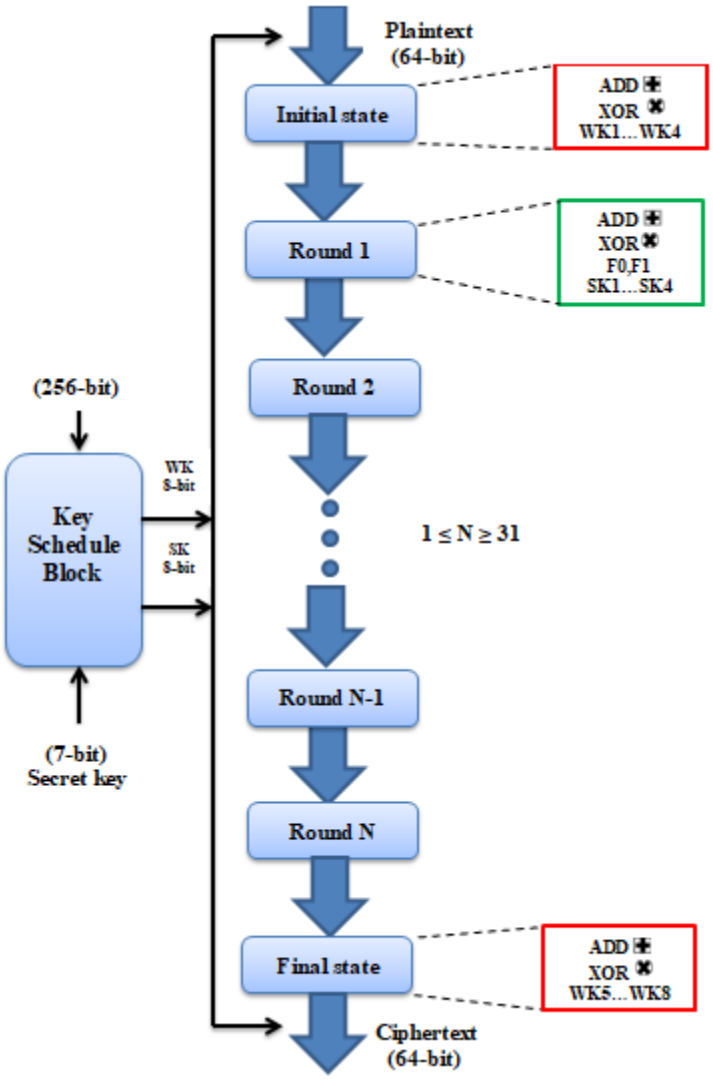

Figure 3.1 : Structure of modified cryptosystem

In the following, the encryption process, decryption process, and the key generation process will be outlined.

3.4 Encryption process

The message to be encrypted is segmented into plaintext blocks (each is referred to as M) of 64 bits length, as shown in figure 3.1. Then each block is encrypted through the following steps:

Step 1. The input 64 bits block, M is divided into 8 sub-blocks of 8 bits each, namely P1, P2, P3, P4, P5, P6, P7 and P8.

Step 2. (Initial state steps): Each sub-block is treated by mixing operation from different algebraic groups they are XOR and Addition operations, using work key that are described later .WK1, WK3 Add with P1,P5 . then WK2, WK4 are XORed with P3, P7 . As shown in Figure 3.2 .

This step will be repeated also in the final state using different Work keys WK.

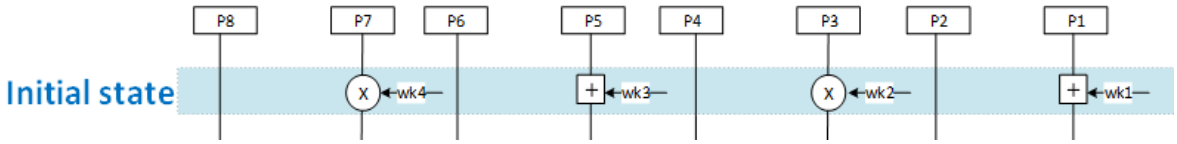

Figure 3.2 : Structure of initial state SHSED encryption proposed

Figure 3.3.a show the block diagram for operations in round 1, in which the following steps are performed.

Step 3. (round 1): The result of P1 becomes (X1,1), use this result again to have (X1,2) by treating it with F1 function, then Add result to sub key SK1 (also will be described later) then XOR it with P2.

Step 4. The result of P3 becomes (X1,3), use this result again to get (X1,4) by treated with F0 function, then XOR the result with sub key SK2 then Add the result with P4.

Step 5. The steps 3 and 4 repeat as they are using the sub-keys SK3, SK4 to get (X1,5), (X1,6), (X1,7) and (X1,8).

[It should be noted that in (X1,1) to (X1,8), the first number represents the round number and the second parameter represents the byte number].

Step 6. The output of the (round 1) are circular shifted to the right, as shown in Figure 3.3.b. Hence the value of (X1,1) becomes (X1,2), (X1,2) becomes (X1,3), (X1,3) becomes (X1,4), (X1,4) becomes (X1,5), (X1,5) becomes (X1,6), (X1,6) becomes (X1,7), (X1,7) becomes (X1,8). (X1,8) becomes (X1,1) for the next round. This step is the end of the round.

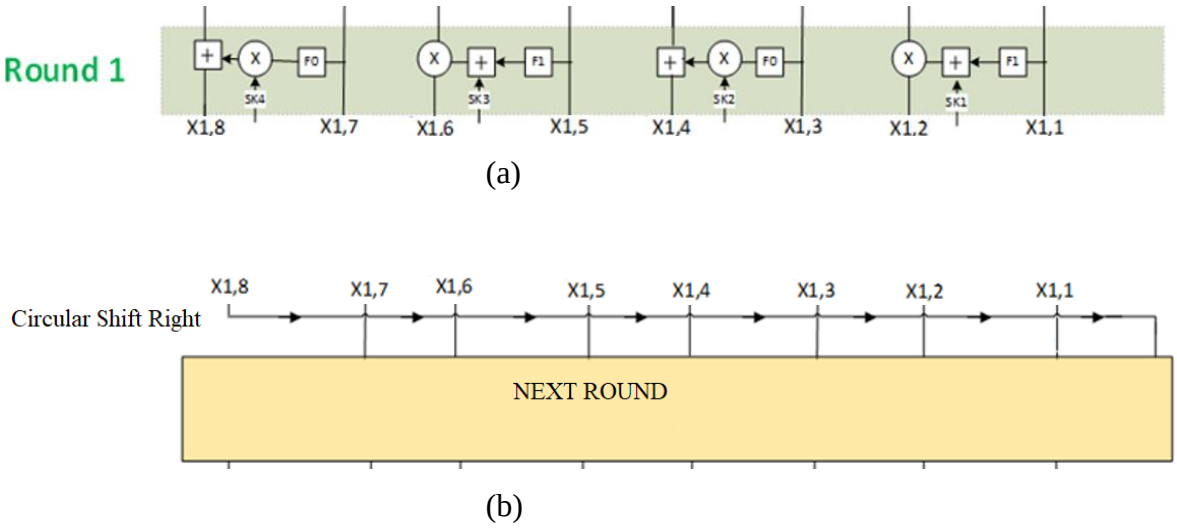

Figure 3.3: Block diagram for (a) Round 1 operations, (b) circular shift operation.

Step 7. The same operations are performed for (n) of rounds, where n is considered as part of the secret key.

Step 8. The result of last round (round n) becomes the new values treated by the same steps of initial state using work keys (WK5, WK6, WK7 and WK8) this step called final state.

Step 9. The output of final state is taken as the ciphertext block Ci of the message block Mi.

All processes involved in performing the proposed SHSED algorithm for n rounds are illustrated in the block diagram of Figure 3.4. Encryption process contains more than one round, the number of rounds ranges from 1 to 31 rounds. After each round the value of (X i , j; where I is the round number and j is the byte number in the processed data block) will be circular shifted. In order to be able to illustrate the processes involved in the proposed algorithm clearly, Figure 3.5 illustrates the block diagram for SHSED algorithm performed for three rounds.

