جامعة آل البيت

Al al-Bayt University

AMELIORATED RC6 ALGORITHM FOR CRYPTOGRAPHIC APPLICATIONS

تحسين خوارزمية RC6 في تطبيقات التشفير

by

Hebah Ali Sanasleh

هبه علي السناسله

Supervisor:

Dr. Khaled Mohammed Batiha

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science

Deanship of Graduate Studies

Al al-Bayt University

2018

نموذج رقم (1)

جامعة آل البيت

عمادة الدراسات العليا

نموذج تفويض

أنا هبه علي السناسله

افوض جامعة آل البيت بتزويد نُسخ من رسالتي ، للمكتبات أو المؤسسات أو الهيئات أو الأشخاص عند طلبهم حسب التعليمات النافذة في الجامعة.

التوقيع: ..... التاريخ:

نموذج رقم (2)

جامعة آل البيت

عمادة الدراسات العليا

نموذج اقرار والتزام بقوانين جامعة آل البيت وانظمتها وتعليماتها لطلبة الماجستير والدكتوراه.

أنا هبه علي سليمان السناسله الرقم الجامعي: 1320901024

تخصص: علم حاسوب كلية: تكنولوجيا المعلومات

أُعلنُ بأني قد التزمت بقوانين جامعة آل البيت وانظمتها وتعليماتها وقراراتها السارية المفعول المتعلقة بإعداد رسائل الماجستير والدكتوراه عندما قمت شخصياً بإعداد رسالتي بعنوان:

AMELIORATED RC6 ALGORITHM FOR CRYPTOGRAGHIC APPLICATIONS

خوارزمية RC6 المحسنة في تطبيقات التشفير

وذلك بما ينسجم مع الأمانة العلمية المتعارف عليها في كتابة الرسائل والأطاريح العلمية. كما أنني أُعلن بأن رسالتي هذه غير منقولة أو مستلة من رسائل أو أطاريح أو كتب أو أبحاث أو أي منشورات علمية تم نشرها أو تخزينها في أي وسيلة اعلامية، وتأسيساً على ما تقدم فأنني اتحمل المسؤولية بأنواعها كافة فيما لو تبين غير ذلك بما فيه حق مجلس العمداء في جامعة آل البيت بإلغاء قرار منحي الدرجة العلمية التي حصلت عليها وسحب شهادة التخرّج مني بعد صدورها دون أن يكون لي الحق في التظلم أو الاعتراض أو الطعن بأي صورة كانت في القرار الصادر عن مجلس العمداء بهذا الصدد.

التوقيع ..... التاريخ:

Committee Decision

This Thesis (AMELIORATED RC6 ALGORITHM FOR CRYPTOGRAGHIC APPLICATIONS) was Successfully Defended and Approved on 3rd may. 2018.

Examination Committee	Signature
Dr. Khaled Mohammad Batiha , (Supervisor) Associate professor Dep. of Computer Science, Al al-Bayt university batihakhalid@aabu.edu.jo	.....
Dr. Akram Aref Hamarshi , (Member) Associate professor Dep. of Computer Science, Al al-Bayt university hamarshi@aabu.edu.jo	.....
Dr. Mohammad El-Bashir , (Member) Assistant professor Dep. of Computer Science ,Al al-Bayt university mohdelb@aabu.edu.jo	.....
Dr. Mohammad AL-Batah ,(External Member) Associate professor Dep. of Computer Science & Software Engineering, Jadara university albatah@jadara.edu.jo	.....

Dedication

This thesis is dedicated to my mother who have supported me all the way since the beginning of my studies. Also, I dedicate this dissertation to my family and to all my friends. Without their love, affection, motivation and support this thesis would not have been possible. I am grateful for having the opportunity to work with my supervisor Dr. Khaled Mohammad Batiha.

Acknowledgments

"All praises and thanks to ALLAH for his blessings at every stage of my life "

Firstly, I would like to thank my supervisor Dr. Khaled Mohammad Batiha, for his sincere advice and guidance provided throughout my research and thesis preparation. Special thanks to my family; my parents, my brothers and sisters for their encouragement.

Finally, I would like to give my thanks to all my faculties of Computer Science department of Al al-Bayt University for their kind assistance, and for all my friends for their constant support.

Table of Contents

List of Tables

List of Figures

AMELIORATED RC6 ALGORITHM FOR CRYPTOGRAPHIC APPLICATIONS

A Master Thesis By

Hebah Ali Sanasleh

Supervisor:

Dr. Khaled Mohammed Batiha

Department of Computer Science , Al al-Bayt University, 2017

Abstract

Security in data communication is a very important concern today. Cryptography is the art of converting normal text to cipher text in order to protect the data. RC6 are widely used a cryptographic algorithm for data security. RC6 is a symmetric encryption algorithm designed to meet the requirements of the Advanced Encryption Standard. We have proposed the Ameliorated RC6 algorithm of RC6. Ameliorated RC6 is an enhanced extension of RC6 with increasing the throughput and improving the performance. The proposed algorithm includes two modifications the first is using 2048-bits encryption/decryption block size of data per round instead of 128 bits in RC6, and using of sixty-four working registers for storing plain text/cipher text instead of four registers in RC6 that helps to increase the efficiency and improve security. And the second adding S-Box which does not use in the previous RC6 algorithm. The key size and number of rounds used make Ameliorated RC6 more secure than RC6. The proposed algorithm is resistant to matching and a dictionary attack which increases the security of the previous 128 bits RC6 algorithm by using a block size of 2048-bits instead of 128-bits,

and by using 32 bytes key size with S-Box which does not use in the previous RC6 algorithm. We are expecting to have more efficiency, security, and throughput. Both the algorithms RC6 and Ameliorated RC6 consist of three parts key generation, encryption, and decryption. The basic operations used in the two algorithms are same. Comparative performance evaluation of RC6 and Ameliorated RC6 is introduced. Measuring factors of algorithms are the encryption time decryption time, throughput of encryption, throughput of decryption and security. The experiment results show that Ameliorated RC6 algorithm achieves maximum throughput. Throughput value of Ameliorated RC6 algorithm is 63.8 Mb/sec, while throughput value of RC6 algorithm is 20.19 Mb/sec. So, the Ameliorated RC6 algorithm is satisfy market demands and system security developer goals using advanced processors available.

Keywords – Security, Cryptography, Symmetric Algorithms, AES, RC6, and Ameliorated RC6.

Abstract (Arabic)

تحسين خوارزمية في تطبيقات التشفيرRC6

رسالة ماجستير قُدمت من قبل

هبه علي السناسله

المشرف:

د. خالد محمد بطيحة

قسم علم الحاسوب، جامعة آل البيت، 2017م

ملخص

الأمنية في مجال اتصالات البيانات هو مصدر قلق بالغ الأهمية اليوم. التشفير هو علم تحويل النص العادي إلى النص المشفر من أجل حماية البيانات. تستخدم خوارزمية التشفير RC6 على نطاق واسع من اجل أمنية البيانات. خوارزمية التشفير RC6 هي خوارزمية التشفير الكتلي المتماثل تم تصميمها لتلبية متطلبات خوارزمية معايير التشفير المتقدم AES. وقد اقترحنا تحسين لخوارزمية RC6 وهو Ameliorated RC6 Ameliorated RC6 هو امتداد تحسين RC6 مع زيادة الإنتاجية وتحسين الأداء. الخوارزمية المقترحة تتضمن تعديلات اثنين الأول هو مضاعفة حجم الكتلة للتشفير/ فك التشفير 128 bits RC6 السابقة إلى 2048 bits Ameliorated RC6 واستخدام أربعة وستون سجل عمل لتخزين النص الأصلي/ النص المشفر بدلا من أربعة مما يساعد على زيادة الكفاءة و تحسين الأمنية. والثاني هو إضافة S-Box والذي لم يستخدم سابقا في RC6. حجم المفتاح وعدد الجولات المستخدمة يجعل Ameliorated RC6 أكثر أمنا من RC6. الخوارزمية المقترحة مقاومة إلى هجوم matching و dictionary مما يزيد من الأمنية للخوارزمية السابقة بواسطة استخدام حجم كتلة 2048 bits بدلا من 128 bits

بالإضافة إلى استخدام حجم المفتاح 32 bytes مع S-Box لم يكن مستخدم سابقا. نتوقع أن يكون لدينا المزيد من الكفاءة والأمنية والإنتاجية. تتكون كلتا الخوارزميتين Ameliorated RC6 و RC6 من ثلاثة أجزاء رئيسية هي توليد المفتاح والتشفير وفك التشفير. العمليات الأساسية المستخدمة في الخوارزميتين هي نفسها. قدم في هذه الرسالة تقييم مقارنة لأداء Ameliorated RC6 و RC6. نقارن هذه الخوارزميات على أساس عوامل قياس زمن التشفير وفك التشفير على أساس حجم ملفات مختلفة، وإنتاجية التشفير وفك التشفير والأمنية. وتظهر دراسة المقارنة تفوق خوارزمية Ameliorated RC6 . أظهرت نتائج تقييم الأداء أن Ameliorated RC6 يحقق أقصى إنتاجية. القيمة الإنتاجية للخوارزمية Ameliorated RC6 هي 63.8 ميغابايت / ثانية، في حين أن القيمة الإنتاجية للخوارزمية RC6 هي 20.19 ميغابايت / ثانية. لذا ، الخوارزمية المقترحة تلبي متطلبات السوق وأهداف مطوري أمنية النظام باستخدام المعالجات المتقدمة المتاحة.

الكلمات الأساسية – الأمنية، التشفير، الخوارزميات المتماثلة، AES، AMELIORATED RC6، RC6.

Chapter 1

Introduction

1.1 Background

One of the techniques used to secure the transformed data is cryptography. Due to the great security advantages of cryptography, it is widely used today. Cryptography is a tool used to protect the information in computer systems. The word cryptography is a Greek word it is derived from the Greek words: “kryptós” meaning "hidden" and “gráphein” meaning "to write" or "hidden writing" (Stallings, 2017). Cryptography refers to the art of maintaining the secrecy of data by converting information from its normal form into an unreadable form (Stallings, 2017). The types of cryptography algorithms are symmetric and asymmetric. The most important type of the cryptography algorithms is the symmetric cryptography algorithms. Symmetric key algorithms are dividing into two types: Block cipher and stream cipher. RC6 is a symmetric key block cipher algorithm derived from RC5. It was designed by Ron Rivest, Matt Rob Shaw, Ray Sidney, and Yiqun Lisa Yin to meet the requirements of the Advanced Encryption Standard (AES) competition. RC6 algorithm was selected among the other finalists to become the new federal AES (Nechvatal et al. , 2002)

1.2 Basic Elements of the cryptography operation

A cryptographic algorithm is a set of mathematical function and rules that takes plaintext and a key as input, and produce cipher text as output. Cryptography process consists of two phases: Encryption (convert data to an unintelligible form, called cipher text) and decryption (convert data back to its original form, called plaintext). (Manpreet et al. , 2017).

1.3 Symmetric Key Cryptography

The important type of the cryptography is the symmetric key cryptography. The process of encryption and decryption of data by using a shared key is known as secret key cryptography or symmetric key cryptography. Two ciphers modes are adopted by symmetric key cryptography algorithms: Block ciphers and stream ciphers. (Nikita et al. , 2014).

1.3.1 Stream ciphers

The stream cipher is the method where each bit of data is sequentially encrypted using one bit of the key at a time. The main algorithm in this type is RC4. A stream cipher takes as input the secret key and a parameter called the initialization vector (IV) and outputs a stream of bits called the key stream. The key stream is XORed with the plaintext to produce the cipher text. (Manpreet et al. , 2017).

1.3.2 Block ciphers

In block ciphers, an input is caught as a number of bits and encrypt them as a single unit at a time, padding the plaintext so that it is a multiple of the block size. This specification will identify how much data should be encrypted on each block and also the size of the key is applied to each block. The encryption function is the same for every block. The main algorithms in which type are AES,DES, and Blowfish

.1.4 Advanced Encryption standard

In cryptography, AES algorithm developed by Joan Daemen and Vincent Rijmen, are one of the most significant algorithms used in symmetric key cryptographic algorithms. AES is a cryptographic algorithm which is used universally by the cryptographic community. It is a symmetric block cipher algorithm,

announced by National Institute of Standards and Technology (NIST) on November 26, 2001, Replacing the DES and triple-DES, which designs computation of fifteen algorithms, In this event, Rijndael was chosen as the AES algorithm. The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. Rijndael is fast and compact cipher. It's symmetric and parallel structure provides great flexibility for implementers, with effective resistance against cryptanalytic attacks. In terms of security, no attack has been found for AES. (Nechvatal et al. , 2000).

1.4.1 AES Finalist algorithms

The NIST announced a program to develop and choose an AES algorithm to replace the data Encryption standard (DES). Fifteen algorithms were submitted to NIST in 1998, and NIST chose five finalists in 1999. Finalist algorithms of AES competition by NIST are, five algorithms they are: Rijndael, MARS, RC6, Serpent, and Two fish. NIST selection criteria Focused on the security, cost, implementation, flexibility, and performance in their evaluation of five finalist

algorithms. On the basis of evaluation criteria of AES finalist algorithms, NIST choose the Rijndael algorithm as the AES. (Tharun et al. , 2015) RC6

The RC6 algorithm is a fully parameterized block cipher, based on the RC5 algorithm with better performance and security. It works on 128 bits block size, 128,192,256 bit key length, and the number of rounds is 20. As RC6 operates on the principle of RC that can maintain an extensive range of key sizes, word-lengths and number of rounds, RC6 does not comprise S-boxes and same algorithm is used in reverse for decryption. RC6 is the fastest AES finalist. The RC6 algorithm is a good solution and applies well to most platforms.

Rijndael

The Rijndael algorithm is a symmetric block cipher uses matrix structure rather than a feistel structure, where the mixing involves byte substitution, row shifting, and column multiplication substitution has the most different structure when compared with the other AES finalists. AES encryption is fast and flexible. (Harsh et al. , 2012)

MARS

The MARS algorithm is a symmetric block cipher based on generalized feistel network, it works with 128-bits block size, and the key length ranging from 128 to 400 bits. MARS was designed to avoid potential future attacks, especially in its heterogeneous structure, a keyed core surrounded by unkeyed forwards and backwards mixing functions. (NEETA et al. , 2013)Serpent

The serpent algorithm is a symmetric block cipher algorithm based on a conservative design tries to build on the vast amount of DES. It works on 128 bits block size, 256 bits key length, and the number of rounds is 32. It is faster than the DES algorithm and more secure than the 3DES algorithm. The serpent is the slowest of the five AES finalists on most platforms. (NEETA et al. , 2013)

Two fish

The two fish algorithm is a symmetric block cipher algorithm based on Feistel structure. It works on 128 bits block size, 256 bits key length, and the number of rounds is 16. Two fish is a byte-oriented feistel cipher with great flexibility of implementation, allowing a wide range of time/space.(Harsh et al. , 20121.4.2 Evaluation Criteria and Final Score of AES Finalist Algorithms

Finalist candidate algorithms of AES competition program arranged by NIST in 1997 are, five algorithms they are Rijndael, MARS, RC6, Serpent, and Twofish. Rijndael selected as the AES algorithm. NIST focused their evaluation of each algorithm based on the following criteria: security (the most important factor in the evaluation), cost, and algorithm and implementation characteristics, table 1-1 shows final score of AES finalist algorithms. From the following table, we can see that Rijndael got the most numbers of votes and selected as the AES. (Tharun et al. , 2015).