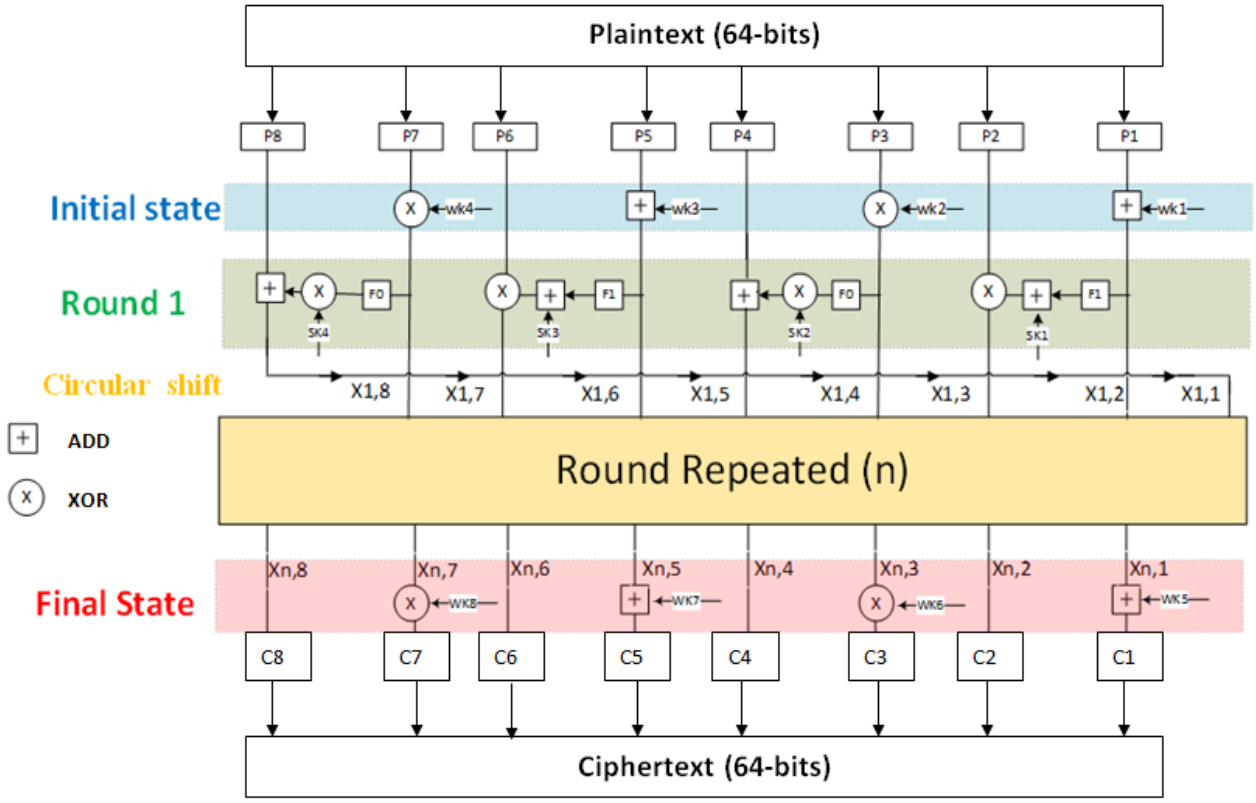

Figure 3.4 : Structure of encryption processes for the proposed SHSED algorithm.

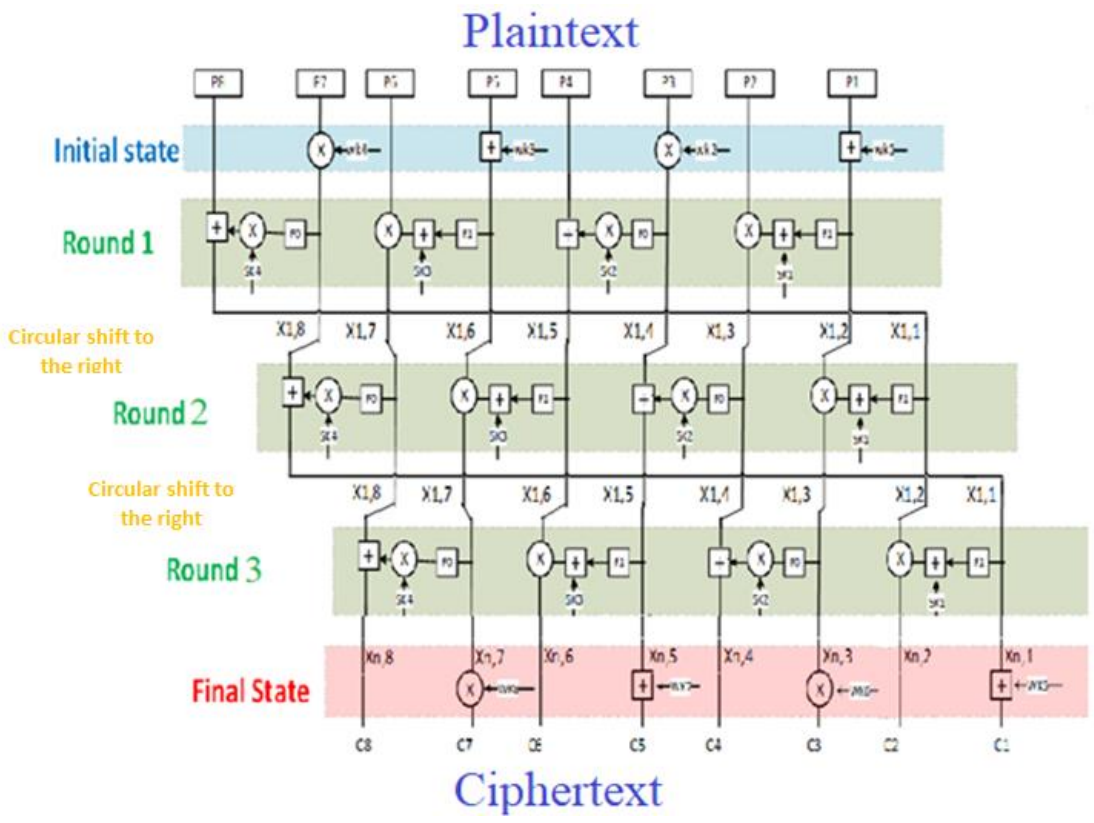

Figure 3.5 : Flowchart for 3 rounds encryption Process

3.5 Details of SHSED Algorithm Processes

1- Initial state

$X_{0,1} = P_1 \boxplus WK_1$,

$X_{0,3} = P_3 \otimes WK_2$,

$X_{0,5} = P_5 \boxplus WK_3$,

$X_{0,7} = P_7 \otimes WK_4$

$X_{0,2} = P_2$, $X_{0,4} = P_4$, $X_{0,6} = P_6$, $X_{0,8} = P_8$

2- Final state

$C_1 = X_{n,1} \boxplus WK_5$,

$C_3 = X_{n,3} \otimes WK_6$,

$C_5 = X_{n,5} \boxplus WK_7$,

$C_7 = X_{n,7} \otimes WK_8$

$C_2 = X_{n,2}$, $C_4 = X_{n,4}$, $C_6 = X_{n,6}$, $C_8 = X_{n,8}$,

3- Round state(r) with circular shift (Each round has 4 different sub keys)

$X_{r,1} = ((F0 \ (X_{r-1,7})) \otimes SK_4) \boxplus X_{r-1,8}$

$X_{r,2} = X_{r-1,1}$

$X_{r,3} = ((F1 \ (X_{r-11})) \boxplus SK_1) \otimes X_{r-1,2}$

$X_{r,4} = X_{r-1,3}$

$X_{r,5} = ((F0 \ (X_{r-1,3})) \otimes SK_2) \boxplus X_{r-1,4}$

$X_{r,6} = X_{r-1,5}$

$X_{r,7} = ((F1 \ (X_{r-1,5})) \boxplus SK_3) \otimes X_{r-1,6}$

$X_{r,8} = X_{r-1,7}$

Key scheduling:

1- Work key generation

```
function wk=Work Key(K)
%% create the work key that is used before the first round and after the last round
 for i=1:8
  if i<=4
   wk(i)=K(i+12);
  else
   wk(i)=K(i-4);
  end
 end
end
```

2- Light Weight key generation

The 256 bits encryption/decryption key is to be divided into 32 equal parts:

K1, K2… K32:

$$WK_i = WK_i \otimes WK_{33-i \ldots i} = 1 \longrightarrow 32$$

3- Secrete number generation(cst):

The following function to be used to generate this constant:

```
function cst=constantGen()
%% Generate constants used for the round
   cst='1101100';
 for i=1:127
   cst(7+i)=dec2bin(bitxor(bin2dec(cst(i+3)), bin2dec(cst(i))), 1);
 end
end
```

4- Extract the correct constant for the round i from the vector cst using the following function:

```
function c=extract_const(cst, i)
%% Extract the correct constant for the round i from the vector cst
 c='';
 for j=i:i+6
   c=[cst(j+1) c];
 end
end
```

5- Generate sub key (SK_i) using the following function:

```
function sk=subkey(K, cst)
%% Generated the subkey used during the round
 for i=0:7
   for j=0:7
     c=extract_const(cst, 16*i+j);
     sk(16*i+j+1)=mod(K(mod(j-i, 8)+1)+ bin2dec(c), 256);
     c=extract_const(cst, 16*i+j+8);
     sk(16*i+j+9)=mod(K(mod(j-i, 8)+9)+ bin2dec(c), 256);
   end
 end
end
```

6- Rotate function F**0(X)**

$F_0(X) = (X <<< 1) \otimes (X <<< 2) \otimes (X <<< 7)$

7- Rotate function **F1(X)**

$F_1(X) = (X <<< 3) \otimes (X <<< 4) \otimes (X <<< 6)$

3.6 Decryption

The computational process used for decryption of the ciphertext block Ci is essentially the same as that used for encryption of Mi block .

Ci block of 64 bits length is first divided into 8 sub-block, then treated by mixed XOR and Sub operations using the same work keys. These steps are illustrated in figure 3.5, and since they are the inverse of the encryption operation, there will be no need to write the detailed steps for the process.

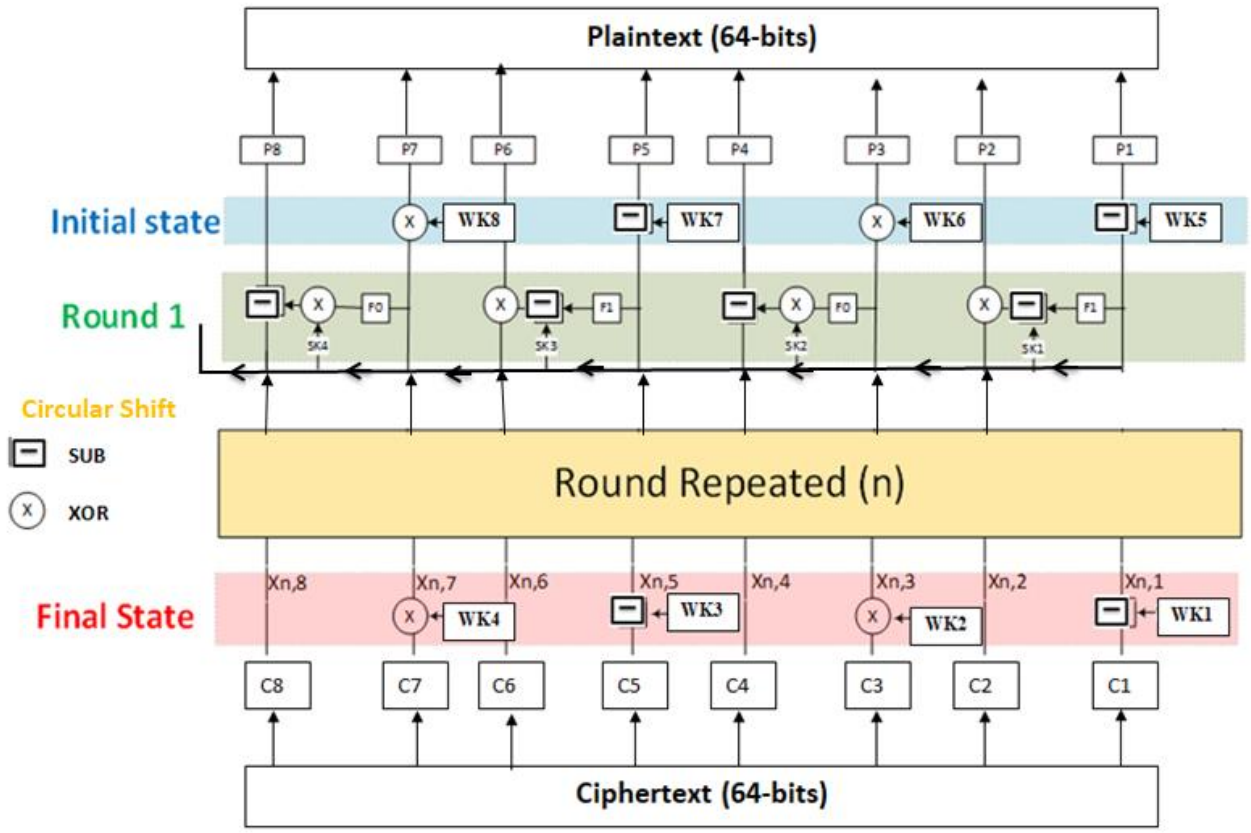

Figure 3.6 : Structure of decryption algorithm of SHSED proposed

Decryption process can have a number of rounds equal to the same number of those used for the encryption. After each round the value of (X i , j) will be circular shifted to the left as shown in figure 3.6. An example for decryption process that performs three rounds is illustrated in Figure 3.7. It is obvious that the operations are exactly the inverse of those followed during the encryption processes.

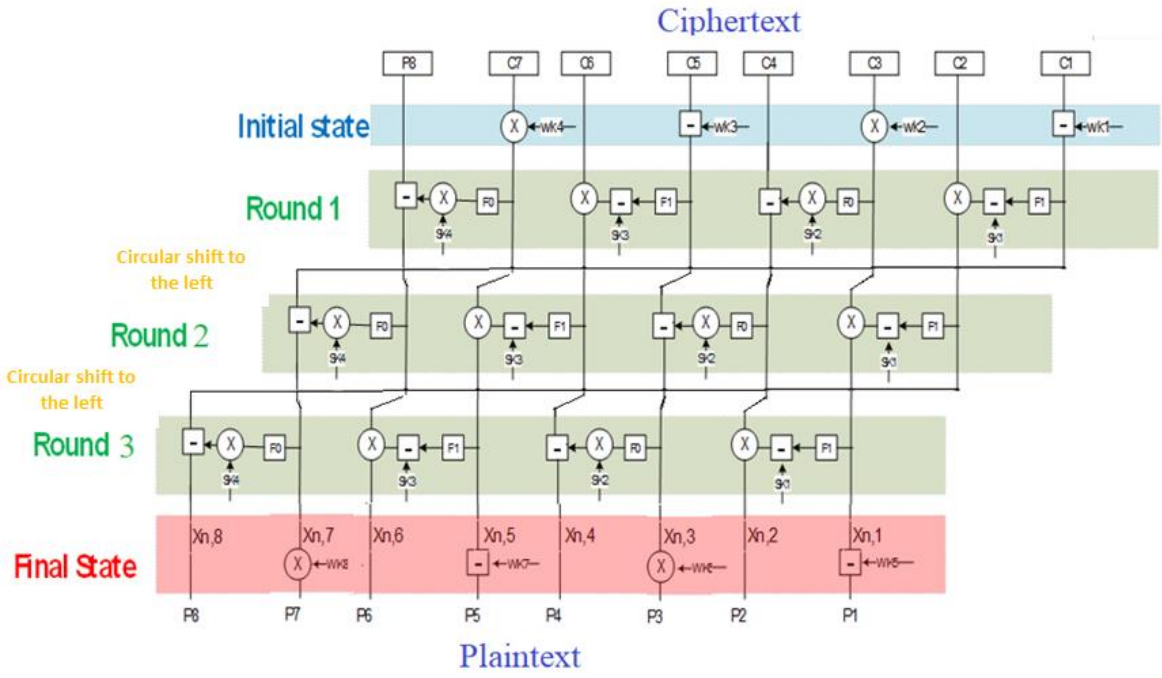

Figure 3.7 : Diagram for 3 rounds decryption Processes using SHSED algorithm

3.7 Experimental calculations results

The inputs are:

1. Data block=

ABCDEFGH → (41 42 43 44 45 46 47 48) hexadecimal

→ (65 66 67 68 69 70 71 72) decimal

2. Encryption key (K) =

(ffeeddccbbaa99887766554433221100050a13203567541806677882829303132) hex.

3. Constant= (1101100) binary, calculated cst= (D833539ED0ABE946) hex.

4. Number of rounds = 3

Table 3.1, list out the obtained values for encrypting a message block for three rounds as an example of the encryption process. The table shows the decimal values for the message block bytes for all the stages of the three rounds from plaintext to ciphertext.

Table 3.1 : Example of SHSED encryption algorithm using 3 rounds

State	Values							
	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**
Original	72	71	70	69	68	67	66	65
Initial	52	71	171	69	146	67	12	65
Round 1	242	52	8	171	121	146	195	12
Round 2	139	242	251	8	167	121	39	195
Round 3	12	139	84	251	151	167	173	39
Final	196	84	41	151	165	173	169	12
Encrypted	12	139	84	251	151	167	173	39

The inverse process for the above example, i.e. the reproduction of the plaintext of the message block from its ciphertext is shown in the Table 3.2.

Table 3.2 : Example of SHSED decryption algorithm using 3 rounds

State	Values							
	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**
Encrypted	12	139	84	251	151	167	173	39
Final	196	84	41	151	165	173	169	12
Round 3	12	139	84	251	151	167	173	39
Round 2	139	242	251	8	167	121	39	195
Round 1	242	52	8	171	121	146	195	12
Initial	52	71	171	69	146	67	12	65
Original	72	71	70	69	68	67	66	65

3.8 Parameter comparison

In table 3.3, a brief comparison is listed showing the key length, block size, cipher type and the security strength for the widely used cryptographic systems, namely DES, AES, IDEA, LED, together with the proposed lightweight algorithm in the thesis (SHSED). From this table, one can notice that the proposed algorithm is highly flexible with high security strength.

Table 3.3 : Comparison of symmetric cryptography algorithms

	AES	DES	LED	IDEA	SHSED
Key length(bit)	128, 192 or 256	56	64 or 128	128	64,128, or 256
Block size(bit)	128	64	64 or 128	64	64, 128, or 256
Possible key	**2^{128} , 2^{192} Or 2^{256}**	**2^{56} bits**	**2^{64} or 2^{128} bits**	**2^{128} bits**	**2^{64} , 2^{128} Or 2^{256}**
Security rate	Secure	Proven inadequate	Secure	secure	Highly Secure

Chapter Four

Implementation and Results

4.1 Overview

This chapter outlines the coding, testing, and implementation processes of the designed algorithm of chapter 3, (i.e. SHSED). First a data set is selected for the experimentation, then the algorithm is tested for different messages lengths, different keys, and different number of rounds. The obtained results for SHED algorithm are analyzed and compared with other cryptographic systems, such as DES, AES, and LED., using the same values for all the parameters involved.