Rijndael algorithm better than RC6 algorithm then, why we choose RC6 in this thesis?

We chose RC6 algorithm because it's characteristics; a secure and simple block cipher and offers good performance and considerable flexibility. So, it is a good choice of encryption algorithm. And also it has the advantage of having a data block size, a number of rounds and key size variables. This provides the opportunity for great flexibility in both performance characteristics and the level of security, so RC6 algorithm is still in progress and is periodically updated to reflect any additional findings. Furthermore its simplicity will allow analysts to quickly refine and improve our estimates of its security.

Table 1-1: Final Score of AES Finalist Algorithms.

Criteria	Rijndael	Serpent	Twofish	Mars	RC6
General Security	2	3	3	3	2
Implementation Difficulty	3	3	2	1	1
Software Performance	3	1	1	2	2

Smart Card Performance	3	3	2	1	1
Hardware Performance	3	3	2	1	2
Design Features	2	1	3	2	1
Total	16	14	13	10	09

1.4.3 Comparison between different AES Finalist Algorithms

Table 1-2 show comparison between different AES finalist algorithms based on the architecture of these algorithms. (Tharun et al. , 2015).Table 1-2: Architectural comparison of different AES finalist algorithms.

Algorithm	Type of structure	Key Length	Block Size With the number of rounds	S-Boxes
Rijndael	Feistel Structure	Variable 128, 192 or 256 bits	128 bit With variable 10, 12 or 14 rounds	No
Twofish	Fiestel structure	Variable 128, 192 or 256 bits	128 bit with 16 rounds	Four
Serpent	Substitution permutation network structure	Variable 128, 192 or 256 bits	128 bit with 32 rounds	Eight

MARS	Heterogeneous structure	Variable 128 to 448 bits in multiples of 32-bit	128 bit with 32 rounds	One
RC6	Substitution permutation network structure	Variable 128, 192 or 256 bits	128 bit with 20 rounds	No

1.5 problem Statement

In this thesis I come to know that what is the problem of previous research or we can say what point was missed by them, for example the value of throughput is low. Here I am trying to overcome that drawback and make the RC6 algorithm achieves maximum throughput and strong resistance against the cryptanalysis attacks.

1.6 Motivation

The proposed Ameliorated RC6 algorithm expected to increase security by using 32 bytes instead of 16 bytes key size with adding the S-Box with key generation algorithm and throughput and improving performance by using of sixty-four working registers instead of four working registers in RC6, and by using a block size of 2048 bits instead of 128 bits. Many enhancements have been introduced in this algorithm. However, these enhancements can still be developed in order to maximize throughput and increase security.

1.7 Goals of the study

The main goal of this study is to maximize throughput and increase security to achieve the best performance. In addition, the system designed in this study compares between the original RC6 and Ameliorated RC6 algorithms. This study aims to enhance the security of RC6 as well as by adding the S-Box, by using of sixty-four working registers instead of four registers in RC6, and by using a block size of 2048 bits instead of 128 bits.

1.8 Thesis Outlines

The rest of the thesis is organized as follows:

Chapter 2 explores the previous studies on this subject and explains both strength and weakness points in each one of them.

Chapter 3 clarifies the original algorithm the RC6 symmetric cipher algorithm.

Chapter 4 clarifies the proposed algorithm in this thesis as well as the system design.

Chapter 5 presents the results of the study and compares them with the previous studies.

Chapter 6 introduces the conclusion of the thesis and the possible future improvements over it.

Chapter 2

iterature Review

2.1 Previous Studies

In this chapter, the information related to the RC6 algorithm was described by referring to variety of literature through a different type of resources. It provides a description of the related work of the RC6 algorithm.

A Modification of RC6 block cipher algorithm for data security (MRC6)

In (Nawal A. El-Fishawy, et al., 2004), they proposed a Modified RC6 (MRC6). MRC6 is 512 bits block size instead of 128 bits in RC6. MRC6 offers a simple, compact, and flexible block cipher. MRC6 has much faster diffusion than all others versions of the RC6 algorithm, Since that using integer multiplication to compute rotation amounts. Simulation results show that MRC6 achieves minimum encryption/decryption time and maximum throughput compared with the RC6 algorithm.

An improved RC6 algorithm with the same structure of encryption and decryption

In (Gil-Ho Kim, et al., 2009), they proposed an improved RC6 algorithm that has the same algorithm for encryption and decryption by inserting symmetric layer using simple rotation and logical operation. Based on that study, the proposed algorithm has no difference with the original RC6 algorithm in the speed. However it has an advantage, it has improving security because a differential and linear analysis attack has difficulty in analyzing cipher texts. On other hands, it has disadvantage it is area increases twice compared with the original structure.

A new version of the RC6 algorithm, stronger against χ2 cryptanalysis

In (Routo Terada Eduardo T. Ueda, 2009), they proposed a Modified version of RC6 is called RC6T. RC6T is RC6 with the addition of simple data dependent swapping function. This function called T() function consists of exchanging the two halves of 32-bit block if the Hamming weight of the block is odd. The only difference to the original RC6 is the addition of B=T(B) and D=T(D) inside the main loop. Statistical experiments results show that the RC6T is stronger against the X2 cryptanalysis attack than the original RC6 algorithm.

A Proposed 512 bits RC6 Encryption Algorithm

In (Ashwaq T. Hashim, et al., 2010), they proposed 512 bits RC6 algorithm includes doubling 128 bits RC6 to 256 bits and adapting a Feistal network. The structure of 512 bits RC6 algorithm consists of splitting the block size into two 256 bits halves. In 512 bits RC6 the number of bits per word is doubled to 64 bits instead of 32 bits in the original 128 bits RC6. The proposed 512 bits RC6 algorithm is a secure, simple block cipher, and resistant to matching and dictionary attack. The results show that the proposed algorithm increases the security when compared to the RC6 algorithm.

Proposed Cascaded Design of 640-bit RC6 Block CipherIn (Ashwaq T. Hashim, and Dr.Yossra H. Ali, 2010), they proposed RC6-cascade. RC6-cascade is a 640 bits RC6 block cipher algorithm. The block size is 640 bits instead of 128 bits, which is subdivided into five parts p1,p2,p3,p4,p5 each of which is 128 bits. The F-Function in RC6 cascade will be used cascaded design instead of rounds. The results show that the RC6-cascade algorithm is a fast and secure symmetric key algorithm. The average of the avalanche effect of RC6-cascade is 324, compared to RC6 is 61.77. The RC6-cascade achieved the best result on images of binary data.

Measurement of Encryption Quality of Bitmap Images with RC6, and two modified version Block Cipher

In (Ashwaq et al. , 2010), they analyzed RC6, modified version 512-bit RC6 and 640-bit RC6-Cascade algorithms, to investigate the encryption efficiency for them to digital images and providing a new mathematical measure of encryption efficiency. This study inspection three encryption algorithms RC6, 512 bit RC6, and 640 bit RC6-Cascade on encrypting images of different constructions. Comparative study is based on the measuring quality factors to evaluate and compare the three encryption algorithms. These measuring factors are the maximum deviation, the correlation coefficient, and irregular deviation. The results show that RC6-Cascade achieved the best result on images of binary data.

Enhancement of RC6 (RC6_EN) block cipher algorithm and comparison with RC5 & RC6

In (Vikas et al. , 2012), they proposed an enhanced version of the RC6 algorithm(RC6_EN), which is 256 bits block size, (128,129,256) bits key size and 20 numbers of rounds. RC6_EN uses eight registers for storing plaintext and data-dependent rotations. It uses data-dependent rotations, modular addition, and XOR operations. RC6_EN uses two box-type operations; Box-Type I, Box-Type II, to improves diffusion in each round. This version of RC6 algorithm performs better when file size is large. The results show that the RC6_EN is a fast, high secure and it offers good performance.

Analyzing the performance of RC6 using Complex Vedic Multiplier

In (Thenmozhi et al. , 2013) they proposed the RC6 Algorithm using Complex Vedic Multiplier, which is the multipliers are replaced by Vedic Sutras. These Vedic Sutras helps to reduce the partial products to improve the performance and efficiency of RC6 structure. The performance of RC6 Algorithm improved using Vedic Sutras in Multipliers such as “Urdhvat-tiryakbyham” and “Nikhilam Navatascaramam Dasatah”, which are used to increase the efficiency of RC6. so, the security level of RC6 Algorithm is greatly improved using these sutras. The results of performance study show that the RC6 Algorithm with Vedic Sutras are reduced the partial products of multipliers in RC6 structure, so the efficiency of RC6 Structure is greatly improved.

A new modified RC6 algorithm for cryptographic applications

In (Sritha et al. , 2014), they proposed an improved RC6 algorithm which is also is 128 bits block size with an advanced symmetric layer structure, consists of fixed rotate operation and XOR&AND operations. The half of RC6 round uses encryption process and the rest of it uses decryption process, and the symmetric layer has been put into the middle of the whole rounds of the algorithm which made different algorithm of encryption and decryption. has beenimplemented to have the same algorithm. Therefore the performance of RC6 algorithm has been improved. The results show that the proposed algorithm when use symmetric layer, the security of the core system is improved and the speed increase more than the original RC6 algorithm by 7%.

Design and Implementation of Enhanced version of MRC6 algorithm for data security

In (Nanda et al. , 2015) they proposed Enhanced Modified version of the RC6 algorithm (EMRC6) include the use of thirty-two working registers instead of four. it works on 1024 block size and has 18 number of rounds for the performance of the system. EMRC6 uses the inclusion of integer multiplication as an additional primitive operation. The use of multiplication increases the diffusion achieved per round,

allowing greater security and increased throughput. The results show that the EMRC6 is the best version of the RC6 algorithm. It has more security, and throughput compared to RC6, RC6_EN, and MRC6 algorithms. Throughput value of EMRC6 algorithm is 34.13 MB/sec.

Comparison of RC6, Modified RC6 & Enhancement of RC6

In (Kirti Aggarwal, 2015), they provide details of the RC6 and two advancements of RC6 called as Modified RC6 (MRC6) and Enhancement of RC6 (RC6_EN). Then these algorithms are compared on the basis of parameters used by the algorithms and their execution time and throughput of encryption and decryption on the basis of different file size. Comparative results show that the throughput of RC6_EN is greater than RC6 and the throughput of MRC6 is greater than RC6_EN & RC6. From the results, we can say that MRC6 provides the better result in terms of execution time and throughput.

We summarize previous studies in the following table 2-1 and proposed study in the following table 2-2.

Table 2-1: The Comparison between RC6_EN, MRC6, and EMRC6.

algorithm	MRC6	RC6_EN	EMRC6
Research name	MODIFICATION OF RC6 BLOCK CIPHER	ENHANCEMENT OF RC6 (RC6_EN) BLOCK CIPHER ALGORITHM AND	Design and Implementation of Enhanced version of
Researchers name	Nawal A. El-Fishawy, Talat E. El-	Vikas Tyagi, Shrinivas Singh	Nanda Hanamant Khanapur, and Arun
Year	2004	2012	2015

Features	512-bit block size using 16 registers for storing plain text and cipher text	256-bit block size using 8 registers for storing plain text and cipher text	1024-bit block size using 32 registers for storing plain text and
Advantages	MRC6 achieves minimum Encryption and decryption time and maximum	RC6_En is a fast and secure block ciphering algorithm.	EMRC6 is greater security, fewer rounds, and increased throughput.
Disadvantages	Does not use look-up tables during encryption	This enhanced performs better only when file size is larger, and does not use	Does not use look-up tables during encryption
Throughput		17.3 Mb/sec	

Table 2-2: The Comparison between RC6, and Ameliorated RC6.

algorithm	RC6	Ameliorated RC6
Research name	The RC6 Block Cipher	AMELIORATED RC6 ALGORITHM FOR CRYPTOGRAGHIC
Researchers name	Ronald L. Rivest, M.J.B. Robshaw, R. Sidney	Dr. Khaled Batiha, and Hebah Sanasleh

Features	128-bit block size using 4 registers for storing plain text and cipher text	2048-bit block size using 64 registers for storing plain text and cipher text, but uses s-box during encryption, which does not used in other algorithms
Advantages	RC6 is a secure, compact and simple block cipher. It offers good performance and considerable flexibility.	Ameliorated RC6 using 2048-bit block size with 64 working registers contributes to maximizing Throughput of the algorithm, using s-box contributes to increasing the
Disadvantages	null	Time to encryption and decryption increases compared with the rc6
Throughput	20.19 Mb/sec	63.8 Mb/sec

Chapter Three

THE RC6 Symmetric Block Cipher Algorithm

3.1 Background of The Algorithm

RC6, (hence the RC as in Ron's cipher or code) designed by Dr. Ronald C. Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yin to meet the requirements of the AES competition by NIST. (R. Rivest, et al., 1998). It is based on the RC5 block cipher. (Rivest, R.L 1997). A series of algorithms of Rivest Ciphers: RC1: Designed on paper but never implemented, RC2: A 64-bit block cipher using variable-sized keys designed to replace DES. Its code has not been made public although many companies have licensed RC2 for use in their products, RC3: Found to be breakable during development, RC4: A stream cipher using variable-sized keys; it is widely used in commercial cryptography products, although it can only be exported using keys that are 40 bits or less in length, RC5: A block-cipher supporting a variety of block sizes, key sizes, and number of encryption passes over the data, RC6: An improvement over RC5. RC6 algorithm was one of the five finalists for the AES. (Sheetal et al. , 2014). It is proprietary to RSA Security. Though the algorithm was not eventually selected, RC6 remains a good choice for security applications. RC6 block cipher was proposed, which makes essential heavy use of data-dependent rotations. Its salient features include the use of four working registers instead of two as in RC5 and the inclusion of integer multiplication as an additional primitive operation. The use of multiplication with four working registers greatly increases the diffusion achieved per round, allowing for greater security, fewer rounds, and increased throughput.

RC6 does not use S-boxes and the same algorithm is used for encryption and decryption. It is also capable to handle 128-bits plaintext and cipher text block sizes and suitable to be implemented simply using hardware or software. Advantages of RC6 is a secure, compact and simple block cipher. It offers good performance and considerable flexibility. RC6 has a simple structure and description relative to the other proposed block ciphers. RC6 is more exactly specified as RC6-w/r/b, where the parameters w, r, and b respectively express the word size (in bits), the number of rounds, and the size of the encryption key (in bytes). The RC6 block cipher symmetric algorithm is very flexible in the way that the number of rounds, size of the key and block are flexible since has a variable word size, a variable number of rounds, and a variable-length secret key. A key schedule generates 2r + 4 words (w bits each) from the b-bytes key provided by the user. These values (called round keys) are stored in an array S[0, 2r+3] and are used for both encryption and decryption.

RC6 works on a block size of 128 bits and it is very similar to RC5 in structure, using data-dependent rotations, modular addition, and XOR operations. The following computation is the most critical arithmetic operation of this block cipher:

$$f(x) = \big(x(2x+1)\big) mod\ 2w$$

For further descriptions and notation we refer to (Rivest et al. , 1998).