4.2 Dataset

Dataset is a collection of texts files used in implementing process for the Proposed SHSED. The dataset is chosen with different message sizes, start from 32 bytes to 96802 bytes.

4.3 Implementation Steps

The Proposed (SHSED) algorithm is implemented using MATLAB (version R-2010) programming.

4.3.1 Hardware Specification

For the implementation and testing of SHSED, the personal computer (PC) is used for installation and execution, with the following technical features:

- Operating system: Windows 10 home 64-bit.
- Processor: AND A10-9600P RADEON R5, 10 COMPUTE CORES 4C+6G GHz.
- Installed Memory (RAM): 6.00 GB

4.4 Performance analysis of cryptography algorithms

One of the most important goals of the performance analysis for various cryptography algorithms process is the time computational cost. Hence, the computational cost (execution time) for the proposed SHSED algorithm, together with the other algorithms under consideration, namely AES, DES, and LED are computed for both encryption and decryption and listed in the following sub-sections.

4.4.1 Computational cost for the proposed SHSED algorithm

Table 4.1 lists out the computational cost or the execution time taken for encryption and decryption for the proposed SHSED algorithm. It included a number of experiments, and the average time is also determined. Then, these results are plotted in figure 4.1. It must be stated that the same message is used for all the measurements involved.

Table 4.1: Execution time for the proposed SHSED algorithm

Experiment #	**SHSED** Encryption time(Seconds)	**SHSED** Decryption time(Seconds)
1	0.0980	0.0820
2	0.0950	0.0880
3	0.1020	0.0850
4	0.0990	0.0840
5	0.1000	0.0840
6	0.1050	0.0870
7	0.1010	0.0690
8	0.0970	0.0850
9	0.1010	0.0850
10	0.0930	0.0940
Average	**0.0991**	**0.0843**

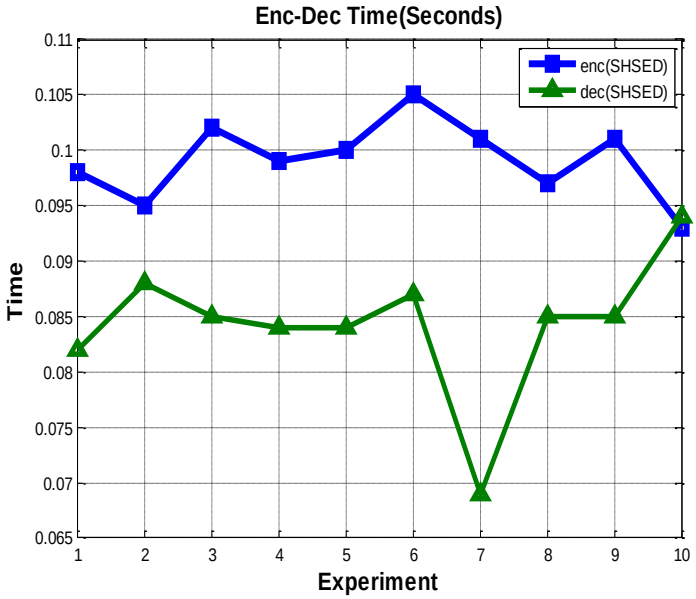

Figure 4.1: Encryption/decryption execution time for SHSED analysis

The fluctuations noticed in execution time can be attributed to the time complexity of the operation in the CPU, however, the average time is more meaningful and reflect the efficiency of the considered cryptographic system.

4.4.2 Computational cost for AES algorithm

Table 4.2 lists out the computational cost or the execution time taken for encryption and decryption for AES algorithm, and these results are plotted in figure 4.2. It must be stated that the same message is used for all the measurements involved.

Table 4.2: Execution time for AES algorithm

Experiment #	**AES** Encryption time(Seconds)	**AES** Decryption time(Seconds)
1	0.4260	0.5920
2	0.4180	0.5030
3	0.4420	0.5450
4	0.5220	0.4670
5	0.3940	0.4520
6	0.3720	0.5370
7	0.3470	0.6000
8	0.5830	0.5570
9	0.4780	0.5320
10	0.4680	0.6230
Average	**0.4450**	**0.5408**
Speed up of SHSED	**0.4450/0.0991= 4.4904**	**0.5408/0.0843= 6.4151**

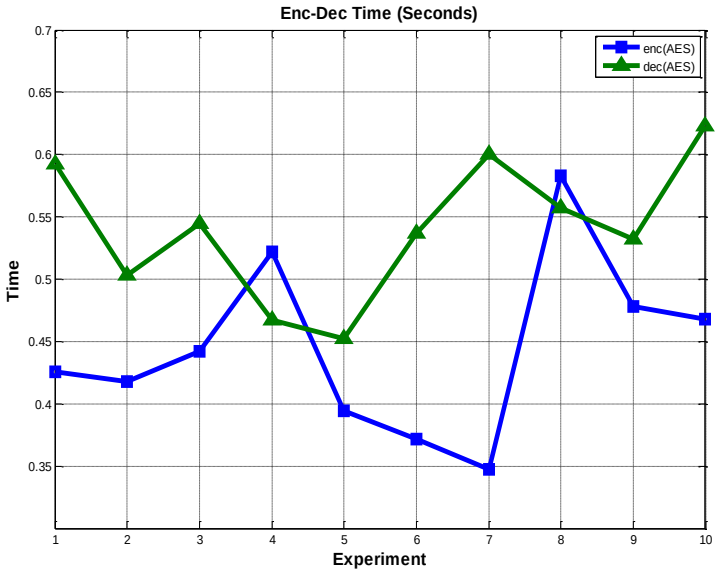

Figure 4.2: Encryption/decryption execution time for AES

4.4.3 Computational cost for DES algorithm

Table 4.3 lists out the computational cost or the execution time taken for encryption and decryption for DES algorithm, and the obtained results are plotted in figure 4.3. also the same message is used for all the measurements involved.

Table 4.2 : Execution time for DES algorithm

Experiment #	**DES** Encryption time(Seconds)	**DES** Decryption time(Seconds)
1	0.0690	0.0670
2	0.0570	0.0620
3	0.0590	0.0690
4	0.0560	0.0830
5	0.0480	0.0690
6	0.0380	0.0570
7	0.0510	0.0430
8	0.0490	0.0660
9	0.0610	0.0670
10	0.0460	0.0590
Average	**0.0534**	**0.0642**
Speed up of SHSED	**0.0534/0.0991= 0.5388**	**0.0642/0.0843= 0.7615**

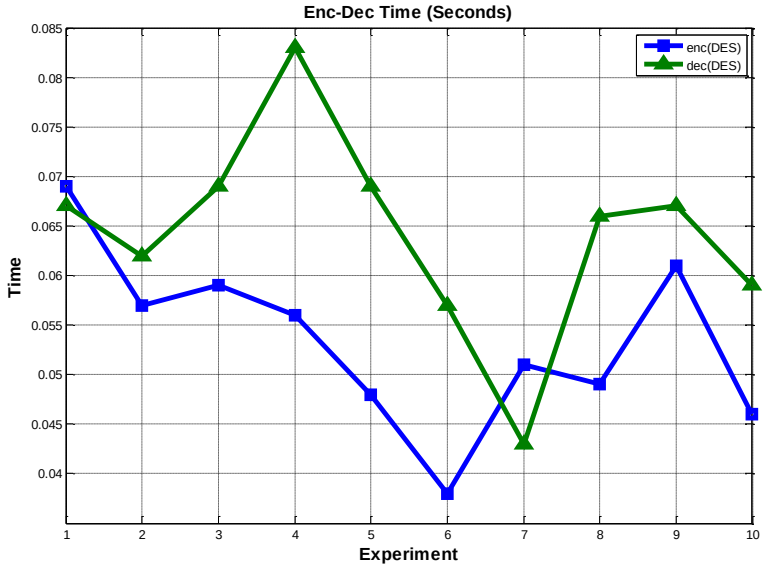

Figure 4.3: Encryption/decryption execution time for DES algorithm

4.4.4 Computational cost for LED algorithm

Table 4.4 lists out the computational cost or the execution time taken for encryption and decryption for LED algorithm. The obtained results for encryption and decryption processes are plotted together with the corresponding results of the proposed SHSED algorithm figure 4.4.

Table 4.4: Execution time for LED algorithm

Experiment #	**LED** Encryption time(Seconds)	**LED** Decryption time(Seconds)
1	0.1240	0.1980
2	0.1210	0.1830
3	0.1230	0.1880
4	0.1230	0.1910
5	0.1210	0.1810
6	0.1230	0.1900
7	0.1200	0.1820
8	0.1190	0.1850
9	0.1230	0.1910
10	0.1230	0.1910
Average	**0.1220**	**0.1880**
Speed up of SHSED	**0.1220/0.0991= 1.2311**	**0.1880/0.0843= 2.2301**

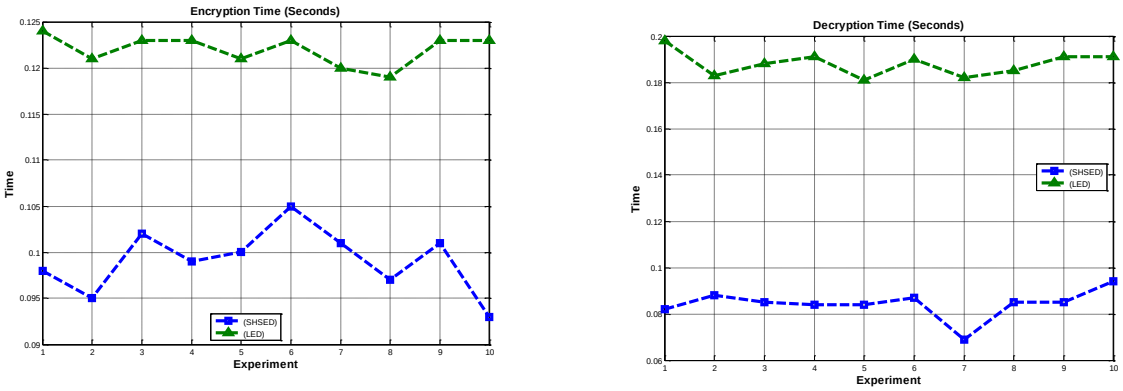

Figure 4.4 : Encryption and decryption execution time comparison for the proposed SHSED algorithm with LED algorithm.

As a result of the computational cost analysis to measure the time of encryption and decryption between LED and SHSED algorithm. It is noted that a significant improvement in the efficiency of the proposed SHSED algorithm as compared with that for light weight algorithms (LED). The speed up gain in the encryption process is 1.2311 times, while the speed up in the decryption process is 2.2301 time.

4.5 Analysis and Comparison of SHSED with other algorithms

The Proposed (SHSED) algorithm achieved lower computational cost in both cases of encryption and decryption when compared with the corresponding computational cost for the other cryptographic system, namely AES, and DES algorithms. Figure 4.5 illustrates the encryption time for these algorithms, and figure 4.6 illustrates the decryption time for the same algorithms under consideration. These figures listed the results for ten runs using the same message block.

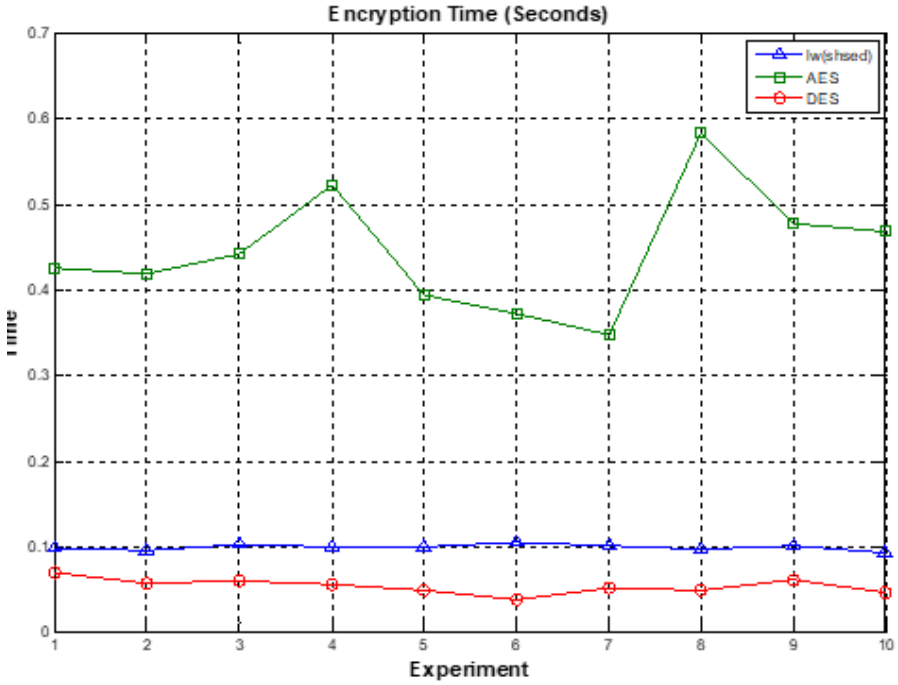

Figure 4.5: Encryption time comparison of SHSED with AES and DES

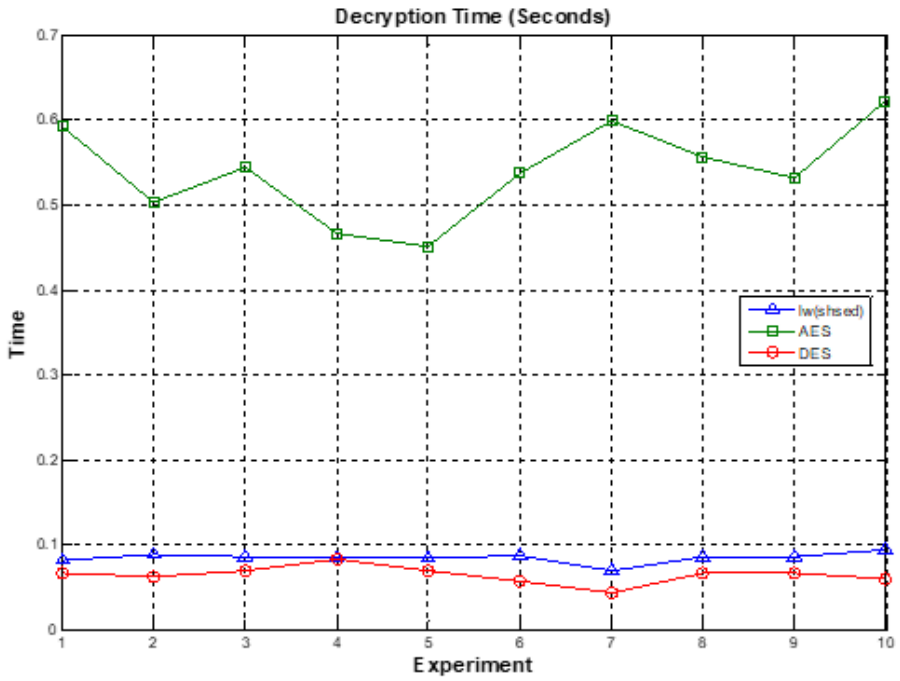

Figure 4.6 : Decryption time comparison of SHSED with AES and DES

It is quite clear that the execution time for SHSED algorithm is much shorter than that for the AES algorithm, but it is slightly longer than those for DES algorithm. This is true for both cases of encryption and decryption processes.

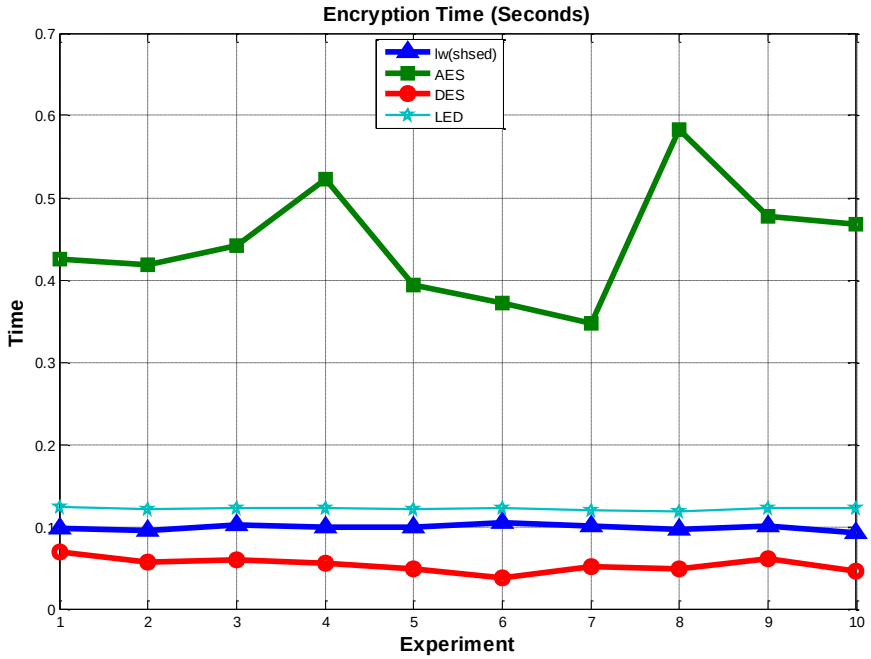

Figure 4.7 : Decryption time comparison of SHSED with AES, DES and LED

When the speed up gains for SHSED algorithm with respect to AES, DES and LED are calculated, they will be as listed in table 4.5. It is shown for both encryption and decryption processes.

Table 4.5: Speed gain comparison for SHSED with respect to AES, DES, and LED

	Speed up	
	Encryption	**Decryption**
AES	**4.4904**	**6.4151**
DES	**0.5388**	**0.7615**
LED	**1.2310**	**2.2301**

4.6 Result analysis of rounds for the Proposed SHSED

The number of rounds for the proposed SHSED algorithm is investigated and the results are listed in Table 4.6, and they are plotted in figure 4.7.