3.2 Overview And Major Features

RC6 (Rivest et al. , 1998) is derived from RC5 (Rivest R.L, 1997). There are two main new features in RC6 compared to RC5: the inclusion of integer multiplication and the use of four w-bit working registers instead of two w-bit registers as in RC5. The security of RC6 relies on the strength of data-dependent rotations, the mixed use of exclusive-or operations and modular addition. Integer multiplication is used to increase the diffusion achieved per round so that leads to an increased throughput and high security. Table 3-1 summarizes a comparison between RC6, RC6e, MRC6, EMRC6 & Ameliorated RC6 for different design parameters such as word size, block size, number of rounds, and secret key size.

Table 3-1: Comparison between RC6, RC6e, MRC6, EMRC6 & Ameliorated RC6 algorithms at different design parameters.

algorithm	RC6	RC6_EN	MRC6	EMRC6	Ameliorated
Working	4	8	16	32	64
w(word size	32	32	32	32	32
R (No. of	20	20	16	18	24
b (key length)	16	16	16	16	32
Block size in	128	256	512	1024	2048
No. of keys derived from key schedule	2r + 4	2r+4	8r+16	16r+32	32r+64

The notations and basic operations used for RC6 are :

1) a + b: Addition of a and b modulo 2w.

2) a – b: subtraction of a and b modulo 2w.

3) a $\oplus$ b: Exclusive-or of a and b.

4) a x b: Multiplication of a and b modulo 2w.

5) a <<< b: Rotate a to the left by the least significant log 2w bits of b.

6) a >>> b: Rotate a to the right by the least significant log 2w bits of b.

3.3 Details of RC6 Block Cipher

How RC6 algorithm work?

3.3.1 RC6 Block diagram

The RC6 has 8 stages; Stages 4 to 8 are repeated 20 times; In the third step, the first and the second sub key S0 and S1 are used. Then each round of RC6 uses two sub keys; the first one uses S2 and S3, and successive rounds use successive sub keys.To begin with, the data is first to read 128 bits and broken down to 4 x 32 bits words (A, B, C and D). Initially, and in case of encryption, the first two words in the S array are added to B and D. For Decryption, the two words are subtracted from C and A.

These four blocks make the initial 128 bits that will be fed to a register before going into the core module through a multiplexer that controls the input for the core for every round. After completing all the rounds the output is sent to a register where it will be saved. Finally, this 128 bit is broken down into four blocks again, so the final addition and subtraction will be done before sending it as the cipher data (Rivest R.L. et al. , 1998). Figure 3.1 depicts Flow chart of the RC6 algorithm.

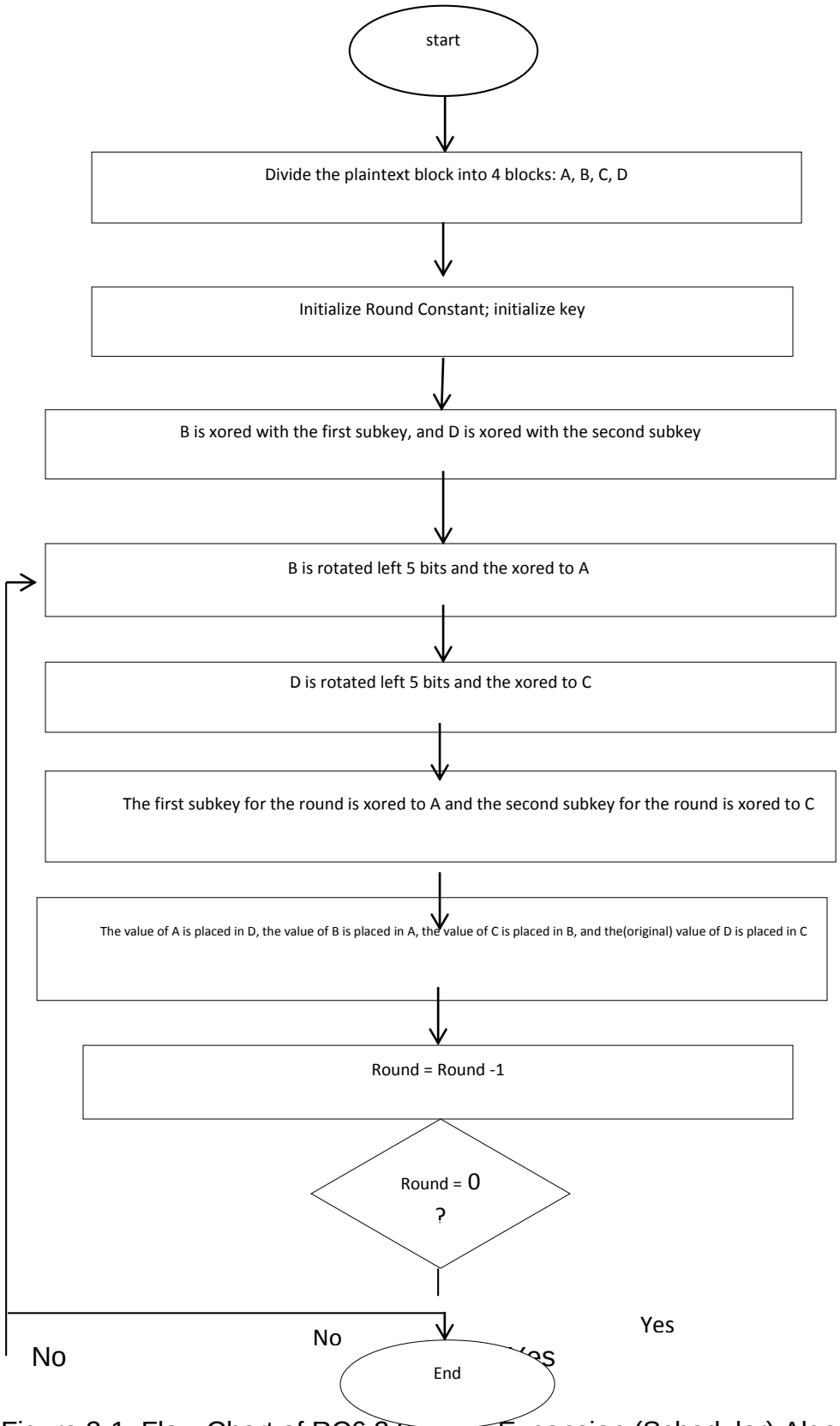

Figure 3-1: Flow Chart of RC6.3.3.2 Key Expansion (Scheduler) Algorithm for RC6

A Key generation process: From the length of b a number of internal sub keys are derived. If the key is not long enough it can be padded with zero bytes so as to achieve the required length. These sub keys are loaded into an array of c w words L[0, ..., c-1], that is the first byte is stored in L[0] and the high order byte, which can be zero-padded if it is not of the required length, goes into L[c -1]. Now the sub keys are ready to be generated. The keys generated are stored into another array S [0, ..., 2r + 3]. The size of this array is 2r+4 and in the case of the AES candidate it is 2 × 20 + 4 = 44. RC6 uses, just as its predecessor the RC5, two ”magic” constants called Pw and Qw. Pw is derived from the binary expansion of e-2, where e is the base of the natural logarithm and Qw is derived from the binary expansion of φ-1, where φ (or Phi) is the Golden Ratio. (Rivest R.L. et al, 1998)

Key generation algorithm:

Input: User-supplied b byte key preloaded into the c-word array L[0,…, c - 1], number r of rounds.

$$P\omega = b7e15\ 163$$

$$Q\omega = 9e3779b9$$

Output: w-bit round keys $S[0, \dots, 2r + 3]$.

Procedure:

1. S[0] = Pw

2. repeat step 3 for i = 1 to (2r + 3) do

3. S[i] = S[i - 1] + Qw

4. A = B = i = j = 0

5. v = 3 x max{c, 2r + 4}

6. repeat step 7 to 10 for s = 1 to v do

7. A = S[i] = (S[i] + A + B) <<< 3

8. B = L[j] = (L[j] + A + B) <<< (A + B)

9. i = (i + 1) mod (2r + 4)

10. j = (j + 1) mod c

3.3.3 Encryption Algorithm for RC6

The encryption algorithm in RC6 is relatively simple. The plaintext, which is the input, is stored in four w -bit input registers called (A, B, C, D). Keys are being stored into an array S [0,...,2r + 3]. The cipher text is the output and is being stored in (A, B, C, D). (Rivest, R.L., et al, 1998) The encryption algorithm consists of following steps: RC6 begins with two initial steps: B is added with the sub key S[0], D is added with the sub key S[1], Every round uses two sub keys, for each round i up to r the sub keys S [2i] and S[2i+1] are being used, that is the first round uses S[2] and S[3] A round can be described as:

B and D are using the function f (x) = x(2x + 1) << log2w, which means that x(2x + 1) is left-shifted 5 bits (or log2w where w = 32)

A = A ⊕ f(B) which is left-shifted with f(D) and added S[2i] C = C ⊕ f(D) which is left-shifted with f(B) and added S[2i + 1]

The four quarters in the block are being rotated as: (A, B, C, D) = (B, C, D, A)

After the last round then:

A is added with sub key S[2r + 2]

C is added with sub key S[2r + 3]

Plaintext

Cipher text

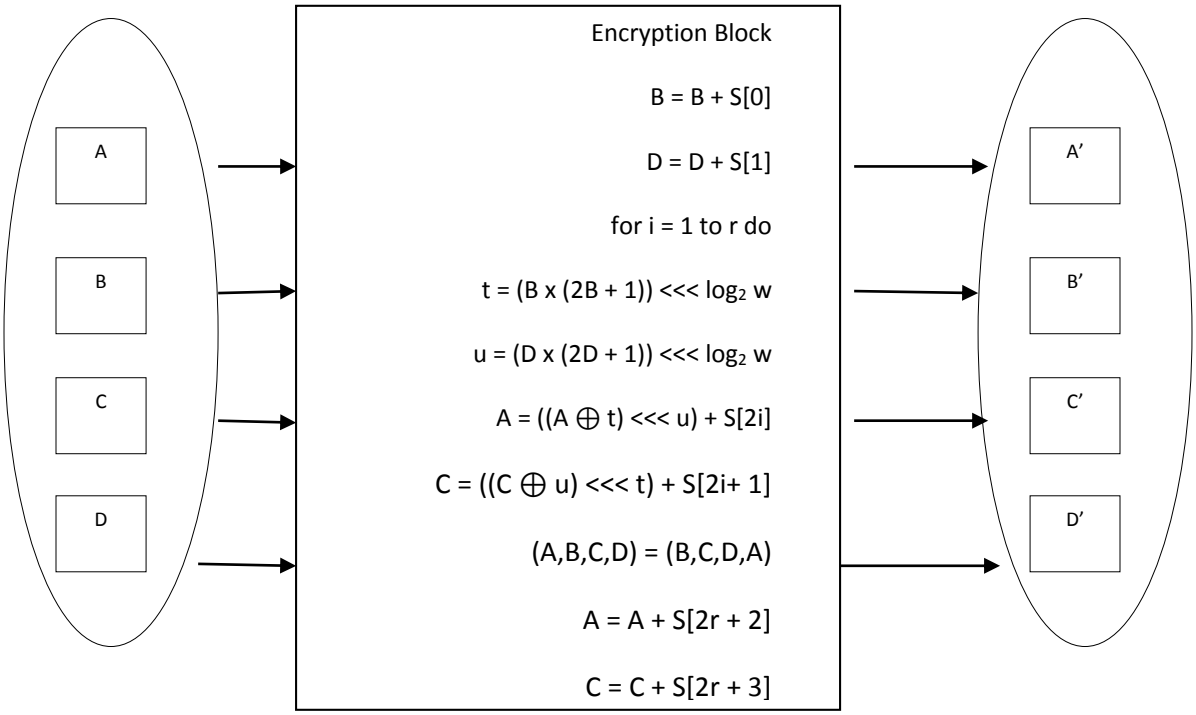

Figure 3-2: RC6 Encryption Block diagram.

The encryption algorithm:

Input: Plaintext stored in four w-bit input registers A,B,C,D Number r of rounds w-bit round keys S[0,…,2r + 3].

Output: Cipher text stored in A,B,C,D.

Procedure:

1. B = B + S[0]
2. D = D + S[1]
3. repeat step 4 to 8 for i = 1 to r do
4. t = (B x (2B + 1)) <<< log2 w
5. u = (D x (2D + 1)) <<< log2 w
6. A = ((A $\oplus$ t) <<< u) + S[2i]
7. C = ((C $\oplus$ u) <<< t) + S[2i+ 1]
8. (A,B,C,D) = (B,C,D,A)
9. A = A + S[2r + 2]
10. C = C + S[2r + 3]3.3.4 Decryption Algorithm for RC6

Decryption works in a similar way as encryption. The difference is that cipher text is the input and plaintext is the output. The use of keys and rounds is the same as for encryption. (Rivest R.L et al. , 1998)

The decryption algorithm consists of the following steps: RC6 begins with two initial steps:

C is subtracted from the sub key S [2r + 3]

A is subtracted from the sub key S [2r + 2]

Every round uses two sub keys, for each round i down to 1 the sub keys S [2i] and S[2i + 1] is being used, that is the first round uses S [21] and S [20] A round can be described as:

The four quarters in the block are being rotated as (A, B, C, D) = (D, A, B, C)

D and B are using the same function as described in the encryption, which is

f (x) = x(2x+ 1) << log2w

C = C − S[2i + 1] which is right-shifted with f(B) and the result is XOR-ed with f(D)

A = A − S[2i] which is right-shifted with f(D) and the result is XOR-ed with f(B)

After the last round then:

The sub key S[1] is subtracted D

The sub key S[0] is subtracted B

Cipher text

Plaintext

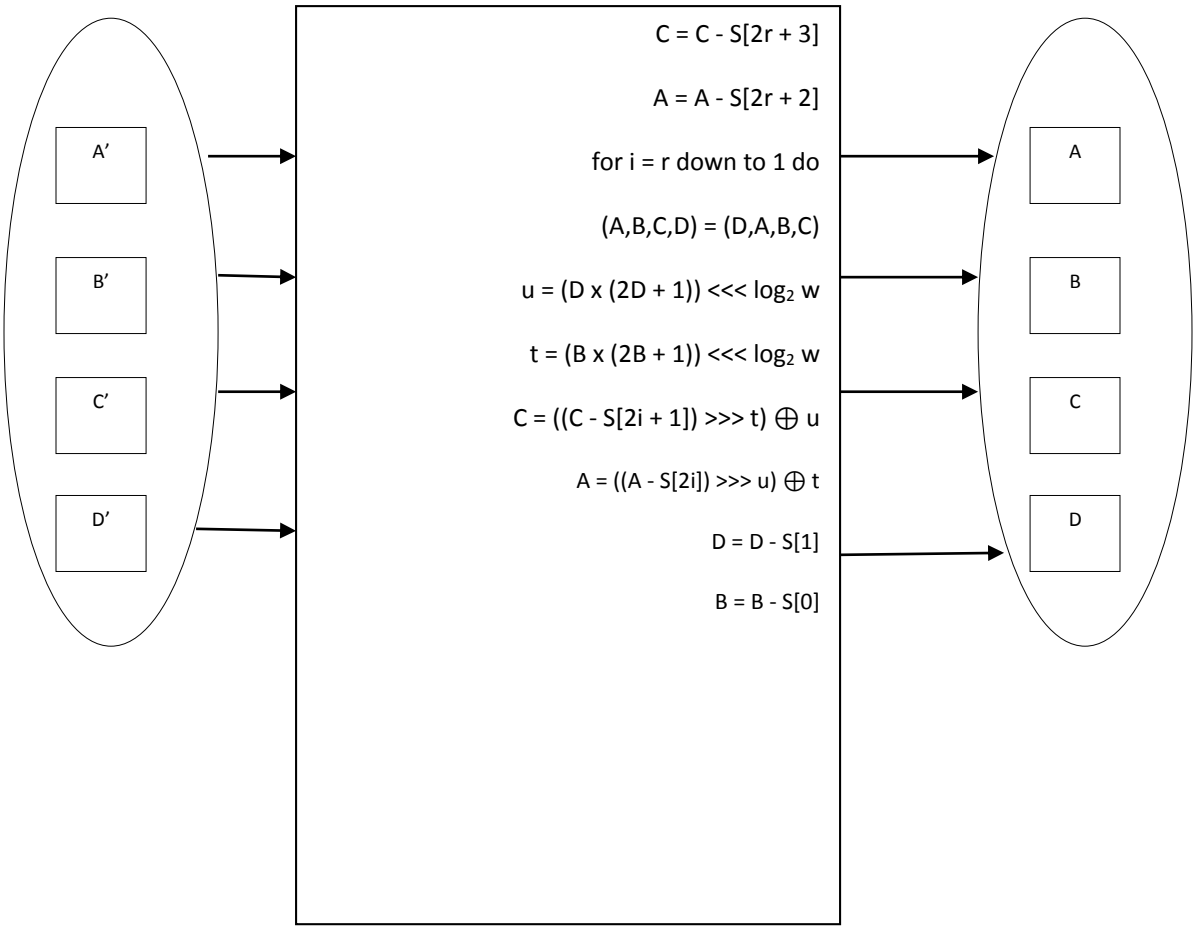

Figure 3-3: RC6 decryption Block diagram.