Table 4.6 : rounds computational time for SHSED algorithm

Rounds	Encryption time(Seconds)	Decryption time(seconds)
2	0.0710	0.0258
3	0.0706	0.0265
4	0.0778	0.0277
5	0.0730	0.0316
6	0.0747	0.0328
7	0.0715	0.0288
8	0.0732	0.0307
16	0.0762	0.0362
20	0.0787	0.0385
24	0.0793	0.0453
30	0.0820	0.0441

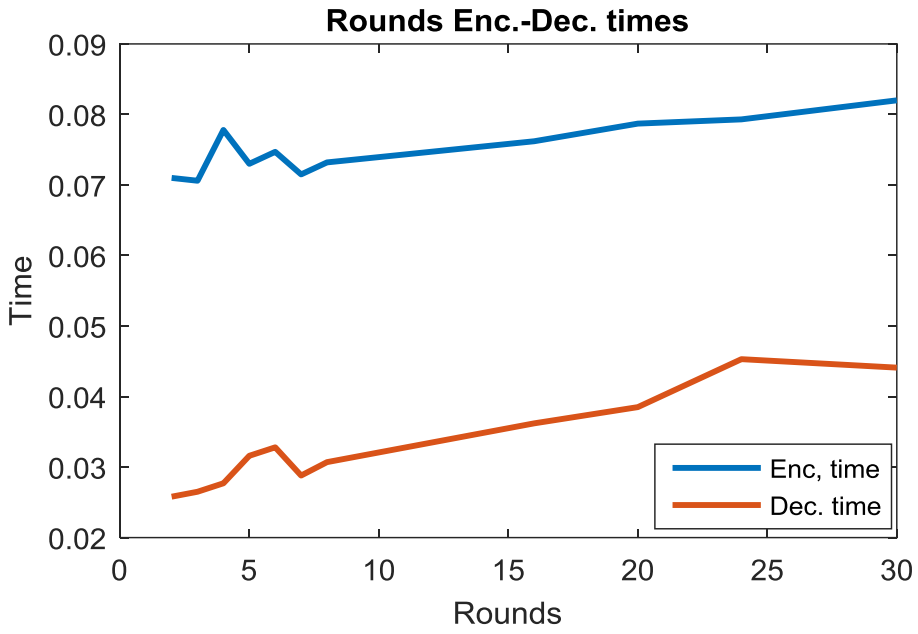

Figure 4.8: Effect of number on encryption and decryption time for SHSED algorithm

4.6.1 Example of number of round on SHSED algorithm

The effect of number of rounds on the resulted ciphertext for the same message is investigated and the results are listed in table 4.7. it list the ciphered message both in hexadecimal representation and as printable characters. It shows too that each round scrambles the message and hence present hardening for the ciphertext.

Table 4.7 : example of SHSED encryption proposed using rounds

Rounds	Plain text	Cipher text (Hex)	Cipher text (characters)
2	awzsexdrcftvgybh	BF96D38EC9657464	¿–ÓŽÉetd
3	awzsexdrcftvgybh	89E053EB67653FBF	‰àSëge?¿
4	awzsexdrcftvgybh	7D607B893C2E3A89	}`{‰<.:‰
5	awzsexdrcftvgybh	7B480A5E082B847D	{H ^+„}
6	awzsexdrcftvgybh	27390A2A72957C7B	'9 *r•|{
7	awzsexdrcftvgybh	54396C94E56DF027	T9l”åmð'
8	awzsexdrcftvgybh	715F900782E15A54	q_ ,áZT
9	awzsexdrcftvgybh	BCA3DEA4954B3B71	¼£Þ¤•K;q
10	awzsexdrcftvgybh	07EDA4B7002A88BC	í¤·
11	awzsexdrcftvgybh	3D97DB229C995907	=—Û"œ™Y
12	awzsexdrcftvgybh	E8E8ACBEC848ED3D	èè¬¾ÈHí=
13	awzsexdrcftvgybh	F19F19EAB8FCECE8	ñŸê¸üìè
14	awzsexdrcftvgybh	682AECDAF9FD05F1	h*ìÚùýñ
15	awzsexdrcftvgybh	47DFC11B72140668	GßÁrh
16	awzsexdrcftvgybh	CEF255948C174847	ÎòU”ŒHG
17	awzsexdrcftvgybh	C56687AE115991CE	Åf‡®Y‘Î
18	awzsexdrcftvgybh	A5B47C33418017C5	¥´|3A€ Å
19	awzsexdrcftvgybh	D44FB46308064EA5	ÔO´cN¥
20	awzsexdrcftvgybh	C987B42AE95F22D4	É‡´*é_"Ô
21	awzsexdrcftvgybh	9587ED0B163322C9	í 3"É
22	awzsexdrcftvgybh	45DE4D38F2336395	EÞM8ò3c•
23	awzsexdrcftvgybh	347EF814BE72B145	4~ø¾r±E
24	awzsexdrcftvgybh	98CBCBE09AA01F34	˜ËËàš 4
25	awzsexdrcftvgybh	97F897BCF90E3898	—ø—¼ù8˜
26	awzsexdrcftvgybh	54A48C1B78296297	T¤Œx)b—
27	awzsexdrcftvgybh	D9BFBC9AA5735754	Ù¿¼š¥sWT
28	awzsexdrcftvgybh	A98FCAC7CE46A4D9	© • ÊÇÎF¤Ù
29	awzsexdrcftvgybh	FFF986F045B5FFA9	ÿù†ðEµÿ©
30	awzsexdrcftvgybh	84B5056747EE9BFF	„µgGî›ÿ
31	awzsexdrcftvgybh	F836F069128A7884	ø6ðiŠx„

4.7 Measurement SHSED algorithm security strength

The security strength of the SHSED algorithm was measured using an avalanche effect equation :

(number of flipped bits in the ciphertext / number of bits in the ciphertext)×100%

Table 4.8 lists out result of avalanche effect with fixed plaintext **(4142434445464748)** and varied key, in the rounds=3.

Table 4.8 : avalanche effect with fixed plaintext

test	key	Ciphertext (Hex)	Number of changed bits	Avalanche effects %
1	ffeeddccbbaa99887766554433221100	89E053EB67653FBF	21	48
2	ff00ddccbbaa9912776655ac33ff1100	9BE0B5CA01FA11F4	34	53
3	1500d7ccb4ac991972665bad33ff11f0	2616B6BC17705499	33	52
4	ab09d7ccafac991c72365bad33af110f	2616B6BC17705499	33	52
5	ac02d9ccb5ac961c72385ba9330f1acf	D2F2AC8BA4BCE3A6	39	61
6	a4c279ccbfac991c72b85ba9330f1a78	1C267EC2D3FDCC96	37	58
7	f4cd795cbf2c9a1c62b84bb9cc0f1a78	62916358451E7A97	27	42
8	f4cd795cbf2cccab62b84bb9cc0f1a21	62916358451E7A97	27	42
9	24cd795cbfccccab6ab84bb95c0f1a2c	F6DDF021A288924B	39	61
10	24cd795cefccccbb6ab84bb95c0f1a7d	8D1717DC8A9808AC	35	55

Table 4.9 lists out the result of avalanche effect with fixed key (**ffeeddccbbaa99887766554433221100**) and varied plaintext, in the rounds=3

Table 4.9 : avalanche effect with fixed key

test	Plaintext (Hex)	Ciphertext (Hex)	Number of changed bits	Avalanche effects %
1	4142434445464748	89E053EB67653FBF	21	48
2	4942434446564748	390661 D2F16D97BC	30	47
3	92325544465647005	B67B478C4A5937AE	24	38
4	a232c544465646261	C3E38A1521F4993E	32	50
5	b232c244d65548241	4CB2AC35015C38F4	31	48
6	a53dc254d65544221	65F652337740B713	29	45
7	f53cc258d65944201	DB0985B0CC897CEE	31	48
8	f63cc258d6a944241	A626DB7EC5865031	31	48
9	af3cc2b8d6b9c42f1	D7C33861C7E4CCBB	35	55
10	a93bb2b8d6b9c42f	D7C3B63AD8002CAB	28	44

Table 4.10 lists out the result of avalanche effect with fixed key

(**ffeeddccbbaa99887766554433221100**) and fixed plaintext, with varied rounds.

Table 4.10 : avalanche effect with fixed plaintext and key

Number of Rounds	Plaintext (Hex)	Ciphertext (Hex)	Number of changed bits	Avalanche effects %
1	4142434445464748	649AA5796C1474E0	31	48
2	4142434445464748	BF96D38EC9657464	30	47
3	4142434445464748	89E053EB67653FBF	29	45
4	4142434445464748	7D607B893C2E3A89	31	48
8	4142434445464748	715F900782E15A54	31	48
16	4142434445464748	CEF255948C174847	31	48
20	4142434445464748	C987B42AE95F22D4	33	52
24	4142434445464748	98CBCBE09AA01F34	33	52
28	4142434445464748	A98FCAC7CE46A4D9	27	42
31	4142434445464748	F836F069128A7884	37	58

4.7.1 Result of avalanche effect test

From the obtained results shown in the previous table we can see that a small change in the key or the plain text or in the number of rounds should be resulted with a significant change in the cipher text, thus indicate the strength of the proposed algorithm.

Chapter Five

Conclusions and Future Work

5.1 Conclusion

A Light weight algorithm simple and highly secure encryption_decryption (SHSED)algorithm can be used for Cloud based applications. The developed SHSED algorithm which is inspired by International data encryption algorithm (IDEA). Experimental results have demonstrated powerful security level and a clear improvement in the encryption/decryption execution times compared with other cryptographic systems widely used in cloud computing.

The speed gain obtained compared with AES and the lightweight LED algorithms is encouraging for a practical application as the efficiency was 4.4 times for encryption and 6.41 times for decryption in the case of AES algorithm. It is also 1.23 times faster than LED for encryption and 2.23 times for decryption in the case of LED. However, it was slightly slower than DES.

5.2 Recommendations and Future Work

For this thesis, the researcher suggested the following recommendations:

1- Implementing SHSED algorithm in hardware may produce much better results and encourage user to apply it in their daily work, therefore it worth researching.

2- A hybrid security system for the cloud computing could be designed that implement SHSED algorithm. As this research work is suggesting a new light weight cryptosystem algorithm, a hybrid scheme for the cloud computing security may be proposed that implement this SHSED algorithm in addition to one of the heavy and highly secure algorithm, such as the AES cryptosystem. The idea is to have a scheme that is light

and with some security in addition to a strong one, then compare the results with other reported hybrid schemes, such as those given in references, and may be explained briefly as follows:

The proposed idea in this proposal is a new scheme that combines (SHSED) a light weight cryptosystem algorithm (modified from IDEA cryptosystem) with heavy and highly secure AES algorithm ,the client divides the data into a sensitive and ordinary data , then applies the new scheme of hybrid encryption for sensitive data including light weight and AES cryptosystem . light weight encryption used to encrypt the second type of data , then merge the data and send to the cloud storage, the process of retrieving data from the cloud storage goes through an inverse processes whereby data is split again ,decryption and merged. The proposed is shown in Figure 5.1 .

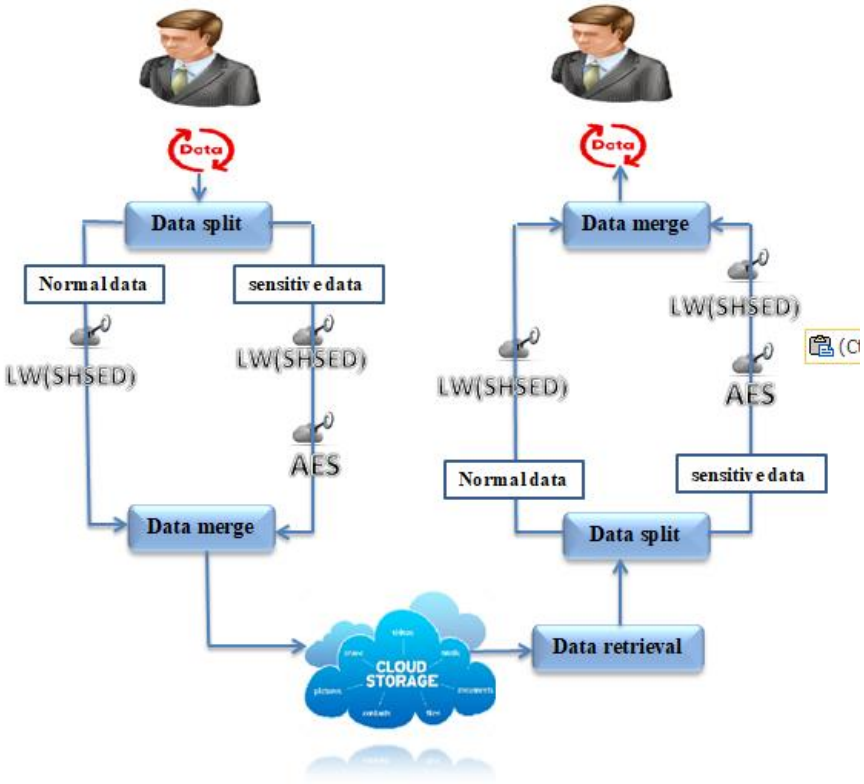

Figure 5.1 : The possible Hybrid scheme using SHSED algorithm

References

Akashdeep Bhardwaj, GVB Subrahmanyam, Vinay Avasthi, Hanumat Sastry, 2016,"Security Algorithms for Cloud Computing", Procedia Computer Science 85 (2016) 535 – 542 , P 535-542 .

Aws Naser Jaber, and Mohamad Fadli Bin Zolkipli, 2013, "Use of Cryptography in Cloud Computing", IEEE International Conference on Control System, Computing and Engineering, 29 Nov. - 1 Dec. 2013, Penang, Malaysia.

A. Antonova, E. Gourova, and N. Roumen, 2011, "Extended architecture of knowledge management system with Web 2.0 technologies." pp. 48-55.

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., ... & Vikkelsoe, C. (2007, September). PRESENT: An ultra-lightweight block cipher. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 450-466). Springer, Berlin, Heidelberg.

Chen, D., & Zhao, H. (2012, March). Data security and privacy protection issues in cloud computing. In Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on (Vol. 1, pp. 647-651). IEEE.

Gholami, A., & Laure, E. (2016). Security and privacy of sensitive data in cloud computing: a survey of recent developments. arXiv preprint arXiv:1601.01498.

Gong Z., S. Nikova, and Y. W. Law, 2012. Klein: A new family of lightweight block ciphers. In Ari Juels and Christof Paar, editors, RFID. Security and Privacy, vol. 7055 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 1–18.

Guo J., T. Peyrin, A. Poschmann, and M. J. B. Robshaw, 2011. The LED block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, Cryptographic Hardware and Embedded Systems CHES 2011, vol. 6917 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 326–341.

Leander, G., Paar, C., Poschmann, A., & Schramm, K. (2007, March). New lightweight DES variants. In International Workshop on Fast Software Encryption (pp. 196-210). Springer, Berlin, Heidelberg.

Hong D., J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee, 2006. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In Cryptographic Hardware and Embedded Systems - CHES 2006, LNCS 4249, Springer-Verlag, pp. 46-59.

Jasleen Kaur , Dr. Sushil Garg , 2015, " Security in Cloud Computing using Hybrid of Algorithms", International Journal of Engineering Research and General Science, Volume 3, Issue 5, P 300-305 .

Jean Raphael Ngnie Sighom, Pin Zhang and Lin You, 2017, " Security Enhancement for Data Migration in the Cloud", Future Internet ,P 1-13.

Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw, 2011, "The LED Block Cipher", L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 50-55, 2008.

Leander G., C. Paar, A. Poschmann, and K. Schramm, 2007. New lightweight DES variants. In the proceedings of Fast Software Encryption - FSE 2007, vol. 4593 of Lecture Notes in Computer Science, Springer-Verlag, pp. 196-210.

Lo'ai Tawalbeh, Nour S. Darwazeh, Raad S. Al-Qassas2 and Fahd AlDosari , 2015, " A Secure Cloud Computing Model based on Data Classification" , Procedia Computer Science 52 (2015) 1153 – 1158 .

Mandeep Kaur, Manish Mahajan, 2013, "Using encryption Algorithms to enhance the Data Security in Cloud Computing", International Journal of Communication and Computer Technologies, Volume 01 – No.12, Issue: 03,P 56-59.

Omer K. Jasim, Safia Abbas, El-Sayed M. El-Horbaty and Abdel-Badeeh M. Salem, 2013, "Efficiency of Modern Encryption Algorithms in Cloud Computing" International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) Volume 2, Issue 6,p 270-274.

Peter Mell and Tim Grance, 2013. "The NIST Definition of Cloud Computing". National Institute of Standards and Technology (NIST), Special Publication 800-145, https://www.tasclinx.com/wp-content/uploads/2013/02/141103-US-SP800-145.pdf

Prerna Mahajan and Abhishek Sachdeva, 2013, " A Study of Encryption Algorithms AES, DES and RSA for Security" Global Journal of Computer Science and Technology Network, Web & Security, Volume 13, Issue 15, ISSN: 0975-4350.

Priya jaiswal, Randeep kaur, Ashok Verma, 2014, "Privacy and Security on Cloud Data Storage Using Hybrid Encryption Technique", International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 1, P 161-164.

Rachna Arora, Anshu Parashar, 2013, "Secure User Data in Cloud Computing Using Encryption Algorithms", International Journal of Engineering Research and Applications (IJERA, Vol. 3, Issue 4, P1922-1926.

Randeep Kaur, Supriya Kinger, 2014, "Analysis of Security Algorithms in Cloud Computing", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 3, Issue 3,P 171-176 .

Rizwana Shaikh and Dr. M. Sasikumar, 2015, " Data Classification for achieving Security in cloud computing", procedia computer science 45(2015) 493-498.

Shakeeba S. Khan, Prof.R.R. Tuteja, 2015, "Security in Cloud Computing using Cryptographic Algorithms", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 1, P 148-154.

Shikha Rani, Shanky Rani, 2016, "Data Security in Cloud Computing Using Various Encryption Techniques", International Journal of Modern Computer Science (IJMCS), Volume 4, Issue 3, P 163-166.