The decryption algorithm:

Input:

Cipher text stored in four w-bit input registers A,B,C,D, Number r of rounds, w-bit round keys S[0,…,2r + 3].

Output: Plaintext stored in A,B,C,D

Procedure:

1. C = C - S[2r + 3]
2. A = A - S[2r + 2]
3. repeat step 4 to 8for i = r down to 1 do
4. (A,B,C,D) = (D,A,B,C)
5. u = (D x (2D + 1)) <<< log2 w
6. t = (B x (2B + 1)) <<< log2 w
7. C = ((C - S[2i + 1]) >>> t) $\oplus$ u
8. A = ((A - S[2i]) >>> u) $\oplus$ t
9. D = D - S[1]
10. B = B - S[0]

Chapter Four

Proposed Approach and System Design

4.1 The Proposed Approach

Like RC6, Ameliorated RC6 proposed algorithm consists of three components, which are the key generation algorithm, an encryption algorithm, and decryption algorithm.

The following Figure 4-1 represents how the Ameliorated RC6 algorithm occurs.

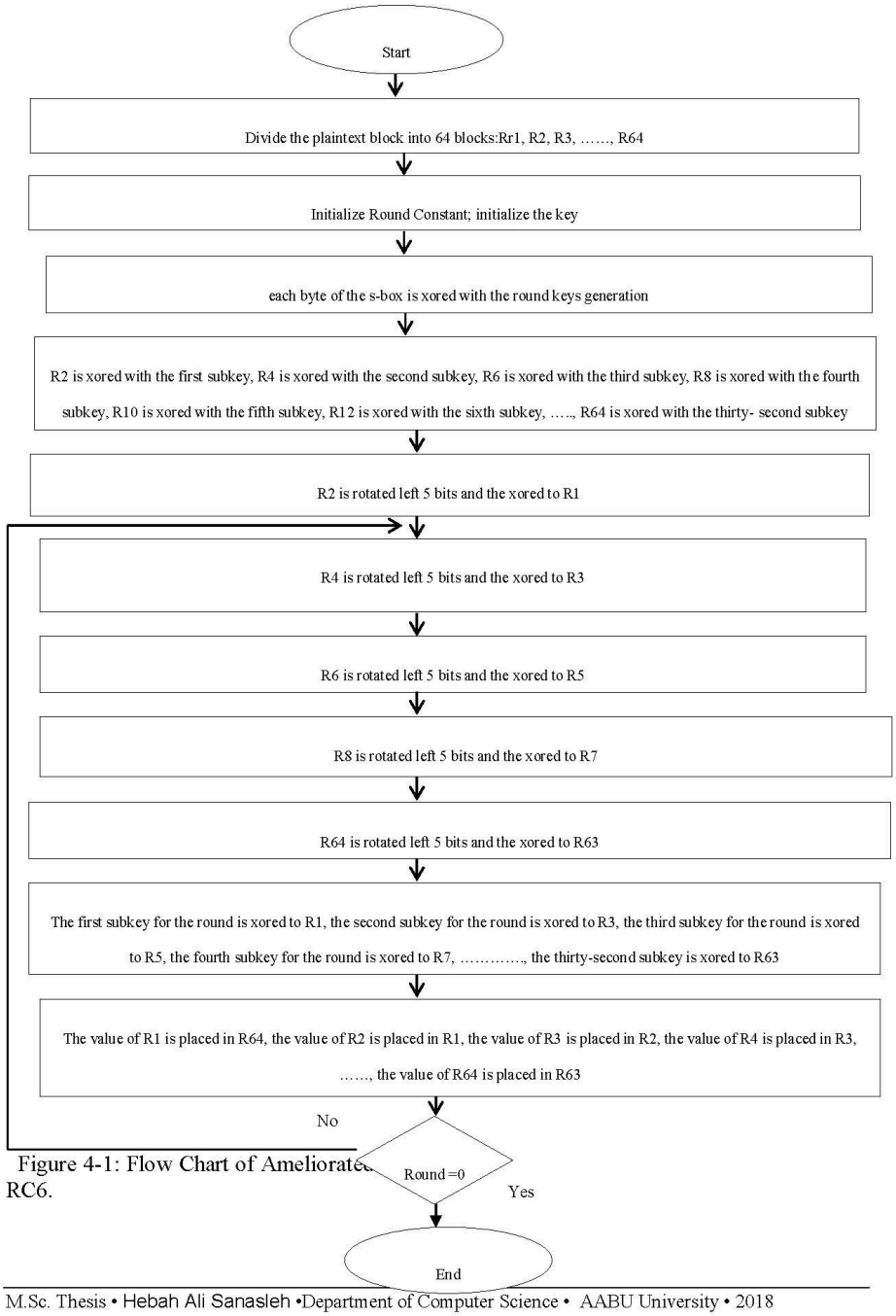

Figure 4-1: Flow Chart of Ameliorated RC6.

Key generation Algorithm:

The key schedule for Ameliorated RC6 is practically identical to the key schedule of previous block cipher RC6 with the same two magic constants Pw, Qw, but here the number of w-bit words that will be generated for the additive round keys is t=(32r+64) and these are stored in the array S[0, ..., 32r+63].

Use two magic constants:-

Pw=odd((e-2) 2^w)

Qw=odd((t-1)2^w)

Where:-

e=2.718281828459 ….. (base of natural logarithm).

t=1.618033988749 …... (golden ratio = (1+5)/2).

odd(x) is the odd integer nearest to x.

Table 4-1: Magic constants values with different word sizes.

w	16	32	64
Pw	b7el	b7el5 163	b7el5 1628aed2a6b
Qw	9e37	9e3779b9	9e3779b97f4a7c15

P32 = b7el5 163 Q32 = 9e3779b9Input: user supplied b byte key pre-loaded into the c-word, Array L[0,….,c-1], r number of rounds.

Output: (32r+64) w-bit round keys S[0,,,,32r+63].

Procedure:

S [0] = Pw

repeat step 3 For i=1 to 32r+63 do

S[i] = S [i-1] +Qw

R1=R2=i=j=0

V=3*max{c, 32r+64}

repeat step 7 to 10 For i=1 to 32r*63 do

R1=S[i] = (S[i] +R1+R2) <<<3

R2=L[j] = (L[j] +R1+R2) <<< (R1+R2)

i= (i+1) mod (32r+64)

10. j= (j+1) mod c

Encryption Algorithm:

Ameliorated RC6 works with sixty-four working registers Ri, such that (i=1,2,3,…..,64), instead of four working registers in RC6 which contain the input of data (plaintext) as well as the output encrypted data (cipher text). The first byte of plaintext or cipher text is placed in the least-significant byte of R1, but the last bit of them is placed in the most-significant bit of R64. (Rl, R2, R3, R4, ..., R63, R64) = (R2, R3, R4 , ..., R63, R64, R1) used to meet the parallel assignment of values on the right to registers on the left. The 32r+63 sub keys are used to encrypt the plain text to give cipher text.

Encryption algorithm

Input: plaintext stored in 64 w-bit input register Ri (i=1 to 64). r is a number of rounds. W-bit round keys S[0 to 32r+63].

Output: cipher text stored in register Ri (i=1 to 64).

Procedure:

R2=R2+ S [0]

R4=R4+S[1]

R6=R6+ S [2]

R8=R8+S [3]

R10=R10+ S [4]

R12=R12+S [5]

R14=R14+ S [6]

R16=R16+S [7]

R18=R18+S [8]

R20=R20+S [9]

R22=R22+S[10]

R24=R24+S[11]

R26=R26+S [12]

R28=R28+S [13]

R30=R30+S [14] R32=R32+S [15]

R34=R34+S[16]

R36=R36+S[17]

R38=R38+S[18]

R40=R40+S[19]

R42=R42+S [20]

R44=R44+S [21]

R46=R46+S[22]

R48=R48+S[23]

R50=R50+S[24]

R52=R52+S [25]

R54=R54+S [26]

R56=R56+S [27]

R58=R58+S [28]

R60=R60+S[29]

R62=R62+S[30]

R64=R64+S[31]

REPEAT STEP 34 to 98 For i=1 to r do

k3= (R2*(2R2+1)) <<<lg w

l3= (R4*(2R4+1)) <<<lg w

m3= (R6*(2R6+1)) <<<lg w

n3= (R8*(2R8+1)) <<<lg w

t3= (R10*(2R10+1)) <<<lg w

u3= (R12*(2R12+1)) <<<lg w

v3= (R14*(2R14+1)) <<<lg w

z3=(R16*(2R16+1))<<<lgw

k2=(R18*(2R18+1))<<<lg w

l2= (R20*(2R20+1)) <<<lg w

m2= (R22*(2R22+1)) <<<lg w

n2= (R24*(2R24+1)) <<<lg w

t2= (R26*(2R26+1)) <<<lg w

u2= (R28*(2R28+1)) <<<lg w

v2= (R30*(2R30+1)) <<<lg w

z2= (R32*(2R32+1)) <<<lg w

k1= (R34*(2R34+1)) <<<lg w

l1= (R36*(2R36+1)) <<<lg w

m1= (R38*(2R38+1)) <<<lg w

n1= (R40*(2R40+1)) <<<lg w

t1= (R42*(2R42+1)) <<<lg w

u1= (R44*(2R44+1)) <<<lg w

v1= (R46*(2R46+1)) <<<lg w

z1=(R48*(2R48+1))<<<lgw

k=(R50*(2R50+1))<<<lg w

l= (R52*(2R52+1)) <<<lg w

m= (R54*(2R54+1)) <<<lg w n= (R56*(2R56+1)) <<<lg w

t= (R58*(2R58+1)) <<<lg w

u= (R60*(2R60+1)) <<<lg w

v= (R62*(2R62+1)) <<<lg w

z= (R64*(2R64+1)) <<<lg w

R1= ((R1 ⊕ k3)) <<<l3) +S [32i]

R3= ((R3 ⊕ 13)) <<<k3) +S [32i+1]

R5= ((R5 ⊕ m3)) <<<n3)+S [32i+2]

R7= ((R7 ⊕ n3)) <<<m3)+S[32i+3]

R9= ((R9 ⊕ t3))<<<u3)+S[32i+4]

R11= ((R11 ⊕ u3)) <<<t3) +S [32i+5]

R13= ((R13 ⊕ v3)) <<<z3) +S [32i+6]

R15= ((R15 ⊕ z3)) <<<v3)+S [32i+7]

R17= ((R17 ⊕ k2)) <<<l2) +S [32i+8]

R19= ((R19 ⊕ l2)) <<<k2) +S [32i+9]

R21= ((R21 ⊕ m2)) <<<n2) +S [32i+10]

R23= ((R23 ⊕ n2)) <<<m2) +S [32i+11]

R25= ((R25 ⊕ t2)) <<<u2) +S [32i+12]

R27= ((R27 ⊕ u2)) <<<t2) +S [32i+13]

R29= ((R29 ⊕ v2)) <<<z2) +S [32i+14]

R31= ((R31 ⊕ z2)) <<<v2) +S [32i+15]

R33= ((R33 ⊕ k1)) <<<l1) +S [32i+16]

R35= ((R35 ⊕ l1)) <<<k1) +S [32i+17]

R37= ((R37 ⊕ m1)) <<<n1) +S [32i+18]

R39= ((R39 ⊕ n1)) <<<m1) +S [32i+19]

R41= ((R41 ⊕ t1)) <<<u1) +S [32i+20]

R43= ((R43 ⊕ u1)) <<<t1) +S [32i+21]

R45= ((R45 ⊕ v1)) <<<z1) +S [32i+22]

R47= ((R47 ⊕ z1)) <<<v1) +S [32i+23]

R49= ((R49 ⊕ k)) <<<l) +S [32i+24]

R51= ((R51 ⊕ l)) <<<k) +S [32i+25]

R53= ((R53 ⊕ m)) <<<n) +S [32i+26]

R55= ((R55 ⊕ n)) <<<m) +S [32i+27]

R57= ((R57 ⊕ t)) <<<u) +S [32i+28]

R59= ((R59 ⊕ u)) <<<t) +S [32i+29]

R61= ((R61 ⊕ v)) <<<z) +S [32i+30]

R63= ((R63 ⊕ z)) <<<v) +S [32i+31]

(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29R30,R31,R32,R33,R34,R35,R36,R37,R38,R39,R40,R41,R42,R43,R44,R45,R46,R47,R48,R49,R50,R51,R52,R53,R54,R55,R56,R57,R58,R59,R60,R61,R62,R63,R64)=(R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29 R30,R31,R32,R33,R34,R35,R36,R37,R38,R39,R40,R41,R42,R43,R44,R45,R46,R47,R48,R49,R50,R51,R52,R53,R54,R55,R56,R57,R58,R59,R60,R61,R62,R63,R64,R1) R1=R1+S [32r+32]

R3=R3+S[32r+33]

R5=R5+S [32r+34]

R7=R7+S [32r+35]

R9=R9+S [32r+36]

R11=R11+S [16r+37]

R13=R13+S [32r+38]

R15=R15+S [32r+39]

R17=R17+S [32r+40]

R19=R19+S [32r+41]

R21=R21+S [32r+42]

R23=R23+S [32r+43]

R25=R25+S [32r+44]

R27=R27+S [32r+45]

R29=R29+S [32r+46]

R31=R31+S [32r+47]

R33=R33+S [32r+48]

R35=R35+S [32r+49]

R37=R37+S [32r+50]

R39=R39+S [32r+51]

R41=R41+S [32r+52]

R43=R43+S [32r+53]

R45=R45+S [32r+54]

R47=R47+S [32r+55]

R49=R49+S [32r+56]

R51=R51+S [32r+57]

R53=R53+S [32r+58]

R55=R55+S [32r+59]

R57=R57+S [32r+60]

R59=R59+S [32r+61]

R61=R61+S [32r+62]

R63=R63+S [32r+63]

text

R2=R2+ S [0]

R4=R4+S[1]

R6=R6+ S [2]

R8=R8+S [3]

R10=R10+ S [4]

R12=R12+S [5]

R14=R14+ S [6]

R16=R16+S [7]

R18=R18+S [8]

R20=R20+S [9]

R60=R60+S[29]

R62=R62+S[30]

R64=R64+S[31]

For i=1 to r do

k3= (R2*(2R2+1)) <<<lg w

l3= (R4*(2R4+1)) <<<lg w

m3= (R6*(2R6+1)) <<<lg w

n3= (R8*(2R8+1)) <<<lg w

t3= (R10*(2R10+1)) <<<lg w

u3= (R12*(2R12+1)) <<<lg w

v3= (R14*(2R14+1)) <<<lg w

z3=(R16*(2R16+1))<<<lgw

k=(R50*(2R50+1))<<<lg w

l= (R52*(2R52+1)) <<<lg w

m= (R54*(2R54+1)) <<<lg w

n= (R56*(2R56+1)) <<<lg w

t= (R58*(2R58+1)) <<<lg w

u= (R60*(2R60+1)) <<<lg w

v= (R62*(2R62+1)) <<<lg w

z= (R64*(2R64+1)) <<<lg w

R1= ((R1 ⊕ k3)) <<<l3) +S [32i]

R3= ((R3 ⊕ 13)) <<<k3) +S [32i+1]

R5= ((R5 ⊕ m3)) <<<n3)+S [32i+2]

R7= ((R7 ⊕ n3)) <<<m3)+S[32i+3]

R9= ((R9 ⊕ t3))<<<u3)+S[32i+4]

…….

R59=R59+S [32r+61]

R61=R61+S [32r+62]

R63=R63+S [32r+63]

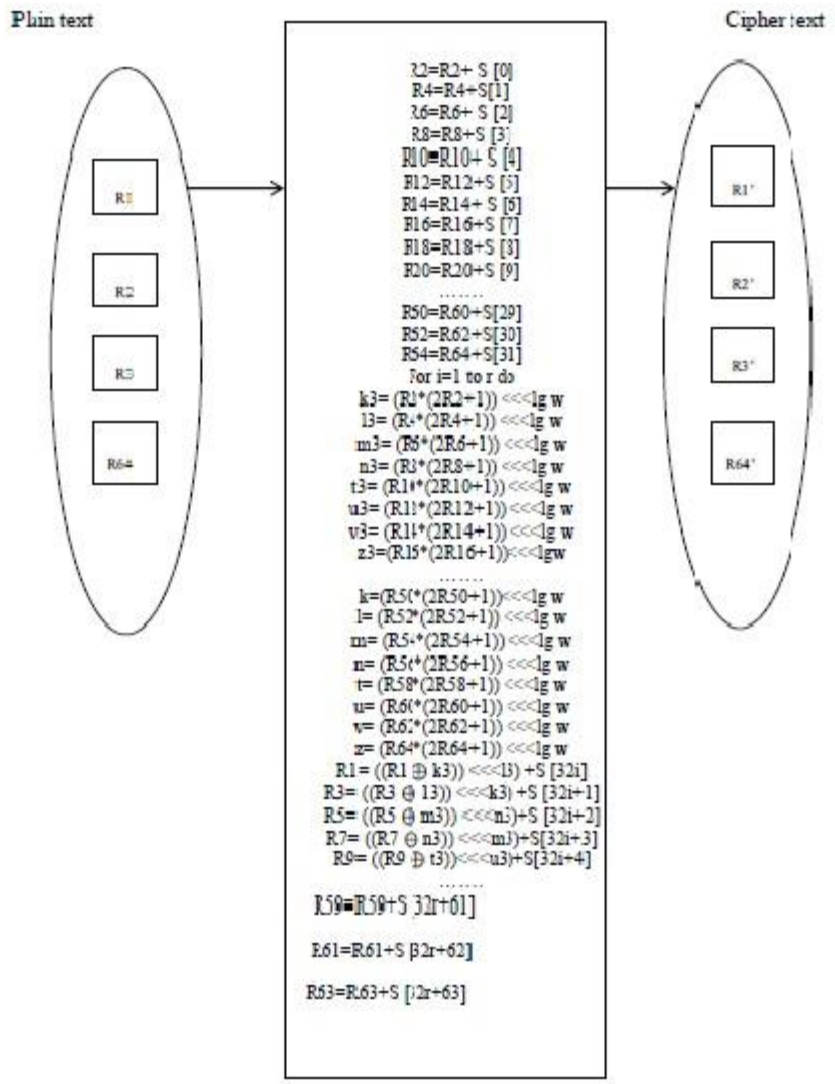

Figure 4-2: Ameliorated RC6 Encryption Block diagram

Decryption Algorithm:

In this proposed algorithm the decryption process is just inverse of the encryption process.

Decryption algorithm

Input:

cipher text stored in 64 w-bit input register Ri (i=1 to 64).

r is number of rounds. W-bit round keys S [0 to 32r+63].

Output:

plaintext stored in register Ri (i=1 to 64).

Procedure:

R63=R63-S [32r+63]

R61=R61-S [32r+62]

R59=R59-S [32r+61]

R57=R57-S [32r+60]

R55=R55-S [32r+59]

R53=R53-S [32r+58]

R51=R51-S [32r+57]

R49=R49-S [32r+56]

R47=R47-S [32r+55]

R45=R45-S [32r+54]

R43=R43-S [32r+53]

R41=R41-S [32r+52]

R39=R39-S [32r+51]R37=R37-S [32r+50]

R35=R35-S [32r+49]

R33=R33-S [32r+48]

R31=R31-S [32r+47]

R29=R29-S [32r+46]

R27=R27-S [32r+45]

R25=R25-S [32r+44]

R23=R23-S [32r+43]

R21=R21-S [32r+42]

R19=R19-S [32r+41]

R17=R17-S [32r+40]

R15=R15-S [32r+39]

R13=R13-S [32r+38]

R11=R11-S [16r+37]

R9=R9-S [32r+36]

R7=R7-S [32r+35]

R5=R5-S [32r+34]

R3=R3-S[32r+33]

R1=R1-S [32r+32]

REPEAT STEP 34 to 98 For i=r down to 1 do

(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29R30,R31,R32,R33,R34,R35,R36,R37,R38,R39,R40,R41,R42,R43,R44,R45,R46,R47,R48,R49,R50,R51,R52,R53,R54,R55,R56,R57,R58,R59,R60,R61,R62,R63,R64)=(R64,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29
R30,R31,R32,
R33,R34,R35,R36,R37,R38,R39,R40,R41,R42,R43,R44,R45,R46,R47,R48,R49,R50,R51,R52,R53,R54,R55,R56,R57,R58,R59,R60,R61,R62,R63)

z= (R64*(2R64+1)) <<<lg w

v= (R62*(2R62+1)) <<<lg w

u= (R60*(2R60+1)) <<<lg w

t= (R58*(2R58+1)) <<<lg w

n= (R56*(2R56+1)) <<<lg w

m= (R54*(2R54+1)) <<<lg w

l= (R52*(2R52+1)) <<<lg w

k=(R50*(2R50+1))<<<lg w

z1=(R48*(2R48+1))<<<lgw

v1= (R46*(2R46+1)) <<<lg w

u1= (R44*(2R44+1)) <<<lg w

t1= (R42*(2R42+1)) <<<lg w

n1= (R40*(2R40+1)) <<<lg w

m1=(R38*(2R38+1)) <<<lg w

l1=(R36*(2R36+1)) <<<lg w

k1=(R34*(2R34+1)) <<<lg w

z2= (R32*(2R32+1)) <<<lg w

v2= (R30*(2R30+1)) <<<lg w u2= (R28*(2R28+1)) <<<lg w

t2= (R26*(2R26+1)) <<<lg w

n2= (R24*(2R24+1)) <<<lg w

m2= (R22*(2R22+1)) <<<lg w

l2= (R20*(2R20+1)) <<<lg w

k2=(R18*(2R18+1))<<<lg w

z3=(R16*(2R16+1))<<<lgw

v3= (R14*(2R14+1)) <<<lg w

u3= (R12*(2R12+1)) <<<lg w

t3= (R10*(2R10+1)) <<<lg w

n3= (R8*(2R8+1)) <<<lg w

m3= (R6*(2R6+1)) <<<lg w

l3= (R4*(2R4+1)) <<<lg w

k3= (R2*(2R2+1)) <<<lg w

R63= ((R63 -S [32i+31])) >>>v) ⊕ z

R61= ((R61 -S [32i+30])) >>>z) ⊕ v

R59= ((R59 -S [32i+29])) >>>t) ⊕ u

R57= ((R57 -S [32i+28])) >>>u) ⊕ t

R55= ((R55 -S [32i+27])) >>>m) ⊕ n

R53= ((R53 -S [32i+26])) >>>n) ⊕ m

R51= ((R51 -S [32i+25])) >>>k) ⊕ l

R49= ((R49 -S [32i+24])) >>>l) ⊕ k

R47= ((R47 -S [32i+23])) >>>v1) ⊕ z1

R45= ((R45 -S [32i+22])) >>>z1) ⊕ v1

R43= ((R43 -S [32i+21])) >>>t1) ⊕ u1

R41= ((R41 -S [32i+20])) >>>u1) ⊕ t1

R39= ((R39 -S [32i+19])) >>>m1) ⊕ n1

R37= ((R37 -S [32i+18])) >>>n1) ⊕ m1

R35= ((R35 -S [32i+17])) >>>k1) ⊕ l1

R33= ((R33 -S [32i+16])) >>>l1) ⊕ k1

R31= ((R31-S [32i+15])) >>>v2) ⊕ z2

R29= ((R29 -S [32i+14])) >>>z2) ⊕ v2

R27= ((R27 -S [32i+13])) >>>t2) ⊕ u2

R25= ((R25 -S [32i+12])) >>>u2) ⊕ t2

R23= ((R23 -S [32i+11])) >>>m2) ⊕ n2

R21= ((R21-S [32i+10])) >>>n2) ⊕ m2

R19= ((R19 -S [32i+9])) >>>k2) ⊕ l2

R17= ((R17-S [32i+8])) >>>l2) ⊕ k2

R15= ((R15-S [32i+7])) >>>v3) ⊕ z3

R13= ((R13-S [32i+6])) >>>z3) ⊕ v3

R11= ((R11 -S [32i+5])) >>>t3) ⊕ u3

R9= ((R9 -S[32i+4]))>>>u3) ⊕ t3

R7= ((R7-S[32i+3])) >>>m3) ⊕ n3

R5= ((R5 -S [32i+2])) >>>n3) ⊕ m3R3= ((R3 -S [32i+1])) >>>k3) ⊕13

R1= ((R1 -S [32i])) >>>l3) ⊕ k3

R64=R64-S[31]

R62=R62-S[30]

R60=R60-S[29]

R58=R58-S [28]

R56=R56-S [27]

R54=R54-S [26]

R52=R52-S [25]

R50=R50-S[24]

R48=R48-S[23]

R46=R46-S[22]

R44=R44-S [21]

R42=R42-S [20]

R40=R40-S[19]

R38=R38-S[18]

R36=R36-S[17]

R34=R34-S[16]

R32=R32-S [15]

R30=R30-S [14]

R28=R28-S [13]

R26=R26-S [12]

R24=R24-S[11]

R22=R22-S[10]

R20=R20-S [9]

R18=R18-S [8]

R16=R16-S [7]

R14=R14- S [6]

R12=R12-S [5]

R10=R10- S [4]

R8=R8-S [3]

R6=R6- S [2]

R4=R4-S[1]

R2=R2-S[0]

R63=R63-S [32r+63]

R61=R61-S [32r+62]

R59=R59-S [32r+61]

R57=R57-S [32r+60]

R55=R55-S [32r+59]

R53=R53-S [32r+58]

R51=R51-S [32r+57]

R9=R9-S [32r+36]

R7=R7-S [32r+35]

R5=R5-S [32r+34]

R3=R3-S[32r+33]

R1=R1-S [32r+32]

i=r down to 1 do

(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,

R11,R12,R13,R14,R15,R16,R17,R18,

R19,R20,R21,R22,R23,R24,R25,R26,

R27,R28,R29R30,R31,R32,R33,R34,

R35,R36,R37,R38,R39,R40,R41,R42,

R43,R44,R45,R46,R47,R48,R49,R50,

R51,R52,R53,R54,R55,R56,R57,R58,

R59,R60,R61,R62,R63,R64)=

R64,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,

R11,R12,R13,R14,R15,R16,R17,R18,

R19,R20,R21,R22,R23,R24,R25,R26,

R27,R28,R29 R30,R31,R32, R33,R34,

R35,R36,R37,R38,R39,R40,R41,R42,

R43,R44,R45,R46,R47,R48,R49,R50,

R51,R52,R53,R54,R55,R56,R57,

R58,R59,R60,R61,R62,R63)

z= (R64*(2R64+1)) <<<lg w

v= (R62*(2R62+1)) <<<lg w

……..

R10=R10- S [4]

R8=R8-S [3]

R6=R6- S [2]

R4=R4-S[1]

R2=R2-S [0]

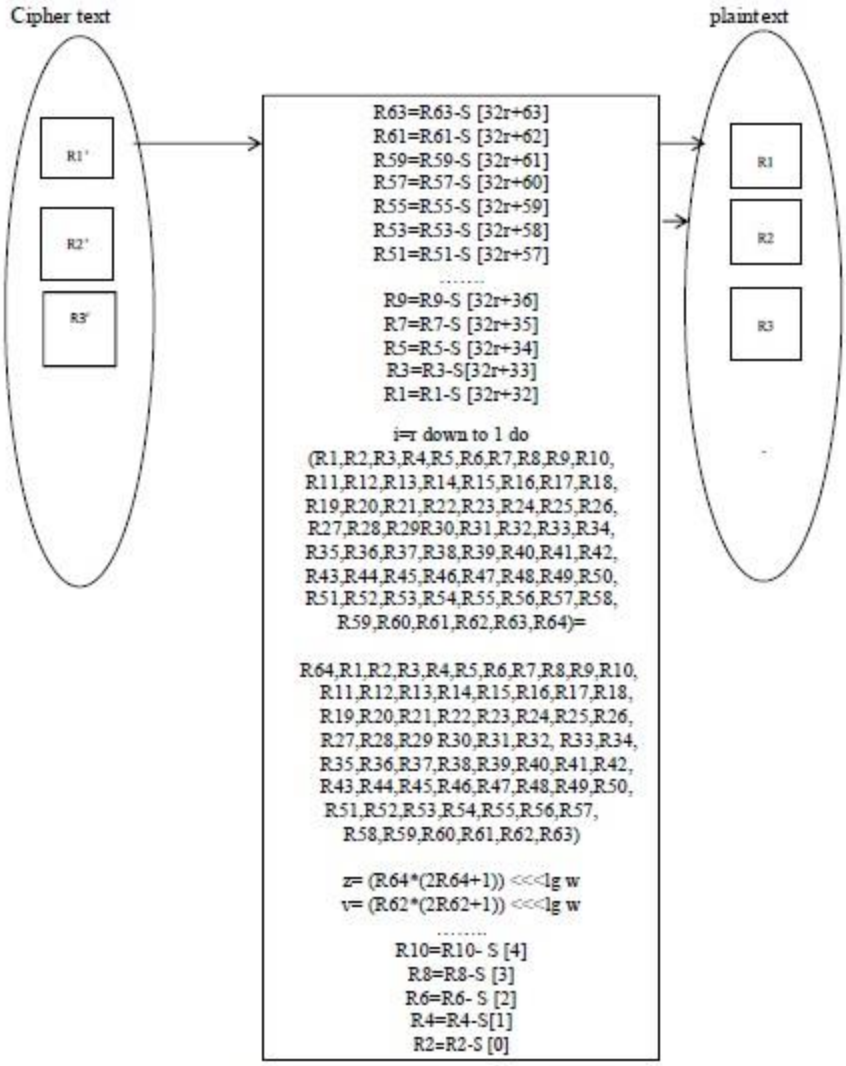

Figure 4-3: Ameliorated RC6 decryption Block diagram.