Shilpashree Srinivasamurthy, and David Q. Liu, "Survey on Cloud Computing Security", Proceeding of the 2nd IEEE International Conference on Cloud Computing Technology and Science, Indianapolis, IN, Winter 2010.

Shirai T., K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, 2007. The 128-bit blockcipher cle_a (extended abstract). In Fast Software Encryption - FSE 2007, vol. 4593 of Lecture Notes in Computer Science, Springer-Verlag, pp. 181-195.

Suzaki T., K. Minematsu, S. Morioka, and E. Kobayashi, 2013. TWINE: A lightweight block cipher for multiple platforms. In LarsR. Knudsen and HuapengWu, editors, Selected Areas in Cryptography, vol. 7707 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, pages 339–354.

S.Sandhya, U.Reshma and Dr.V.Praveena, 2017, " Survey on Various Data Encryption Algorithms Used in Cloud Security", International Journal of Innovative Research in Computer and Communication Engineering,Vol. 5, Issue 9.

Vinita Keer, Dr. Syed Imran Ali, Prof. Neeraj Sharma, 2016, "Hybrid Approach of Cryptographic Algorithms in Cloud Computing", International Journal of Emerging Technology and Advanced Engineering , Volume 6, Issue 7, P 87-90.

Wenling Wu and Lei Zhang, 2011. Lblock: A lightweight block cipher. In Javier Lopez and Gene Tsudik, editors, Applied Cryptography and Network Security, vol. 6715 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, pages 327–344.

Xuejia Lai and James L. Massey, 1991, " Markov Ciphers and Differential ryptanalysis", Yibin Li, Keke Gai, Longfei Qiu, Meikang Qiu, Hui Zhao, 2016, "Intelligent cryptography approach for secure distributed bigdata storage in cloud computing", Information Sciences, P 1-13.

Zhang W., Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, 2014. Rectangle: A bit-slice ultra-lightweight block cipher suitable for multiple platforms. Cryptology ePrint Archive, Report 2014/084. http://eprint.iacr.org/.

Appendix A: matlab code

```
function cst=constantGen()
%% Generate constants used for the round
  cst='0101101';
  for i=1:127
    cst(7+i)=dec2bin(bitxor(bin2dec(cst(i+3)), bin2dec(cst(i))), 1);
  end
end
function state=create_state(X, m)
  state=[];
  if m<=0
    for i=1:8:(length(X)-7)
      state=[bin2dec(X(i:i+7)) state];
    end
  else
    j=0;
    for i=1:2:length(X)-1
      j=j+1;
      state=[hex2dec(X(i:i+1)) state];
    end
  end
end
function c=extract_const(cst, i)
```

```
%% Extract the correct constant for the round i from the vector cst generated by the function constantGen
  c='';
  for j=i:i+6
    c=[cst(j+1) c];
  end
end
function x=f0(s) %% Function f0 of the round
  bin=dec2bin(s,8);
  x=bitxor(bitxor(bin2dec(rotl(bin,1)),bin2dec(rotl(bin,2))),
bin2dec(rotl(bin,7)));
end
function x=f1(s) %% Function f1 of the round
  bin=dec2bin(s,8);
  x=bitxor(bitxor(bin2dec(rotl(bin,3)),bin2dec(rotl(bin,4))),
bin2dec(rotl(bin,6)));
end
%%  Final transformations
%%%%
function c=FinalTransformation(x, wk) %% Encryption
  c(1)=mod(x(2)+wk(5), 256);
  c(2)=x(3);
  c(3)=bitxor(x(4), wk(6));
```

```
  c(4)=x(5);
  c(5)=mod(x(6)+wk(7), 256);
  c(6)=x(7);
  c(7)=bitxor(x(8), wk(8));
  c(8)=x(1);
end
function c=FinalTransformation_d(x, wk) %% Decryption
  c(1)=mod(x(1)-wk(1), 256);
  c(2)=x(2);
  c(3)=bitxor(x(3), wk(2));
  c(4)=x(4);
  c(5)=mod(x(5)-wk(3), 256);
  c(6)=x(6);
  c(7)=bitxor(x(7), wk(4));
  c(8)=x(8);
end
%%%%
%%  Initial transformations
%%%%
function x=initialTransformation(P, wk)
  x(1)=mod(P(1)+wk(1), 256);
  x(2)=P(2);
  x(3)=bitxor(P(3), wk(2));
```

```
  x(4)=P(4);
  x(5)=mod(P(5)+wk(3), 256);
  x(6)=P(6);
  x(7)=bitxor(P(7), wk(4));
  x(8)=P(8);
end
function x=initialTransformation_d(P, wk)
  x(2)=mod(P(1)-wk(5), 256);
  x(3)=P(2);
  x(4)=bitxor(P(3), wk(6));
  x(5)=P(4);
  x(6)=mod(P(5)-wk(7), 256);
  x(7)=P(6);
  x(8)=bitxor(P(7), wk(8));
  x(1)=P(8);
end
%%%%
%% Luban algorithm
%%%%
clear all;clc


%%  Example of test vectors
%%%%
```

```
%MSG='0000000000000000';
%Key='00112233445566778899aabbccddeeff';
MSG='4142434445464748';
Key='ffeeddccbbaa99887766554433221100';
NofR=2;

%%%%

tic
C=Lubna_enc(MSG, Key,NofR);
encryptiontime=toc;
%%%%

tic
M=Lubna_dec(C, Key,NofR);
decryptiontime=toc;
MSG
C
 M
 encryptiontime
 decryptiontime
function M=Lubna_dec(C, k,nOFr) %% Decryption
  %% Initialization
```

```
  state_m=create_state(C, 1)
  state_k=create_state(k, 1);
  wk=workKey(state_k);
  cst=constantGen();
  sk=subkey(state_k, cst);
  state_m=initialTransformation_d(state_m, wk)
  %% Rounds
  for i=0:nOFr
       state_m=Round_d(state_m, sk,nOFr-i)
  end
  %% Final transformation
  M_tmp=FinalTransformation_d(state_m, wk)
  %% reformat message
  M_tmp=dec2hex(M_tmp);
  M='';
  for i=1:length(M_tmp)
    M=[M_tmp(i,:) M];
  end
end
%%  Cipher algorithm
%%%%

function C=Lubna_enc(m, k,nOFr) %% Encryption
```

```
  %% Initialization
  state_m=create_state(m, 1)
  state_k=create_state(k, 1);
  wk=workKey(state_k)
  cst=constantGen();
  sk=subkey(state_k, cst);
  state_m=initialTransformation(state_m, wk);
  %% Rounds
  for i=0:nOFr
    state_m=Round(state_m, sk, i);
  end
  %% Final transformation
  C_tmp=FinalTransformation(state_m, wk);
  %% reformat cipher
  C_tmp=dec2hex(C_tmp);
  C='';
  for i=1:length(C_tmp)
    C=[C_tmp(i,:) C];
  end
end


function res=rotl(bin, d) %% Left rotation of d bits
```

```
  if d>1
    res=rotl([bin(2:end) bin(1)], d-1);
  else
    res=[bin(2:end) bin(1)];
  end
end

function x=Round(state, sk, i) %% Encryption
  x(1)=bitxor(state(8), mod(f0(state(7))+sk(4*i+4), 256));
  x(2)=state(1);
  x(3)=mod(state(2)+bitxor(f1(state(1)),sk(4*i+1)), 256);
  x(4)=state(3);
  x(5)=bitxor(state(4), mod(f0(state(3))+sk(4*i+2), 256));
  x(6)=state(5);
  x(7)=mod(state(6)+bitxor(f1(state(5)),sk(4*i+3)), 256);
  x(8)=state(7);
end
function x=Round_d(state, sk, i) %% Decryption
  sk_3=sk(4*i+4)
  x(8)=bitxor(state(1), mod(f0(state(8))+sk(4*i+4), 256));
  x(1)=state(2);
  sk_0=sk(4*i+1)
  x(2)=mod(state(3)-bitxor(f1(state(2)),sk(4*i+1)), 256);
```

```
  x(3)=state(4);
  sk_1=sk(4*i+2)
  x(4)=bitxor(state(5), mod(f0(state(4))+sk(4*i+2), 256));
  x(5)=state(6);
  sk_2=sk(4*i+3)
  x(6)=mod(state(7)-bitxor(f1(state(6)),sk(4*i+3)), 256);
  x(7)=state(8);
end
function sk=subkey(K, cst)
%% Generated the subkey used during the round
  for i=0:7
    for j=0:7
      c=extract_const(cst, 16*i+j);
      sk(16*i+j+1)=mod(K(mod(j-i, 8)+1)+ bin2dec(c), 256);
      c=extract_const(cst, 16*i+j+8);
      sk(16*i+j+9)=mod(K(mod(j-i, 8)+9)+ bin2dec(c), 256);
    end
  end
end
%%%%
%% Keychedule
%%%%
```

```
function wk=workKey(K)
%% create the work key that is used before the first round and after the last round
  for i=1:8
    if i<=4
      wk(i)=K(i+12);
    else
      wk(i)=K(i-4);
    end
  end
end
```