4.2 System Design

4.2.1 System Block Diagram

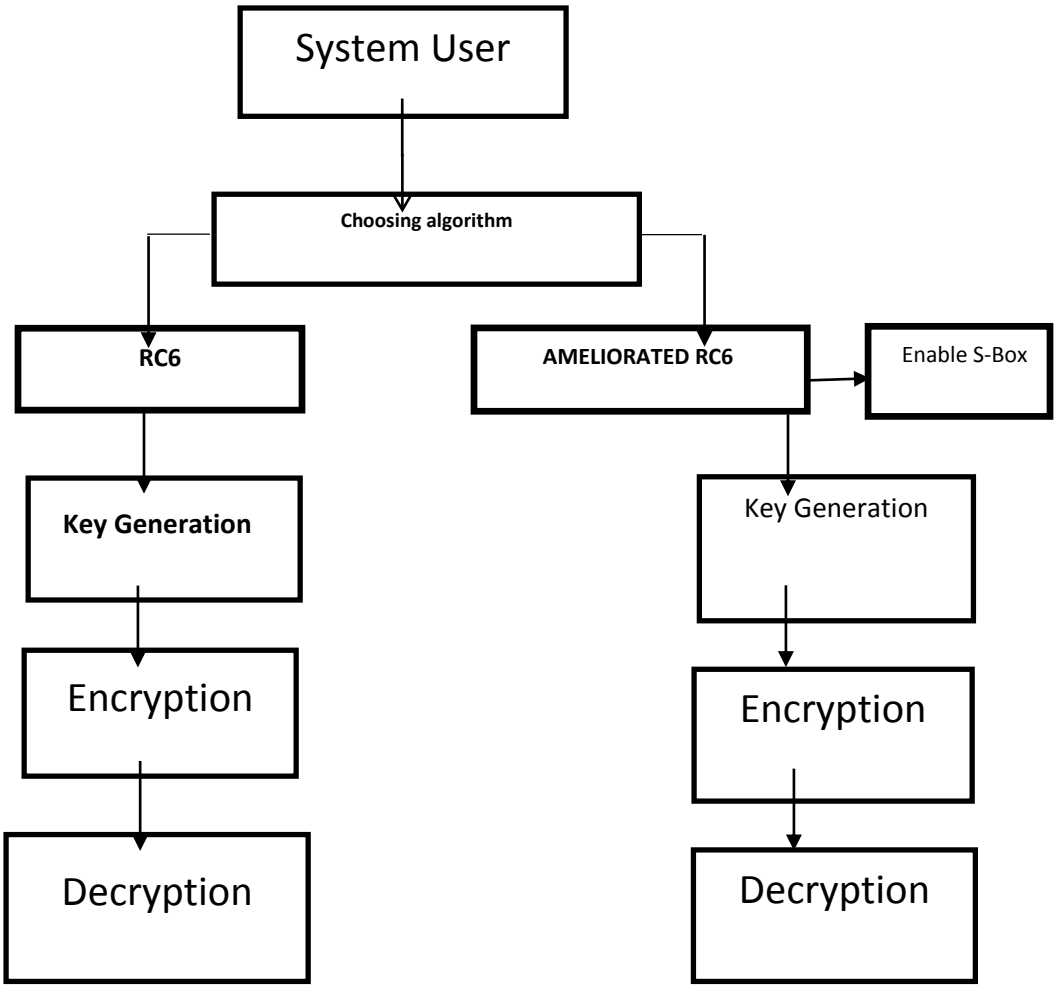

Figure 4-4: System Block Diagram

The S-Box used in Ameliorated RC6 is a single S-box of 512 32-bit words to provide good resistance against linear and differential attacks.

In key generation algorithm, keys after generated XOR-ed with values from s-box in key scheduler ameliorated RC6 system.

```
if (ckb_sbox.Checked)

for (int t = 1; t < size_a; t++)

S[t] = S[t] ^ Sbox[t % 512];
```

The implementation of the proposed method is represented by a system designed using c#. This system is called "Start", which refers to "Choose Algorithm". The system processes the operations of key generation and encryption as well as decryption, but before that there is a choosing algorithm which wanted to use. If a user chooses RC6 they will enter the RC6 system. Otherwise, they will enter RC6 Ameliorated system. Three operations are used in this system; three for RC6 and three for Ameliorated RC6 algorithm.

Input Text or data set used in results is: "it is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularized in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software".

- First Stage:

4.2.2 Choose Algorithm

This is the first stage of the system. It is used to choose the user to select the algorithm will want to use. This stage works as follows: when the user enters the system, the starting window contains two buttons, by clicking on the "RC6 " button, the user will enter to RC6 algorithm system, and by clicking on the " Ameliorated " button, the user will enter to RC6 Ameliorated algorithm system.

The welcoming interface consists of two buttons named " RC6 " and " AMELIORATED ":

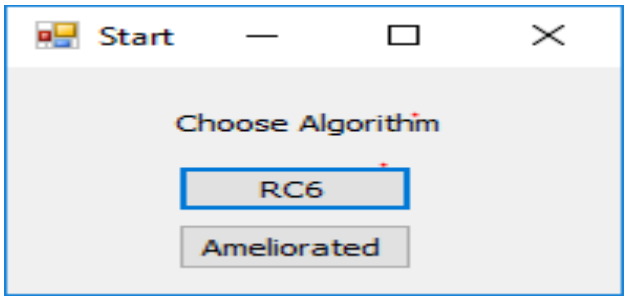

Figure 4-5: First System Screen.

If the user clicks on "RC6" button, they will enter the RC6 algorithm directly. But if the user clicks on the " Ameliorated " button, they will enter the RC6 Ameliorated algorithm directly. - Second Stage:

4.2.3 RC6 algorithm

If the user chooses the RC6 algorithm, the key generation process starts by computing the time taken to generate a key which's 16 bytes size. After that, the encryption process starts by entering a text or loading a text file, then compute the time taken to encrypt text or text file. Finally, the decryption process starts, then compute the time taken to decrypt text or text file.

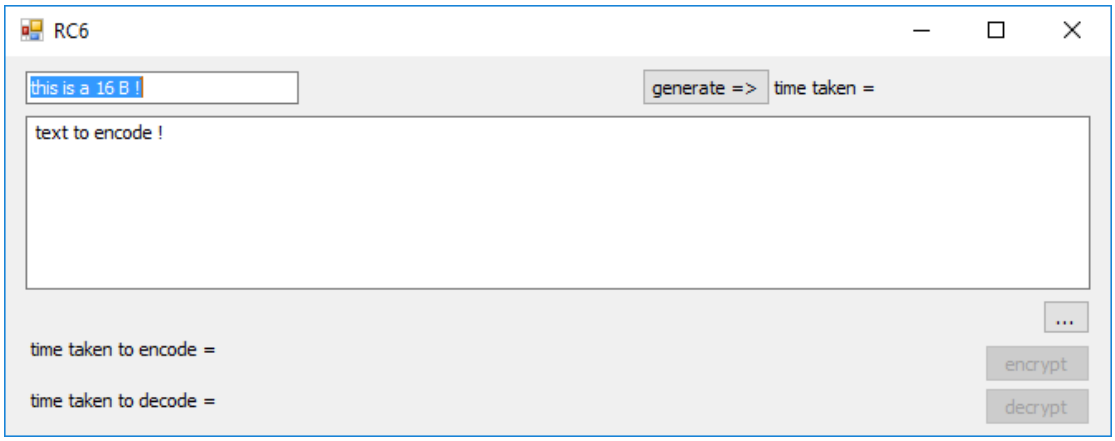

Figure 4-6: Second System Screen.

The figure above represents the RC6 algorithm which press "generate" to generate key and computation generation key time, then asks the user to enter the text or load a text, then press "encrypt" to complete the operation of encryption, and computation encryption time. After that press "decrypt" to complete the operation of decryption, and computation decryption time.

4.2.4 Ameliorated RC6 algorithm

If the user chooses the Ameliorated RC6 algorithm, firstly the user must check S-Box to enable S-Box, then the key generation process starts by computing the time taken to generate a key which's 32 bytes size. After that the encryption process starts by entering a text or loading a text file, then compute the time taken to encrypt text or text file. finally, the decryption process starts, then compute the time taken to decrypt text or text file.

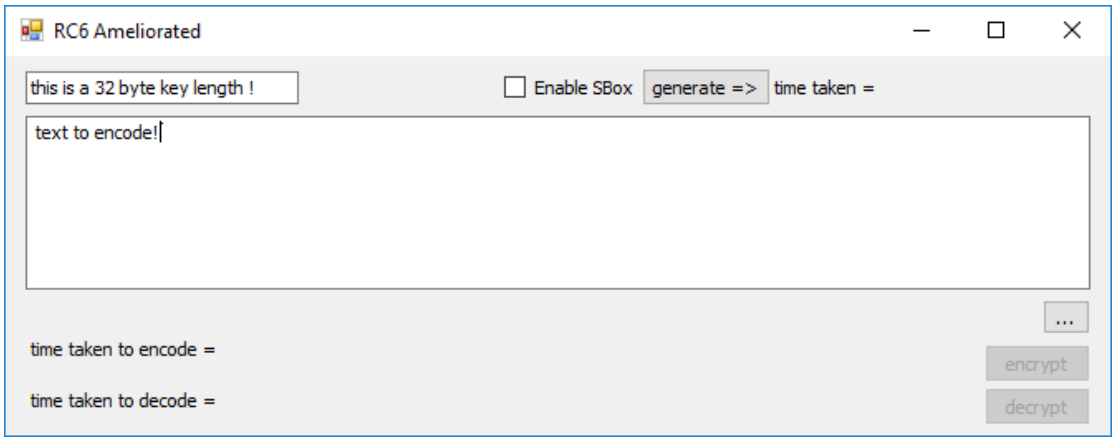

Figure 4-7: Third System Screen.

The figure above represents the Ameliorated RC6 algorithm which checks Enable S-Box then press "generate" to generate key and computation generation key time, then asks the user to enter the text or load a text, then press "encrypt" to complete the operation of encryption, and computation encryption time. After that press "decrypt" to complete the operation of decryption, and computation decryption time.

4.3 Summary

In chapter 4, we explained the proposed algorithm. Both encryption and decryption have been clarified; we start by choosing the algorithms. The RC6 algorithm encrypts and decrypts a text then computes time of key generation, encryption and decryption. The Ameliorated RC6 encrypt and decrypt a text then computes time of key generation, encryption and decryption. Moreover, both algorithms; encrypt and decrypt a text then computes time of key generation, encryption and decryption. But the difference between them lies in different block size, different number of rounds, different key size, and use a box called S-Box in Ameliorated RC6, which is not used in RC6.

Chapter Five

Experiment Results and discussion

The performance of any cryptographic algorithm is measured basically by three factors: encryption and decryption time, encryption and decryption throughput, and security.

5.1 Parametric Comparison

Table 5-1 summarizes the comparison between RC6 and Ameliorated RC6 for different design parameters such as word size, block size, number of rounds and secret key size.

Table 5-1: Comparison on the basis of parameters between RC6 and Ameliorated RC6 Block Cipher

Parameters	Algorithm type	
	RC6	Ameliorated RC6
w (word size in bits)	32	32
R (No. of rounds)	20	24
b (key length) in bytes	16	32
No. of registers	4	64
Block size in bits	128	2048
No. of keys derived	2r + 4	32 r + 64
s-box	does not use s-box	Use s-box

5.2 Analysis Comparison

The comparative analysis between RC6 and Ameliorated RC6 is performed to provide some measurements on the encryption and decryption processes. Effects of several parameters such as number of rounds(r), Block size, and the length of the secret key (b) on the performance evaluation criteria are investigated. The measuring of encryption time, decryption time, throughput of encryption and throughput of decryption of both block ciphers are considered. The analysis described in sections (A-G).

The algorithms were implemented using c# 2017. Following results were obtained on Intel (R) Core (TM) i3 CPU @ 1.40 64 bit system with 2 GB of RAM running Windows 10 Home. Several performance metrics are collected: key generation time, encryption time, decryption time, and the throughput of the encryption/decryption. The encryption/decryption time is considered the time that an encryption/decryption algorithm takes to produce a cipher text from a plaintext. Encryption and decryption time is used to calculate the throughput of an encryption and decryption. It indicates the speed of encryption/decryption.

The comparison between RC6 and Ameliorated RC6 based on:

Encryption time and decryption time.

Throughput of encryption and throughput of decryption.

Security

5.2.1 Encryption time and decryption time

A. Key generation Time for RC6, and Ameliorated RC6

Table 5-2, and Fig. 5-1, show the Key generation time for RC6, and Ameliorated RC6 at the same design parameter, word size (w)=32, but with different Block size, number of rounds (r), and the length of secret key (b).Which is parameter of RC6; block size=128 with 4 registers, r=20, b=16 bytes, while Ameliorated RC6; block size=2048 with 64 registers, r=24, b=32 bytes.

Table 5-2 Key generation Time (sec) for RC6, and Ameliorated RC6

Ameliorated RC6 key generation	RC6 key generation
0.0003136	0.0002572
0.0003171	0.0002603
0.0003185	0.0002434
0.0003243	0.0002456
0.0003799	0.0002404
0.0003651	0.0002528
0.0003599	0.000369
0.0003586	0.0003825
0.0004257	0.0004041
0.0004558	0.0004164

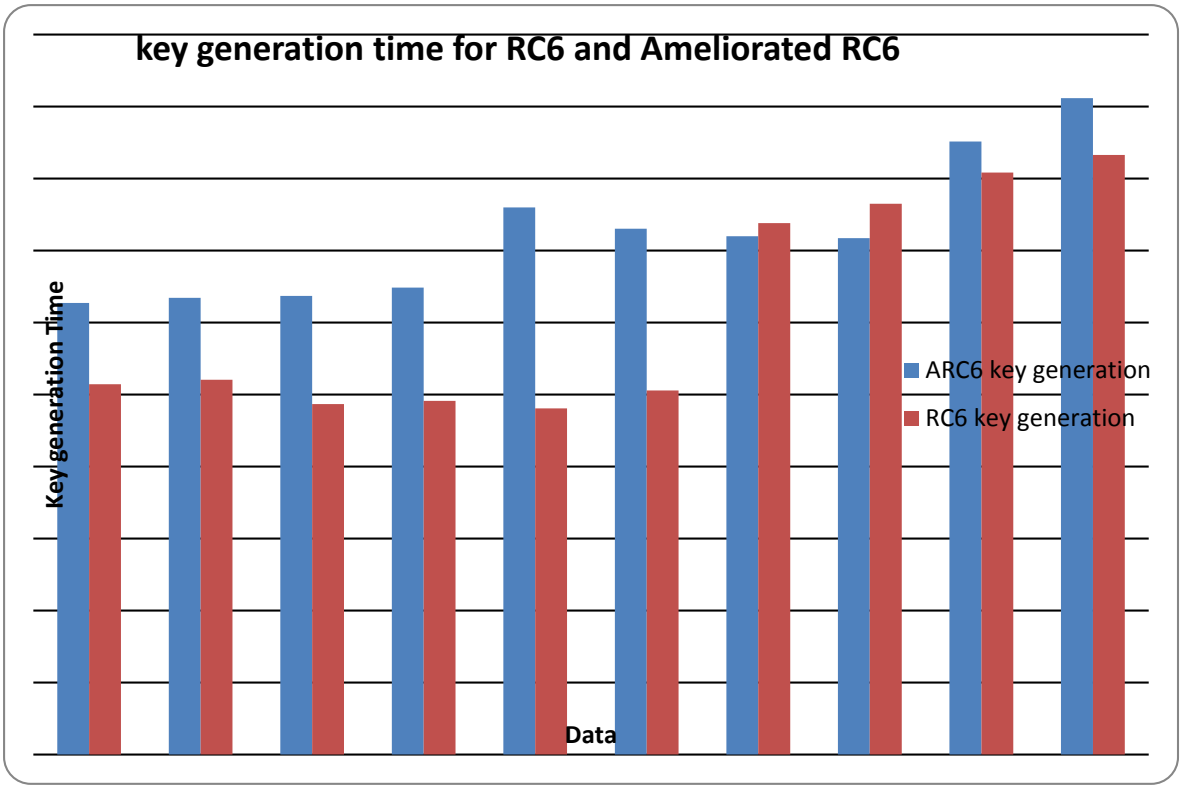

Figure 5-1: key generation time (second) for RC6, and Ameliorated RC6B. Encryption Time for RC6, and Ameliorated RC6

Table 5-3, Fig. 5-2, show the encryption time for RC6, and Ameliorated RC6 at the same design parameter, word size (w)=32, but different Block size, number of rounds (r), and the length of secret key (b).Which is design parameter of RC6; block size=128 with 4 registers, r=20, b=16 bytes, while Ameliorated RC6; block size=2048 with 64 registers, r=24, b=32 bytes. These results are obtained by encrypting several sizes of data blocks. The comparisons confirm that the Ameliorated RC6 consume more encryption time than RC6, but it has maximum throughput than RC6, because it works on 2048 bits block size with 64 registers instead of four in RC6.

Table 5-3: Time encryption (sec) for RC6 and Ameliorated RC6

Ameliorated RC6 encryption	RC6 encryption	File size (MB)
0.0000006	0.000019	0.056
0.0000006	0.0000192	0.139
0.0000008	0.0000279	0.222
0.0000007	0.0000289	0.377
0.0000008	0.0000308	0.492
0.0000009	0.000031	0.983
0.000001	0.0000311	1.33
0.0000011	0.0000313	2.659
0.0000012	0.0000315	5.318
0.0000013	0.0000328	7.091

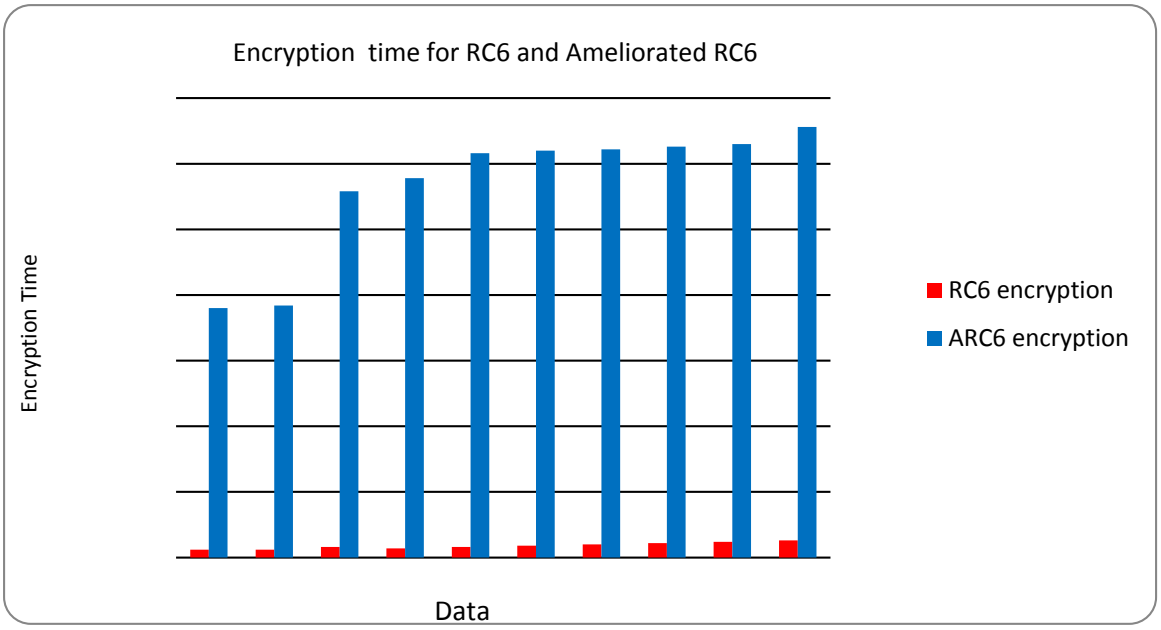

Figure 5-2: Comparison of encryption time (second) for RC6, and Ameliorated RC6C.
Decryption Time for RC6, and Ameliorated RC6

Table 5-4, Fig. 5-3, show the decryption time for RC6, and Ameliorated RC6 at the same design parameter, word size (w)=32, but different Block size, number of rounds (r), and the length of secret key (b).Which is design parameter of RC6; block size=128 with 4 registers, r=20, b=16 bytes, while Ameliorated RC6; block size=2048 with 64 registers, r=24, b=32 bytes. These results are obtained by decrypting several sizes of data blocks. The comparisons confirm that the Ameliorated RC6 consume more decryption time than RC6, but it has maximum throughput than RC6, because it works on 2048 bits block size with 64 registers instead of four in RC6.

Table 5-4: Time decryption (sec) for RC6 and Ameliorated RC6

Ameliorated RC6 encryption	RC6 encryption	File size (MB)
0.0000191	0.0000007	0.056
0.0000294	0.0000008	0.139
0.0000279	0.0000008	0.222
0.0000299	0.0000008	0.377
0.0000308	0.0000009	0.492
0.000031	0.0000009	0.983
0.0000311	0.000001	1.33
0.0000313	0.0000012	2.659
0.0000317	0.0000011	5.318
0.0000328	0.0000013	7.091

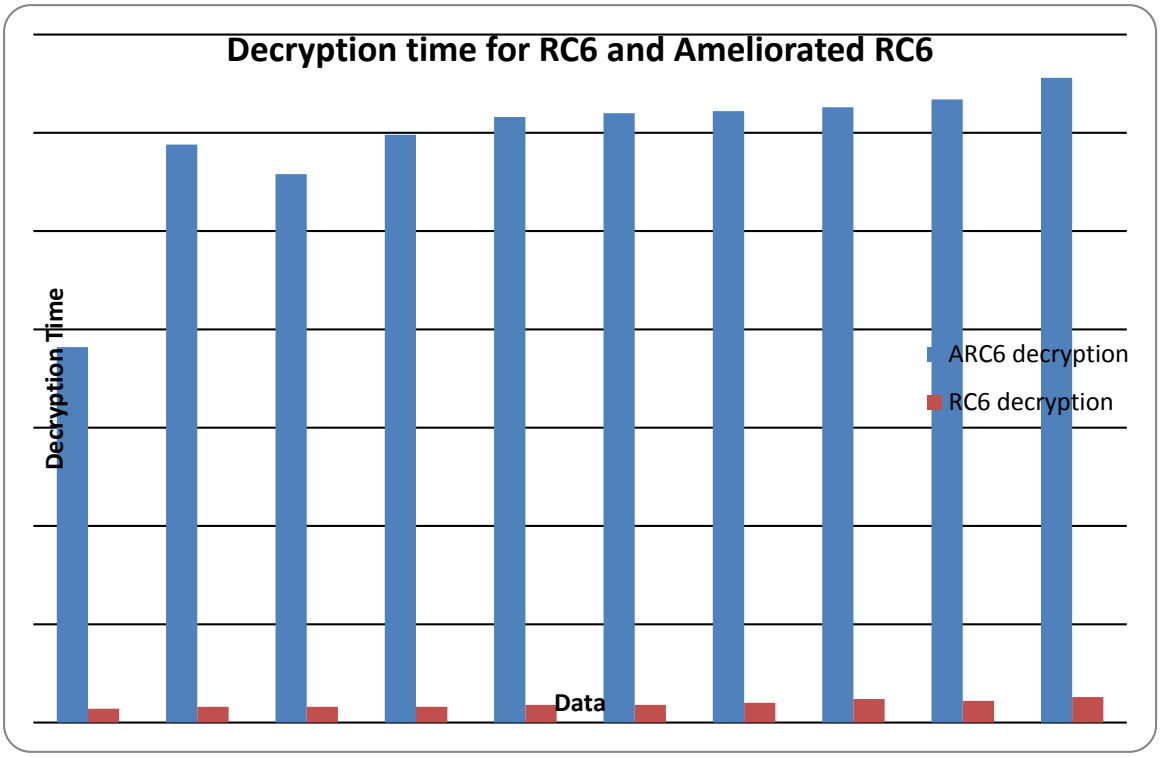

figure 5-3: comparison of decryption time (second) for RC6, and Ameliorated RC6

5.2.2 Throughput of encryption and throughput of decryption

D. Encryption Throughput for RC6, and Ameliorated RC6

The throughput of the encryption is calculated as the total plaintext in bytes encrypted divided by the encryption time. The throughput is computed for RC6, and Ameliorated RC6 at word size (w)=32, but different Block size, number of rounds (r), and the length of secret key (b).Which is design parameter of RC6; block size=128 with 4 registers, r=20, b=16 bytes, while Ameliorated RC6; block size=2048 with 64 registers, r=24, b=32 bytes. Table 5-5 and Figure 5-4 show the obtained results, and also show that Ameliorated RC6 achieves maximum encryption throughput than RC6.

Table 5-5: Encryption Throughput for RC6, and Ameliorated RC6

Throughput of encryption	Algorithm
20.74	RC6
64.3	Ameliorated RC6

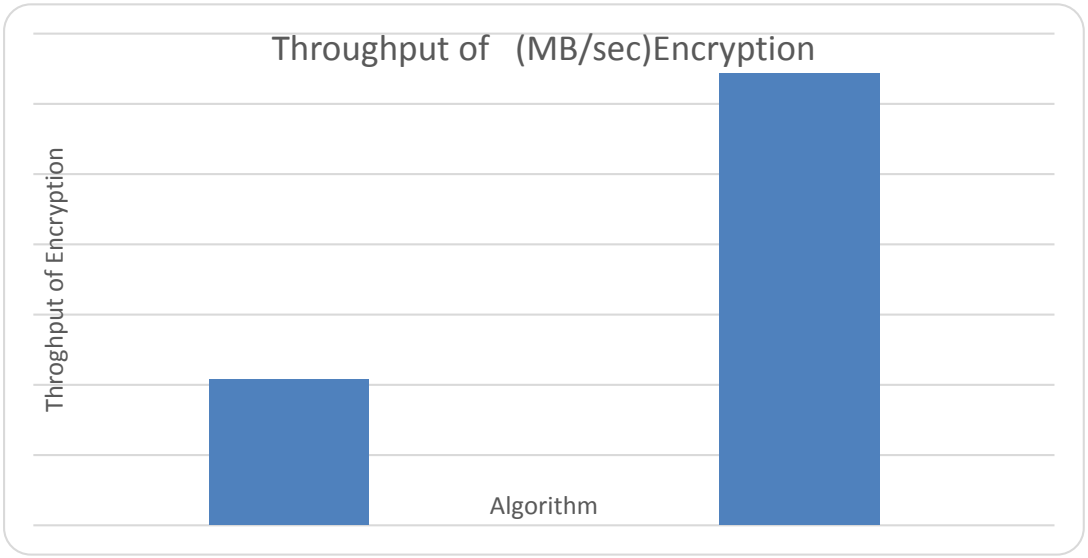

Figure 5-4: Throughput (MB/sec) of Encryption for RC6, and Ameliorated RC6

E. Decryption Throughput for RC6, and Ameliorated RC6

The throughput of the decryption is calculated as the total plaintext in bytes decrypted divided by the decryption time. The throughput is computed for RC6, and Ameliorated RC6 at word size (w)=32,

but different Block size, number of rounds (r), and the length of secret key (b).Which is design parameter of RC6; block size=128 with 4 registers, r=20, b=16 bytes, while Ameliorated RC6; block size=2048 with 64 registers, r=24, b=32 bytes. Table 5-6 and Figure 5-5 show the obtained results, and also show that Ameliorated RC6 achieves maximum decryption throughput than RC6.

Table 5-6: Throughput of Decryption for RC6 and Ameliorated RC6

Throughput of decryption	Algorithm
19.65	RC6
63.27	Ameliorated RC6

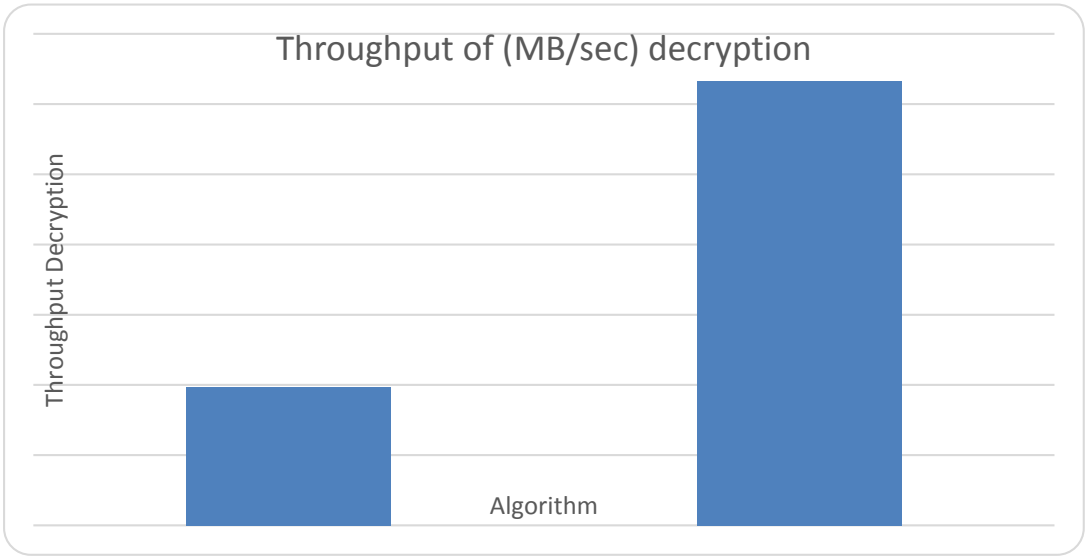

Figure 5-5: Decryption Throughput (MB/sec) for RC6, and Ameliorated RC6

F. Effect of Secret key length on throughput of RC6, and Ameliorated RC6

Table 5-7, and Figure 5-6 illustrates the effect of secret key length on the throughput of encryption and decryption, and the security of encryption and decryption for RC6, and Ameliorated RC6. The throughput is computed as a function of secret key length (b). The results show that the secret key length has insignificant effect on increasing or decreasing the throughput within the same type of encryption algorithm. But significant increase in throughput values is achieved with Ameliorated RC6 compared with RC6, when applying the different secret key length. Increasing the secret key length contributes to increase the complexity and security of the algorithm, so Ameliorated RC6 achieves more security than RC6.

Table 5-7: Throughput As a function of the secret key length (b)

Algorithm	The key length (b)	Throughput of encryption	Throughput of decryption
RC6	16	20.74	19.65
Ameliorated RC6	32	64.3	63.27

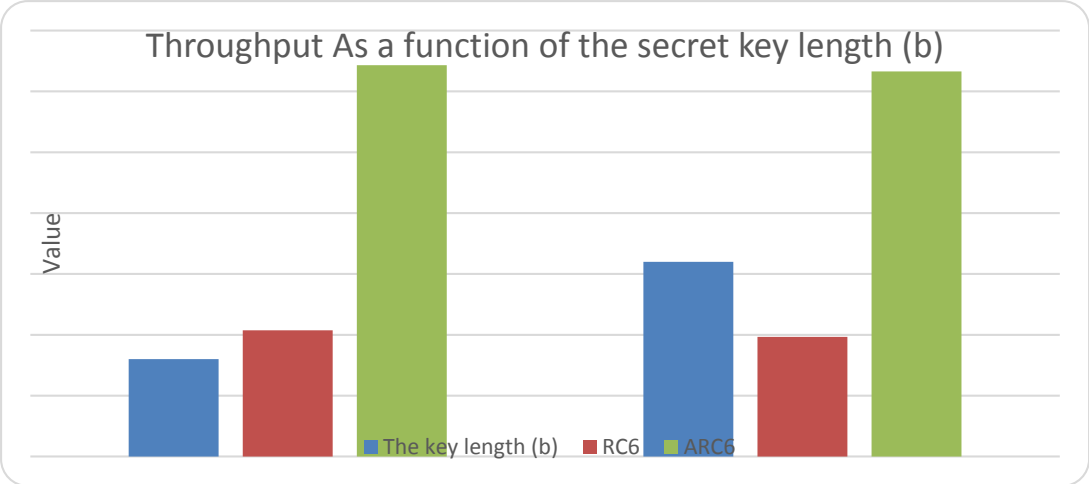

Figure 5-6: Effect of the secret key length (b) on throughputG. Effect of number of round on the throughput for RC6, and Ameliorated RC6

Table 5-8 and Figure 5-7 illustrate the effect of a number of rounds on both the throughput of encryption and decryption, and the security of encryption and decryption for RC6, and Ameliorated RC6. The throughput is computed as a function of number of rounds (r). The results show that Ameliorated RC6 has the highest throughput compared with RC6. The throughput decreases with increasing number of rounds and vice versa. High throughput requires less number of rounds, but high security requires a large number of rounds, so there is a tradeoff between high security and high throughput. The purpose for having more rounds is to achieve a higher level of security, because an increase in rounds translates to an increase in encryption. By increasing the amount of encryption that is done, the resulting cipher text becomes more statistically unrelated to the original plaintext.

Table 5-8: Throughput as a function of number of round (r)

Algorithm	The number of round (r)	Throughput of encryption	Throughput of decryption
RC6	20	20.74	19.65
Ameliorated RC6	24	64.3	63.27

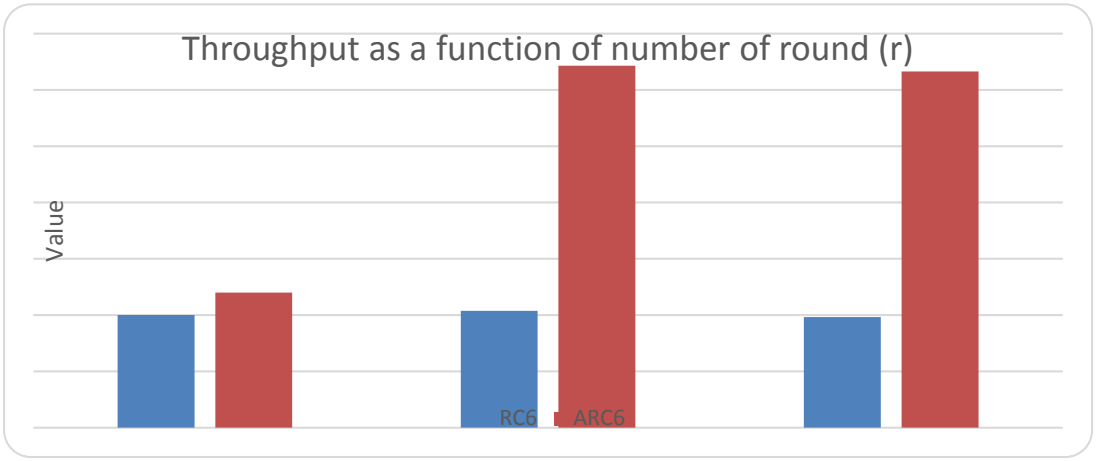

Figure 5-7: Effect of the number of round (r) on throughput

5.2.3 The Security

The most important requirement is stated succinctly in the AES announcement:"The security provided by an algorithm is the most important factor in the evaluation".

The Security of the Cryptography depends on the key which is used for both encryption and decryption. In Ameliorated RC6 achieves more security by increasing number of rounds from 20 to 24, and key size from 16 byte to 32 byte and then XOR-ed with S-Box, which is not used in previous RC6.

5.2.3.1 Security of the Key Schedule

The key schedule of RC6 is the number of words (2r + 4) derived from the user-supplied key for encryption and decryption. While, the key schedule of Ameliorated RC6 is the number of words (32r + 64) derived from the user-supplied key for encryption and decryption, with consider the key size 32 bytes instead of 16 bytes. The S-box used in Ameliorated RC6 is S-box of the MARS algorithm. It is a single S-Box of 512 32-bit words to provide good resistance against linear and differential attacks. After key generation, we XOR-ed keys generated with each byte of the s-box. In this way we might the exclusive-or that provide the biggest contribution to security. The differential attacks do not apply to Ameliorated RC6 because of the high complexity of the key schedule.

5.2.3.2 cryptanalysis attacks

Matching cipher text attack:

The matching cipher text attack requires about 2n/2 cipher text blocks to succeed, where n is the block size. With n = 128 as in RC6, 264 cipher text blocks are required after which an attacker would be able to deduce information about the plaintext blocks, while With n = 2048 as in Ameliorated RC6, 21024 cipher text blocks are required after which an attacker would be able to deduce information about the plaintext blocks. Table 5-9 presents comparison between RC6 and Ameliorated RC6 with matching cipher text attack of algorithms.

Table 5-9 : matching cipher text attack of algorithms

A matching cipher text attack: 2n/2	
RC6	264
Ameliorated RC6	21024

An exhaustive key search Attacks:

An exhaustive key search Attacks will take 2k operations to succeed, where k is the key size. Table 5-10 presents comparison between RC6 and Ameliorated RC6 with exhaustive key search attack of algorithms.

Table 5-10 : exhaustive key search Attacks of algorithms

An exhaustive key search attack: 2k	
RC6	216
Ameliorated RC6	232

Dictionary Attacks:

As the block size is 128 bits, a dictionary attack will require 2 128 different plaintexts to allow the attacker to encrypt or decrypt arbitrary message under an unknown key. So , the proposed algorithm with 2048 bits block size requires 2 2048 different plaintexts. Table 5-11 presents comparison between RC6 and Ameliorated RC6 with dictionary Attacks of algorithms.

Table 5-11 : dictionary Attacks of algorithms

A dictionary Attacks: 2n	
RC6	2128
Ameliorated RC6	22048

In conclude, we show that Ameliorated RC6 performs good resistance to cryptanalysis attack. As a result Ameliorated RC6 achieves more security than RC6 algorithm and any hacker finds difficult to break the algorithm to know the original text.

Chapter 6

Conclusion

6.1 Conclusion

In this work, an enhanced algorithm of RC6 has been proposed, called Ameliorated RC6. The proposed Ameliorated RC6 algorithm increasing security by using 32 bytes instead of 16 bytes key size with adding the S-Box with key generation algorithm and improving performance and maximizing throughput by using of sixty-four working registers instead of four working registers in RC6, and by using a block size of 2048 bits instead of 128 bits. A system of Ameliorated RC6 and RC6 has been built. This system has an enhancements in order to enhance previous RC6, which maximizing throughput and increasing the security level of the system algorithm. The system designed in this thesis compares between the original RC6 and Ameliorated RC6 algorithms by encryption and decryption time, throughput of encryption and decryption for both algorithms.

6.2 Future work

A further improvement would be changing the existing parameters or adding new operations. Moreover, we hope to develop a technique in order to decrease the time of the encryption and decryption as well as a technique to improve the performance.

References

Arora N. Gigras Y., 2014. Block and Stream Cipher Based Cryptographic Algorithms:A Survey. International Journal of Information and Computation Technology, Volume 4, Number 2, 189-196.

Asaithambi.N,. April 2015. A Study on Asymmetric Key Cryptography Algorithms. Journal of Computer Science and Mobile Applications, Vol.3 Issue 4, 8-13.

Charbathia Sh., Sharma S, 2014. A Comparative Study of Rivest Cipher Algorithms,

International Journal of Information & Computation Technology, Vol. 4, No. 17, 1831-183.

Dilpeet Kaur D., Singh Bhathal G., July-September 2014. Improving Encryption Process by Making a Hybrid Encryption Scheme. International Journal for Multi Disciplinary

Engineering and Business Management (IJMDEBM), Volume-2, Issue-3.

El-Fishawy N. A., El-Danaf T. E., Abou Zaid O. M., 2004. A modification of RC6TM Block Cipher Algorithm for Data Security (MRC6). Institute of Electrical and Electronics

Engineers.

Gil-Ho Kim G. H., Jong-Nam Kim J. N., Cho G. Y., Feb. 15-18 2009. An improved RC6

algorithm with the same structure of encryption and decryption. ICACT.

Hashim A. T, Dr.Ali. Y. H., 2010. Proposed Cascaded Design of 640-bit RC6 Block Cipher.

Hashim A. T., Helal B. H., February 2010. Measurement of Encryption Quality of Bitmap Images with RC6, and two modified version Block Cipher, International Journal for Technological Research and Engineering, Vol 28, No.17.

Hashim A. T., Mahdi J. A. , Abdullah S. H., 2010.A Proposed 512 bits RC6 Encryption Algorithm, IJCCCE, Vol.10, No.1.

Hercigonja Z., Gimnazija D., Croatia V., 2016. Comparative Analysis of Cryptographic Algorithms. International Journal of DIGITAL TECHNOLOGY & ECONOMY, Volume 1, Number 2, 127 – 134.

Kirti Aggarwal K., 2015. Comparison of RC6, Modified RC6 & Enhancement of RC6, International Conference on Advances in Computer Engineering and Applications (ICACEA).

ManpreetKaur Er, Manpreet J., Er. Kaur J., May 2017. Data Encryption Using Different Techniques: A Review. International Journal of Advanced Research in Computer Science, Volume 8, No. 4.

Mohamed B., Zaibi Gh., Kachouri A., 2011. Implementation of Rc5 and RC6 Block Ciphers on Digital Images , 2011. IEEE.

Nanda Hanamant Khanapur N. H., Patro A., June 2015. Design and Implementation of Enhanced version of MRC6 algorithm for data security, International Journal of Advanced Computer Research ISSN, Vol. 5 Issue 19.

Nechvatal, J., Barker, E., Bassham, L., Burr, W., Dworkin, M., Foti, J., & Roback, E., 2002. Report on the Development of the Advanced Encryption Standard (AES), National Institute of Standards and Technology.

Niveditha R, Akshaya, Tumkur, Karnataka, 2014. A Survey on Cryptography and

Steganography. International Journal of Science and Research (IJSR), Volume 3

Issue 4.Rivest, R.L, 1997. The RC5 Encryption Algorithm.

Rivest, R.L., Robshaw, M.J.B., Sidney, R., Yin, Y.L, 1998. The Security of the RC6 Block Cipher.Rivest R., Robshaw M., Sidney M., and Yin Y., 1998. The RC6TM Block Cipher, First Advanced Encryption Standard (AES) Conference, Ventura, CA.

Rivest, R., Robshaw, M.J.B., Sidney, R., Yin, Y., Rivest, R.L., Robshaw, M.J.B., Sidney, R., Yin, Y.L, 1998. The RC6 Block Cipher.

Singh1 P., Shende P., December 2014. Symmetric Key Cryptography: Current Trends.

International Journal of Computer Science and Mobile Computing (IJCSMC). Vol. 3,

Issue.12, 410 – 415.

Sritha P., R.Ashokkumar R., Bhuvaneswari S., Vidhya M., December 2014. A new modified RC6 algorithm for cryptographic applications. International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 12.

Stallings, W., 2017. Cryptography and Network Security: Principles and Practice,7th

edition,2017.

Tharun V. Deep, Dr. Siva Reddy V., February 2015. Comparative Analysis of AES Finalist Algorithms and Low Power Methodology for RC6 Block Cipher- A Review. International

Journal For Technological Research and Engineering, Volume 2, Issue 6.

Thenmozhi.C, Sonti K., March, 2013. Analyzing the performance of RC6 using Complex Vedic Multiplier, International Journal of Research in Engineering & Advanced Technology (IJREAT), Vol 1, Issue 1.

Tripathi,R., Agrawal S., June 2014. Comparative Study of Symmetric and Asymmetric

Cryptography Techniques, International Journal of Advance Foundation and Research in Computer (IJAFRC), Volume 1, Issue 6.

Tyagi V., Singh S., April 2012, Enhancement of RC6 (RC6_EN) Block Cipher Algorithm and comparison with RC5 & RC6. Journal of Global Research in Computer Science, Volume 3, No. 4.

Ueda R. T. E.,2009. A new version of the RC6 algorithm, stronger against $\chi 2$ cryptanalysis.

Verma H. K., Punjab J., Singh R. K., March 2012. Performance Analysis of RC6, Twofish and Rijndael Block Cipher Algorithms. International Journal of Computer Applications, Volume 42– No.16.

Wadhwa N., Hussain S. Z., Rizvi S. A., July 2013. A study of MARS, RC6 and SERPENT.

International Journal of Computer Science and Engineering.